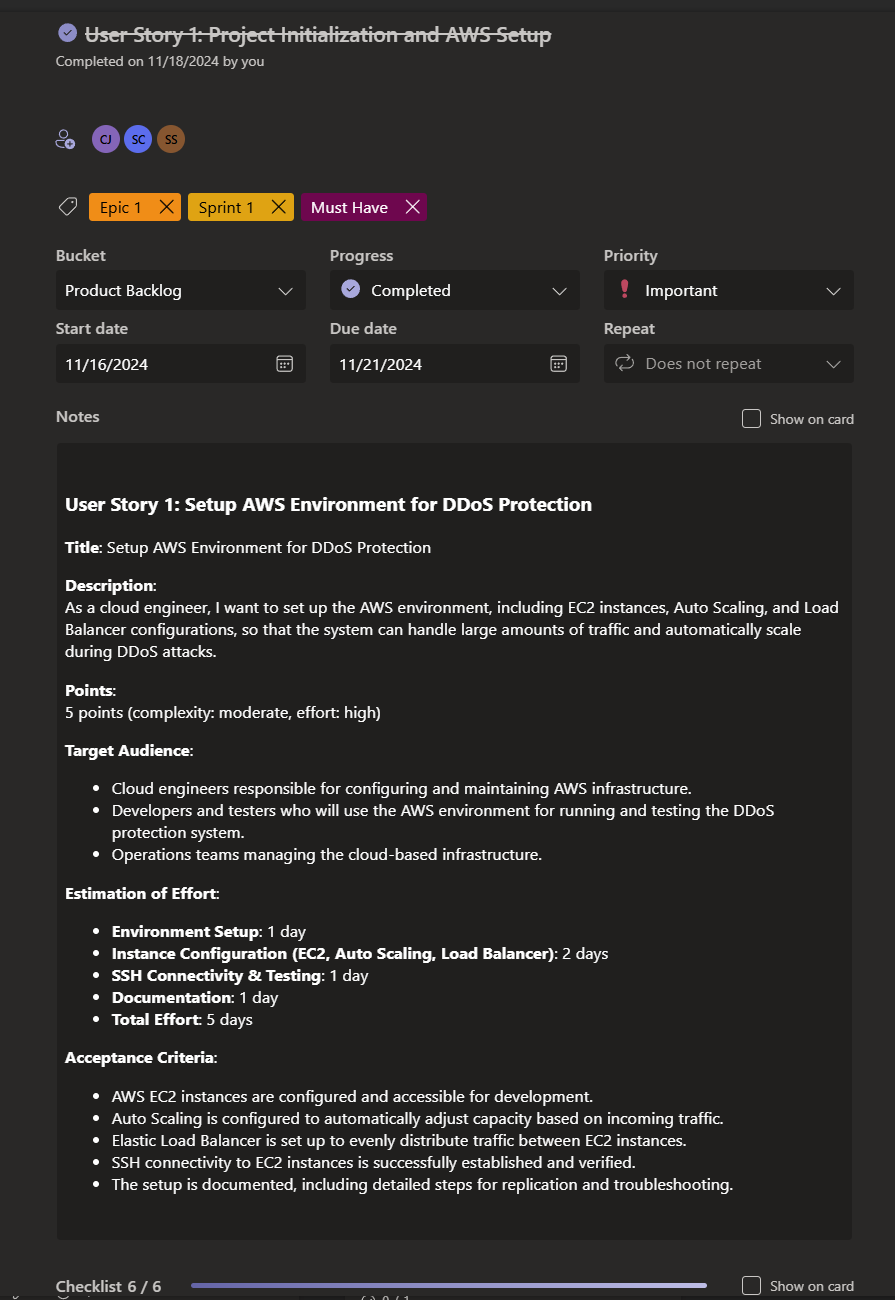
