VPBank

VPBANK TECHNOLOGY HACKATHON

Chinh phục công nghệ, Kiến tạo tương lai cùng VPBank

Hackathon Challenge 1: Secure Web Application on Cloud

A bank has an external facing web application that is deployed as a microservice using AWS container and serverless services. After running the application on AWS for a year, we have encountered various web application vulnerabilities that have led to the exploitation of misconfigured AWS resources and coding issues.

We are looking for a solution that can proactively protect our web application from public attacks. The solution should follow the OWASP Top 10 security standards. Additionally, the solution should proactively monitor and ensure that all cloud resources adhere to cloud security best practices and the bank's internal standards.

In summary, we need a solution that can:

- Protect the application from attacks by following OWASP Top 10 guidelines
- Proactively monitor our AWS resources to ensure proper configuration as per cloud security best practices and the bank's standards
- Help avoid exploitation of misconfigurations and coding vulnerabilities in the future

Expectations

Deliverables

1. Develop a working prototype web application
2. Security attack simulation
3. Security finding dashboard

Requirements

1	The participant choose AWS platform for the development work.
2	The architecture design of the application should keep simple which follow the AWS Well architected
3	The solution should include the centralized dashboard(s) that provide the security observability at Infrastructure and Application level
4	The solution should be tested/simulated the security incident like DDoS attack or OSWAP Top 10 security attack
5	The solution should be deployed by using IaC (e.g. CloudFormation, Terraform, etc.)

Hackathon Challenge 2: Modernize Web Application on Cloud

A bank's IT division wants to modernize their three-tiered web application from their on-premises servers to the AWS cloud. Their current on-premises setup includes Web servers that serve the website content and a MySQL database that stores the content.

They have some issues with their application today:

- Heavy traffic from marketing campaigns crashes the application
- Manual application updates deployment risks to system availability
- Have to take the site down in the midnight to deploy updates
- No tool to monitor infrastructure and application status

To provide a better experience for customers, they want to move to the AWS cloud to:

- Handle heavy traffic without crashing
- Automate updates deployment without any downtime
- Proactive monitoring infrastructure and application status

Expectations

Deliverables

1. Develop a working prototype application
2. Simulation of the ability to handle heavy traffic
3. Automated CI/CD pipeline
4. Monitoring infrastructure and application dashboard

Requirements

1	The participant choose AWS platform for the development work.
2	The architecture design of the application should keep simple which follow the AWS Well architected
3	The solution should be automatically scale out to handle increasing traffic and scale in when the traffic is decreasing
4	The CI/CD pipeline should be able to deploy/update the application without downtime
5	The solution should provide a monitoring dashboard for infrastructure and application
6	The solution should be deployed by using IaC (e.g. CloudFormation, Terraform, etc.)

Hackathon Challenge 3: Talent Acquisition Search Application

A bank's HR division wants to Develop a model application to support the Bank Talent Acquisition (TA) team in finding suitable candidates from the existing CV database. The application will be built using modern technology, leveraging AWS cloud services, and harnessing the power of generative AI for conversational interaction with TA users.

The application should have the following capabilities:

- Automate data processing to handle unstructured data in various formats (PDF, Word, and Excel) for identification, sorting, and categorization into usable data.
- Provide a simple front end for end users (TA) to input search queries, view response results, and list the top 5 Candidates in order of matching score. Candidate with higher suitability will displayed first, while less suitable ones will highlighted differently. TA users should be able to view aggregates such as the number of submitted CVs falling under enterprise development, average percentage of individuals with particular skills, etc.

Expectations

Deliverables

1. Develop a working prototype application
2. LLM AI models for intelligent searching and response
3. Reporting functionality for total token usage, and total cost

Requirements

1	The participant choose AWS platform for the development work.
2	The architecture design of the application should keep simple which follow the AWS Well architected
3	Utilization of LLM AI models for intelligent searching and response, with the ability to configure and switch between LLM models.
4	Configuration options for custom prompts and selecting LLM models (GPT 3.5, Claude)
5	Reporting functionality for total token usage, and total cost (visualization via dashboard).
6	Achieve answer accuracy of over 90%. Ensure answer response time is less than 20 seconds
7	Utilize AWS native cloud services, including ECS, EKS, Lambda, ALB, RDS database
8	Frontend built with Node.js and Typescript. Backend implemented with Node.js/Python.

Hackathon Challenge 4: Customer Lifetime Value Optimization

In today's interconnected digital landscape, businesses operate within complex ecosystems where data flows seamlessly across various platforms. The challenge is to leverage this wealth of data to enhance Customer Lifetime Value (CLV) systematically. Participants in this hackathon will explore innovative solutions to optimize CLV within a data ecosystem, fostering customer loyalty and sustained revenue growth

In this hackathon, participants will leverage ecosystem’s fictionalized datasets to explore techniques for systematically enhancing CLV. Example focus areas include predicting high-value customer segments, optimizing cross-platform advertising, and crafting personalized recommendations to improve retention.

