**User Story 1: Implement Gated Security Reviews per Environments**

Tasks:

* Create a security review checklist for each environment (development, staging, production).
* Define the security gates and criteria for each stage of the software development lifecycle.
* Integrate the security review checklist into the CI/CD pipeline.
* Configure automated security scanning tools to evaluate code and container images.
* Establish a process for handling failed security reviews, including remediation steps and reevaluation.

**User Story 2: Address Vendor Vulnerabilities**

Tasks:

* Identify the vendors whose components are used in the AKS environment.
* Establish a process to track and monitor vulnerabilities in vendor components.
* Create a vulnerability prioritization framework based on severity and impact.
* Develop a process for applying vendor patches and updates in a timely manner.
* Set up communication channels with vendors to receive vulnerability alerts and updates.
* Perform regular vulnerability assessments and scans to identify potential risks.

**User Story 3: Integrate with Observability Tools**

Tasks:

* Identify the observability tools/frameworks to integrate with (e.g., Prometheus, Grafana, Azure Monitor).
* Determine the metrics, logs, and events required for monitoring Starburst and Immuta.
* Set up data collection agents or instrumentation for gathering the required observability data.
* Define thresholds and rules for triggering alerts based on observed metrics.
* Configure the integration with observability tools, ensuring proper visualization and dashboards.

**User Story 4: Design AKS Architecture and Deployment**

Tasks:

* Develop an architecture diagram depicting the AKS infrastructure and component interactions.
* Determine the number and sizing of AKS clusters required for each environment.
* Define the deployment strategy, including containerization, image registries, and deployment manifests.
* Establish the configuration for high availability, scaling, and fault tolerance in AKS clusters.
* Implement monitoring and logging agents within AKS clusters for operational visibility.

**User Story 5: Implement Security and Access Controls**

Tasks:

* Define network policies to restrict inbound and outbound traffic to AKS clusters.
* Configure RBAC roles and permissions to control access to cluster resources.
* Implement authentication mechanisms (e.g., Azure Active Directory, Kubernetes Service Accounts).
* Configure authorization policies to enforce fine-grained access controls.
* Enable encryption for data at rest and in transit within AKS clusters.

**User Story 6: Set up Data Storage and Persistence**

Tasks:

* Determine the storage requirements for Starburst and Immuta (e.g., Azure Disk, Azure Files).
* Set up persistent storage volumes and claims for database and file storage.
* Configure backup and disaster recovery processes for AKS persistent volumes.
* Implement data retention policies and archiving mechanisms.

**User Story 7: Networking Configuration for AKS**

Tasks:

* Design the virtual network architecture for AKS clusters, including subnets and IP ranges.
* Set up network security groups (NSGs) to control inbound and outbound traffic at the subnet level.
* Configure load balancers and ingress controllers for routing external traffic to AKS services.
* Establish secure communication channels between AKS clusters and other network resources.

**User Story 8: Scaling and Performance Optimization**

Tasks:

* Define the scaling strategy for AKS clusters based on expected workloads.
* Implement automatic scaling based on CPU/memory utilization or custom metrics.
* Optimize resource allocation for Starburst and Immuta containers.
* Perform performance tuning and query optimization for Starburst SQL queries.

**User Story 9: Integration with CI/CD Pipelines**

Tasks:

* Identify the CI/CD tooling in use (e.g., Azure DevOps, Jenkins).
* Integrate AKS deployment into the CI/CD pipelines.
* Automate the deployment of Starburst and Immuta containers and configurations.
* Implement testing and validation steps in the CI/CD pipeline for AKS deployments.