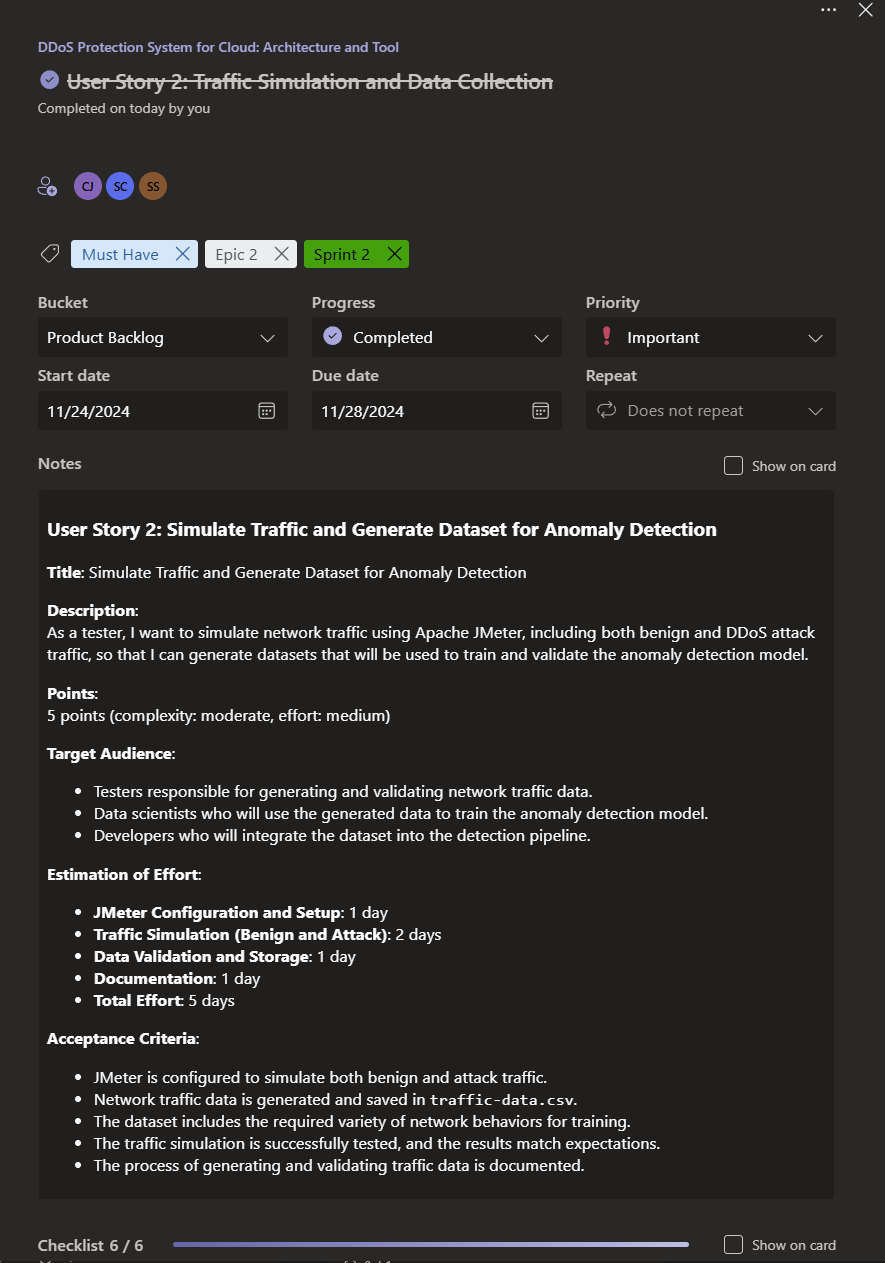
**AGILE DEVELOPMENT**