Successful solutions will creatively harness the provided data to foster loyalty and sustained revenue growth. Participants can showcase machine learning, statistical modeling, or other analytical approaches within their proposals. However, business viability and intuitive interpretation of models are key evaluation criteria. They can simulate data to demonstrate challenges and build the prototype.

Expectations

Deliverables

1. Develop working prototype machine learning model(s)
2. ETL pipeline for data processing
3. Expose API to query the result or prediction
4. GUI as presentation layer for demo

Requirements

1	Teams can simulate data to demonstrate challenges and apply machine learning, statistical modeling or other analytical techniques
2	The architecture design of the application should keep simple which follow the AWS Well architected
3	Understanding of how various components within an ecosystem interact. Knowledge of bank and finance ecosystems and their dynamics.
4	The ETL pipeline should be provide the full data processing capability from raw to ready to use and leverage AWS Services at scale
5	Develop intuitive data visualizations and dashboards or web application for end business users
6	Strong teamwork and collaboration skills, especially when dealing with interconnected ecosystems. Effective communication skills to articulate complex technical concepts to a non-technical audience.
7	The participant choose AWS platform for the development work. The AIML and web application solution will be hosted on the AWS platform

Hackathon Challenge 5: Customer 360

In the ecosystems, it’s challenging to obtain a holistic view of customer profiles (a single customer ID) across different subsidiaries and a view of customer interactions across different channels leading to fragmented data, inefficiencies in communication and missed opportunities for personalized engagement and cross-sell opportunities

In this hackathon, participants will creatively create a comprehensive and accurate profile of each customer by aggregating data from various touchpoints. Participants can showcase how they design and create data models, build ETL pipelines and enrich data from different data sources by leveraging different database systems, big data platform and programing languages for data processing and transformation. They can simulate data to demonstrate challenges and build the prototype.

Expectations

Deliverables

1. Develop working prototype machine learning model(s)
2. ETL pipeline for data processing
3. Expose API to query the result
4. GUI as presentation layer for demo

Requirements

1	Teams can simulate data to demonstrate challenges and build the prototype
2	The architecture design of the application should keep simple which follow the AWS Well architected
3	Designing and creating data models that represent customer interactions across different touchpoints
4	Expertise in building, extract, transform, load process to integrate and clean data form various sources
5	Proficiency in database systems (SQL, NoSQL) for efficient storage and retrieval of customer data
6	Python, Java, Scala for data processing and transformation Familiar with big data platforms like Apache Hadoop, Spark or similar tools for handling large scale data and graph database
7	Develop intuitive data visualizations and dashboards or web application for end business users
8	The participant choose AWS platform for the development work. The solution will be hosted on the AWS platform

Hackathon Challenge 6: Design and Develop a Data Tokenization System

The goal of this challenge is to design and develop a robust data tokenization system capable of preserving the original format and statistical distribution of sensitive data while ensuring data security and compliance.

The system should support the tokenization of the following data types:

- Name
- Email address
- Phone number
- Credit/debit card number
- Date of birth
- Identification number (e.g., Social Security Number, National ID)
- Numeric data (e.g., account numbers, transaction IDs)

The tokenization process must ensure that the original data format and statistical distribution are maintained after tokenization.

Expectations

Deliverables

1. Develop working prototype Data Tokenization System
2. A detailed technical report describing the system architecture, design choices, implementation details, and testing methodology.
3. GUI as presentation layer for demo

Requirements

1	The system should provide seamless integration with Amazon Redshift and Amazon S3 for data storage and retrieval.
2	The solution should support batch data tokenization, allowing for efficient processing of large datasets. And should be scalable and able to handle increasing volumes of data as required.
3	The system must be capable of handling data in English and Vietnamese languages.
4	The solution should be accompanied by comprehensive testing, demonstrating the correctness and reliability of the tokenization and de-tokenization processes.
5	Develop intuitive data visualizations and dashboards or web application for end business users
6	The participant choose AWS platform for the development work. The solution will be hosted on the AWS platform

Hackathon Challenge 7: Architecture Design Generator

Everyday, there are projects/initiatives/developments which are initiated in the bank. Some of them are complex enough to require serious efforts to create the right design of the solution. The design includes but is not limited to software architecture, integration architecture, data architecture, with consideration about security, maintainability, scalability, availability, etc. This task is often done in collaboration between development teams and architecture team. However, due to limited capacity of the centralized architecture team, some projects might need to wait and could not be processed in expected time.

To solve this problem, the bank wants to leverage the power of generative AI and open knowledge on Internet (for example: architecture blueprint shared by AWS, etc.) and on Intranet to help teams and architects generate the first draft of designs based on the description of the requirements.

Expectations

Deliverables

1. Develop working prototype solution
2. Generated architecture design using prompts
3. Admin console for demo

	Requirements
1	Able to generate architecture/design with justification/explanation based on the provided description
2	Able to generate the design in editable format (for ex: Visio, draw.io, etc.)
3	Answer response time: < 10s
4	Ability to take feedback/enhance the designs
5	Plus: Ability to generate templates of code based on the design
6	Plus: Ability to use pre-built design (i.e. inject internal knowledge base into application's intelligence)
7	The participant choose AWS platform for the development work. The solution will be hosted on the AWS platform

Thank You