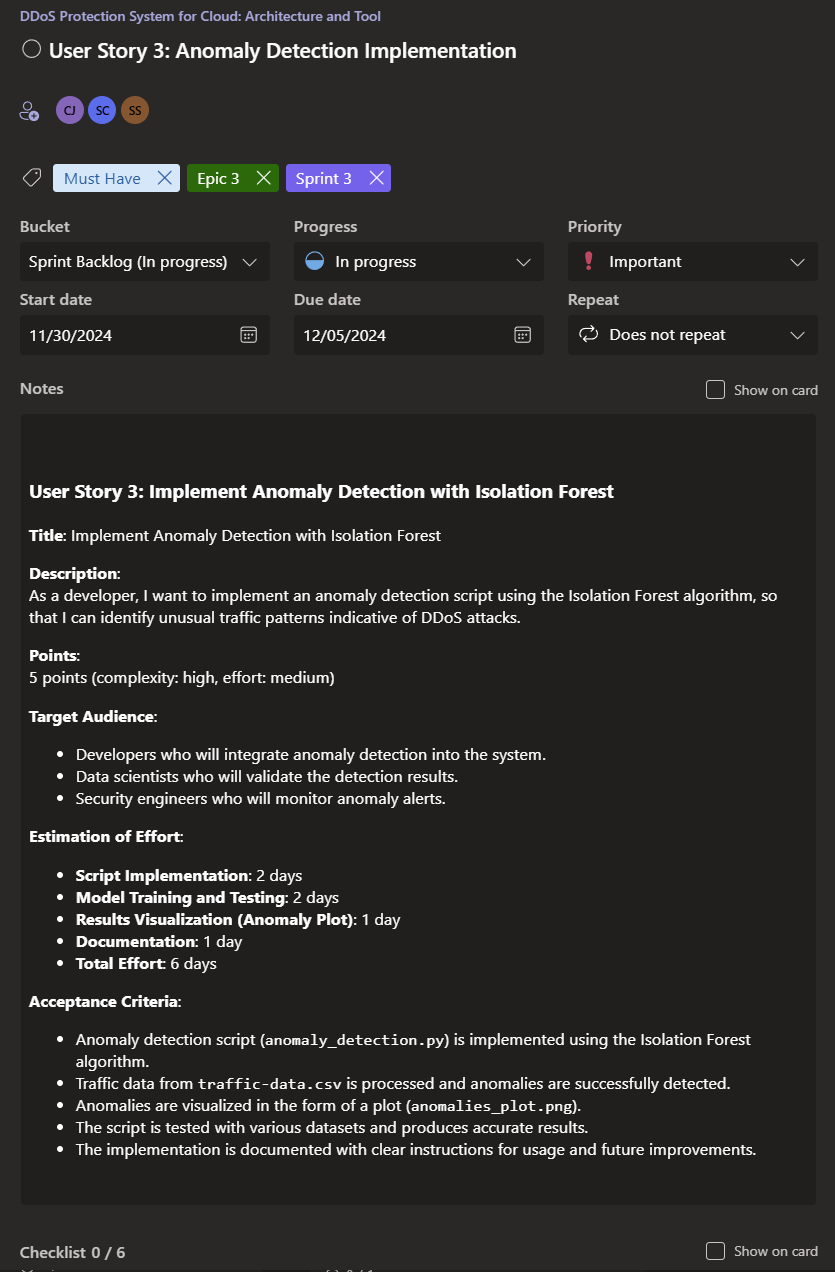
1. **Agile MS Board**
2. **Product Backlog**
   1. **User Story 1**

****

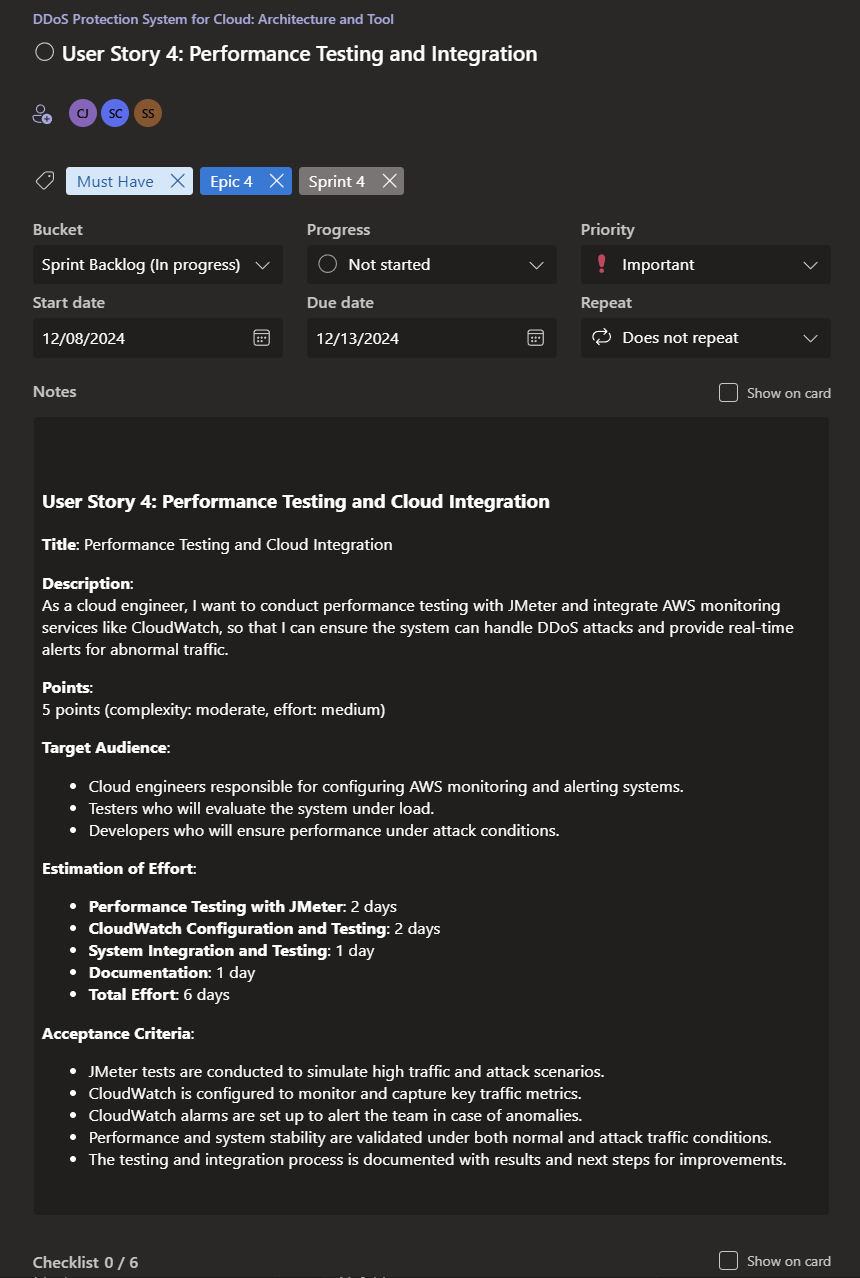
* 1. **User Story 2**

****

* 1. **User Story 3**

****

* 1. **User Story 4**

****

1. **Functional Documents**
   1. **User Story 1**

**1. Introduction**

The objective of this user story is to set up the AWS environment for hosting and managing the DDoS protection system. This involves configuring AWS services such as EC2 instances, Auto Scaling, and Load Balancer to ensure the system can automatically scale in response to increased traffic, particularly during DDoS attacks. This infrastructure setup will serve as the foundation for deploying and testing the DDoS detection and mitigation system.

**2. Product Goal**

The goal is to create a scalable AWS infrastructure to simulate and handle varying traffic loads, including benign and malicious traffic, ensuring that the system is resilient during DDoS attacks. This setup is crucial for testing the performance and effectiveness of DDoS protection mechanisms under real-world conditions.

**3. Demography (Users Location)**

* **Target Users:** Cloud engineers, security testers, developers working on DDoS protection.
* **User Characteristics:** Technical users with expertise in cloud infrastructure and DDoS mitigation strategies.
* **Location:** Global usage with a focus on cloud environments and organizations requiring DDoS protection.

**4. Business Processes**

* **AWS Infrastructure Setup:**
  + Configure EC2 instances for running the DDoS protection system.
  + Set up Auto Scaling to manage server capacity during fluctuating traffic.
  + Configure Elastic Load Balancer to distribute traffic evenly among EC2 instances.
* **Traffic Simulation:**
  + Simulate various traffic patterns, including benign and DDoS attacks, to evaluate system performance.
  + Monitor the infrastructure using AWS CloudWatch for traffic metrics and scaling activity.

**5. Features**

* **EC2 Instance Configuration:**
  + Launch EC2 instances with the necessary configurations (CPU, memory, etc.).
  + Set up security groups and IAM roles for secure access.
* **Auto Scaling Setup:**
  + Configure Auto Scaling policies to scale instances based on CPU usage or incoming traffic.
  + Ensure the system can scale up and down based on defined thresholds.
* **Elastic Load Balancer:**
  + Set up the Elastic Load Balancer (ELB) to distribute traffic among EC2 instances.
  + Ensure the load balancer is properly configured to handle both normal and attack traffic.
* **SSH Connectivity:**
  + Establish SSH access for remote management and troubleshooting of EC2 instances.

|  |  |
| --- | --- |
| **ROLE** | **Access Level** |
| Developer | Full access to configure and manage AWS resources. |
| Security Tester | Access to traffic logs and performance metrics. |
| Admin | Full access to AWS environment and all configurations. |

**6. Authorization Matrix**

**7. Assumptions**

* AWS environment is stable, and resources are available.
* Proper AWS IAM roles and permissions are configured for each user.
* Sufficient network bandwidth is available for testing traffic simulations.
* Security groups and access control lists are set up to restrict access as needed

**8. Target Audience**

* **Audience:** Cloud engineers, system architects, security teams, and developers working on DDoS protection systems.
* **Effort Estimation:** Approximately 5 days to 1 week for setup and testing, depending on complexity and available resources.

**9. Acceptance Criteria**

* AWS EC2 instances are set up and accessible.
* Auto Scaling is configured to automatically scale the system based on incoming traffic.
* Load Balancer is properly distributing traffic among EC2 instances.
* SSH access to EC2 instances is tested and verified.
* The infrastructure setup is documented and includes steps for scaling, troubleshooting, and monitoring.

**10. Checklist**

* EC2 instances are launched and configured.
* Auto Scaling policies are defined and tested.
* Elastic Load Balancer is set up and routing traffic correctly.
* SSH connectivity is established and verified.
* The setup is documented, including troubleshooting steps and scaling guidelines.
  1. **User Story 2**

**1. Introduction**

This user story focuses on simulating traffic and generating datasets using Apache JMeter for training the Isolation Forest anomaly detection model. The datasets will include both benign and DDoS attack traffic to help develop a robust system for detecting anomalies in real-time cloud environments.

**2. Product Goal**

The goal is to simulate realistic network traffic patterns using JMeter and generate labelled datasets that will be used for training the anomaly detection system. The dataset must accurately represent real-world scenarios to ensure the model's reliability and effectiveness.

**3. Demography (Users Location)**

* **Target Users:** Cloud security engineers, data scientists, AI developers.
* **User Characteristics:** Users familiar with traffic simulation, data preprocessing, and model training workflows.
* **Location:** Organizations globally, with a focus on those operating in cloud-hosted environments.

**4. Business Processes**

* **Traffic Simulation:**
  + Configure JMeter to simulate benign traffic with varying patterns.
  + Introduce DDoS attack scenarios using stress-testing configurations.
* **Data Logging:**
  + Capture traffic data and export logs in CSV format.
  + Record key metrics such as source IP, destination IP, traffic volume, and timestamp.
* **Data Preprocessing**:
  + Label traffic entries as **benign** or **malicious** based on predefined criteria.
  + Normalize and clean the data for consistency.

**5. Features**

* **Traffic Simulation with JMeter:**
  + Simulate traffic using pre-configured JMeter test plans.
  + Generate scenarios for both normal usage and DDoS attacks.
* **Data Export and Labelling:**
  + Export raw traffic logs to CSV format.
  + Label datasets with accuracy to differentiate between benign and malicious activities.
* **Dataset Review:**
  + Validate and balance the dataset for model training, ensuring an equal representation of benign and attack traffic.

|  |  |
| --- | --- |
| **ROLE** | **Access Level** |
| Developer | Full access to configure and manage AWS resources. |
| Data Scientist | Access to processed and labelled datasets. |
| Security Analyst | Access to simulation logs for analysis. |

**6. Authorization Matrix**

**7. Assumptions**

* The JMeter test plan runs without errors during simulations.
* Dataset labelling criteria are well-defined and consistently applied.
* Necessary tools and dependencies for data preprocessing are available and functional.

**8. Target Audience**

* **Audience:** Data scientists, cloud security teams, and developers working on anomaly detection systems.
* **Effort Estimation:** Approximately 5 days for simulation, data generation, and preprocessing.

**9. Acceptance Criteria**

* Traffic logs are generated and exported successfully using JMeter.
* Logs are processed into CSV format without errors.
* Data entries are accurately labelled as benign or malicious.
* The final dataset is balanced and ready for model training.

**10. Checklist**

* JMeter configured for traffic simulation.
* Traffic logs exported to CSV format.
* Data labelled correctly for benign and malicious traffic.
* Dataset reviewed and balanced for model training.
  1. **User Story 3**

**1. Introduction**

This user story involves developing an anomaly detection script using the Isolation Forest algorithm. The script will analyse simulated traffic data and detect deviations that indicate potential DDoS attacks. The goal is to build a reliable detection mechanism that identifies anomalies with high precision.

**2. Product Goal**

To implement an anomaly detection system capable of analysing network traffic data and identifying suspicious patterns indicative of DDoS attacks. The output will be visualized to validate the model's effectiveness.

**3. Demography (Users Location)**

* **Target Users:** Data scientists, security analysts, AI developers.
* **User Characteristics:** Users experienced in machine learning and anomaly detection frameworks.
* **Location:** Organizations globally, particularly those with cloud-based systems.

**4. Business Processes**

* **Model Development:**
  + Import and preprocess the labelled dataset.
  + Train the Isolation Forest model using normalized data.
* **Anomaly Detection:**
  + Use the model to predict anomalies in traffic logs.
  + Output results with anomaly scores for further analysis.
* **Visualization**:
  + Generate plots (e.g., anomalies\_plot.png) to highlight detected anomalies.
  + Validate detection results through visual inspection.

**5. Features**

* **Model Training:**
  + Train the Isolation Forest model on the labelled dataset.
  + Optimize hyperparameters to enhance detection accuracy.
* **Anomaly Scoring:**
  + Assign anomaly scores to traffic entries.
  + Classify traffic as normal or anomalous based on threshold values.
* **Visualization:**
  + Generate visualizations to present detected anomalies.
  + Use plots for result validation and debugging.

|  |  |
| --- | --- |
| **ROLE** | **Access Level** |
| Data Scientist | Access to raw datasets and preprocessing tools. |
| Developer | Full access to model training and testing scripts. |
| Security Analyst | Access to detection results and visualizations. |

**6. Authorization Matrix**

**7. Assumptions**

* The labelled dataset is accurate and pre-processed.
* The Isolation Forest algorithm is appropriate for the use case.
* Visualization tools are installed and configured correctly.

**8. Target Audience**

* **Audience:** Security teams, data scientists, cloud solution architects.
* **Effort Estimation:** Approximately 3-4 days for model implementation and testing.

**9. Acceptance Criteria**

* The Isolation Forest model is successfully trained and tested.
* Traffic entries are scored and classified accurately.
* Visualization plots are generated and reviewed for correctness.

**10. Checklist**

* Dataset pre-processed and normalized.
* Isolation Forest model trained and optimized.
* Anomaly scores assigned to traffic entries.
* Visualization plots generated and validated.
  1. **User Story 4**

**1. Introduction**

This user story focuses on performance testing and integrating the anomaly detection system into the cloud environment. The aim is to evaluate system scalability, monitor performance during high-traffic conditions, and set up CloudWatch alarms for real-time monitoring and alerts.

**2. Product Goal**

To integrate the anomaly detection system into a live environment, ensuring it scales under high traffic loads while maintaining accuracy. Alerts and performance metrics will be used to monitor system health.

**3. Demography (Users Location)**

* **Target Users:** Cloud engineers, security teams, operations managers.
* **User Characteristics:** Users responsible for maintaining cloud infrastructure and monitoring performance.
* **Location:** Organizations globally using cloud-hosted services.

**4. Business Processes**

* **Performance Testing:**
  + Use JMeter to simulate high-traffic conditions, including attack scenarios.
  + Monitor system scalability and response times.
* **System Integration:**
  + Deploy the anomaly detection script in the cloud environment.
  + Configure CloudWatch for traffic monitoring and alerts.
* **Alert Setup**:
  + Define thresholds for triggering alerts.
  + Set up email notifications for anomalies and high-traffic scenarios.

**5. Features**

* **Traffic Simulation:**
  + Simulate high traffic loads with JMeter for performance testing.
* **Cloud Integration:**
  + Deploy detection scripts in the cloud environment (AWS EC2).
* **Monitoring and Alerts:**
  + Configure CloudWatch to monitor traffic and system performance.
  + Send alerts via email when thresholds are breached.

|  |  |
| --- | --- |
| **ROLE** | **Access Level** |
| Operations Manager | Read-only access to performance reports. |
| Cloud Engineer | Full access to cloud environment and monitoring. |
| Security Analyst | Access to monitoring metrics and alert configurations. |

**6. Authorization Matrix**

**7. Assumptions**

* Cloud environment is set up and accessible.
* JMeter tests run successfully without errors.
* CloudWatch is correctly configured for monitoring.

**8. Target Audience**

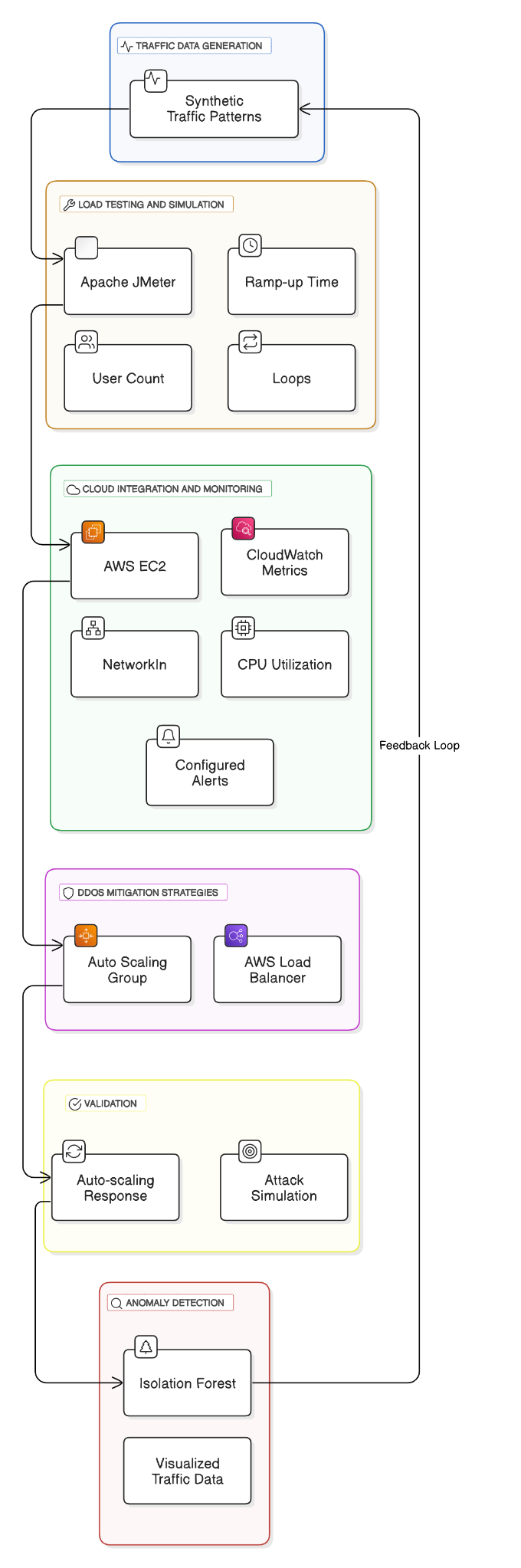
* **Audience:** Cloud security teams, operations managers, cloud engineers.
* **Effort Estimation:** Approximately 4-5 days for testing and integration.

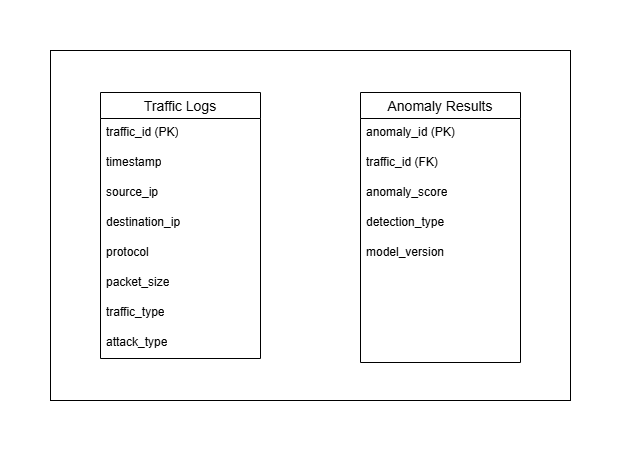
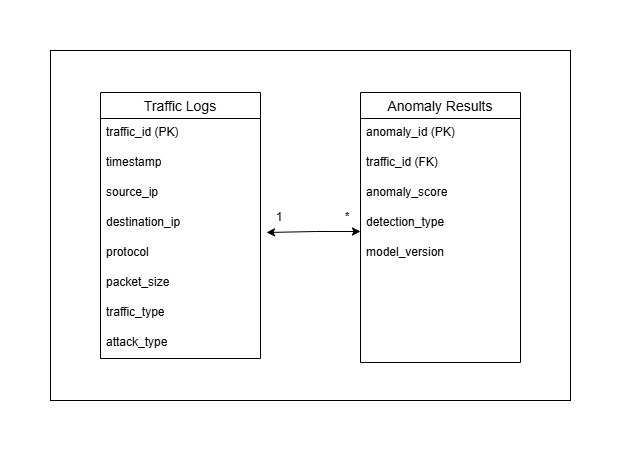
**9. Acceptance Criteria**

* Performance tests are completed with metrics recorded.
* The anomaly detection system is deployed and functional in the cloud.
* CloudWatch alarms are configured and sending alerts for anomalies.

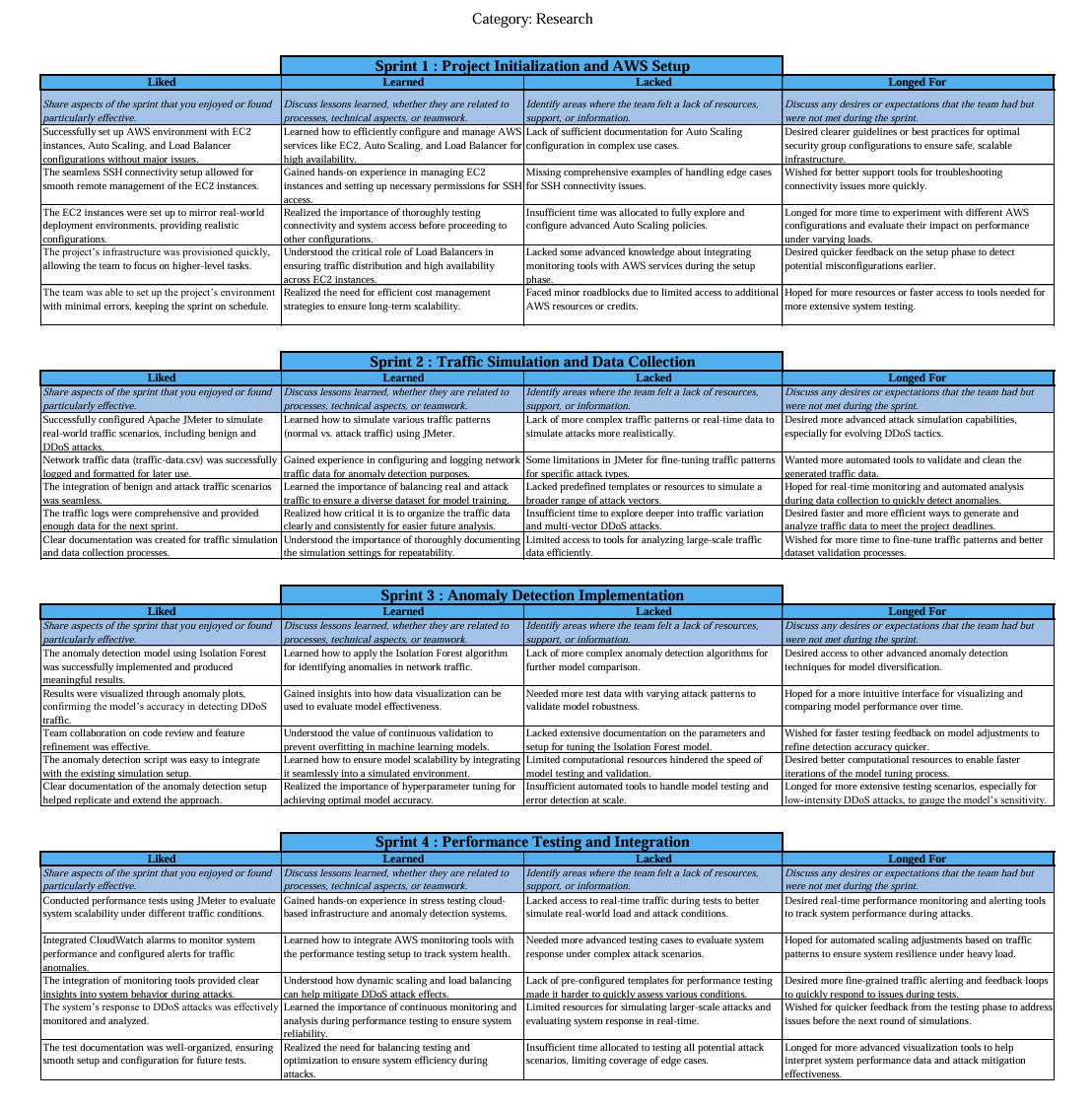
**10. Checklist**

* JMeter traffic simulations executed.
* Anomaly detection script deployed in the cloud.
* CloudWatch metrics and alerts configured.
* Alerts validated for accuracy and timely delivery.

1. **Architecture Document**
   * **Architecture Diagram:**

* + **Scheme Diagram:**
  + **E-R Diagram:**

1. **Sprint Retrospective Document**

****