**Azure Kubernetes Service (AKS) PRD** **Feature:** Support for CiliumEndpointSlices with Azure CNI by Cilium  
 **Document Version:** 1.0  
 **Document Date:** April 1, 2025  
 **Prepared By:** AKS Product Team

### **1. Document Overview**

This PRD defines the requirements for integrating **CiliumEndpointSlices** with **Azure CNI by Cilium** into Azure Kubernetes Service (AKS). The goal is to enhance AKS’s networking capabilities by leveraging Cilium’s dynamic endpoint management and advanced network policy enforcement. This document details the business rationale, feature description, user scenarios, functional and non-functional requirements, assumptions, dependencies, timeline, and future considerations to guide design, development, testing, and release.

### **2. Business Objectives**

* **Enhanced Networking Efficiency:**
  + Enable dynamic grouping of service endpoints for rapid service discovery and efficient resource utilization.
* **Improved Security & Policy Enforcement:**
  + Utilize Cilium’s native network policy engine to enforce granular security rules that align with enterprise compliance requirements.
* **Optimized Developer & Operator Experience:**
  + Provide a seamless configuration experience via Azure Portal, CLI, and APIs that reduces operational overhead and accelerates deployments.
* **Competitive Differentiation:**
  + Position AKS as a leader in managed Kubernetes by integrating state-of-the-art networking features that meet the evolving needs of cloud-native applications.
* **Operational Cost Efficiency:**
  + Enhance the scalability and stability of clusters to reduce downtime and lower management costs, particularly in large-scale deployments.

### **3. Feature Overview**

The **Support for CiliumEndpointSlices with Azure CNI by Cilium** feature will enable AKS to:

* **Dynamically Manage Endpoints:** Automatically group and manage pod endpoints into slices, which allows for faster lookup, improved scalability, and efficient load balancing.
* **Enforce Advanced Network Policies:** Leverage Cilium’s rich network policy language to implement fine-grained security policies across dynamic endpoint slices.
* **Enhance Observability:** Provide detailed logging, metrics, and diagnostic information through integrated Azure Monitor and Cilium observability tools, enabling proactive troubleshooting and performance tuning.
* **Offer Configurability:** Allow customers to enable or disable the feature during cluster provisioning, with customizable parameters (e.g., refresh intervals, slice size thresholds) to suit diverse operational needs.

### **4. User Stories & Use Cases**

* **Cluster Operator:** *“As a cluster operator, I need the ability to deploy clusters with enhanced endpoint management so that service discovery remains fast and reliable, even as I scale to thousands of pods.”*
* **Network Administrator:** *“As a network administrator, I require the integration of advanced network policy controls to restrict unauthorized traffic while ensuring legitimate workloads communicate efficiently.”*
* **Application Developer:** *“As a developer, I want improved service availability and reduced latency for inter-service communications, which will help maintain application performance under dynamic scaling conditions.”*
* **Security Engineer:** *“As a security engineer, I need real-time observability of network policies and endpoint changes to quickly identify and remediate potential vulnerabilities.”*

### **5. Functional Requirements**

1. **Integration & Configuration:**
   * **FR1.1:** Provide an option within the AKS cluster creation workflow (via Azure Portal, CLI, and ARM templates) to enable CiliumEndpointSlices with Azure CNI.
   * **FR1.2:** Allow configuration of endpoint slice parameters such as refresh intervals, maximum slice size, and error thresholds.
   * **FR1.3:** Include toggle options for advanced logging and observability settings specific to endpoint slice operations.
2. **Endpoint Slice Management:**
   * **FR2.1:** Automatically create, update, and delete endpoint slices in real time based on changes in pod status.
   * **FR2.2:** Ensure synchronization between the Kubernetes API and the underlying Cilium management layer to reflect accurate endpoint state.
   * **FR2.3:** Provide mechanisms for manual override and re-synchronization in case of discrepancies.
3. **Network Policy Enforcement:**
   * **FR3.1:** Integrate Cilium’s policy engine to apply network security rules across all endpoint slices.
   * **FR3.2:** Support both ingress and egress policy enforcement with detailed audit logging.
   * **FR3.3:** Maintain backward compatibility with existing Azure CNI policy constructs to ensure a smooth transition for existing customers.
4. **Observability & Monitoring:**
   * **FR4.1:** Integrate with Azure Monitor and Log Analytics to capture key performance indicators, events, and error logs for endpoint slice operations.
   * **FR4.2:** Provide out-of-the-box dashboards that visualize endpoint slice health, latency, and error rates.
   * **FR4.3:** Enable alert configuration for critical events such as synchronization failures or policy enforcement breaches.
5. **Backward Compatibility & Fallback Mechanisms:**
   * **FR5.1:** Ensure that clusters without the feature enabled continue to operate using the default endpoint management system.
   * **FR5.2:** Implement robust error handling that automatically reverts to traditional endpoint management in the event of feature failure.

### **6. Non-Functional Requirements**

* **Performance:**
  + **NFR1:** Endpoint slice updates must propagate within 100ms of pod state changes to ensure near real-time accuracy.
* **Scalability:**
  + **NFR2:** The solution must support clusters with up to 5,000 pods across 500 nodes without degradation in performance.
* **Reliability:**
  + **NFR3:** The networking subsystem, including endpoint slice management, should target a 99.99% uptime.
* **Security:**
  + **NFR4:** All communications between components must be encrypted, and the solution must integrate with Azure Active Directory for secure authentication and authorization.
* **Usability:**
  + **NFR5:** The feature must be easy to enable and configure, with comprehensive documentation and in-app guidance provided through the Azure Portal.
* **Maintainability:**
  + **NFR6:** The codebase should be modular, with clear logging and diagnostic messages, to facilitate future updates and troubleshooting.

### **7. Assumptions & Dependencies**

* **Dependencies:**
  + **D1:** The feature depends on the latest stable release of Cilium that supports EndpointSlices.
  + **D2:** Integration with Azure CNI components, Azure Monitor, and Log Analytics is required.
  + **D3:** Compatibility with the current and future Kubernetes API versions must be maintained.
* **Assumptions:**
  + **A1:** Customers deploying AKS clusters are familiar with basic Kubernetes networking concepts.
  + **A2:** Future updates to Cilium and Azure CNI will preserve core API contracts to ensure backward compatibility.
  + **A3:** The underlying Azure infrastructure (compute, storage, networking) will support the additional load imposed by dynamic endpoint management.

### **8. Timeline & Milestones**

| **Phase** | **Milestones** | **Timeframe** |
| --- | --- | --- |
| **Phase 1: Requirements & Design** | - Finalize feature requirements and design documentation.  - Conduct feasibility study and risk assessment. | April 2025 |
| **Phase 2: Prototype Development** | - Develop a working prototype integrating CiliumEndpointSlices with Azure CNI.  - Internal integration tests and performance benchmarking. | May – June 2025 |
| **Phase 3: Beta Release** | - Deploy beta release to selected enterprise customers.  - Gather user feedback and monitor feature performance in production-like environments. | July – September 2025 |
| **Phase 4: General Availability** | - Finalize documentation, support guides, and training materials.  - Public GA release with full monitoring and rollback capabilities. | October 2025 |
| **Phase 5: Post-Launch Support** | - Monitor feature adoption and performance.  - Plan for iterative improvements and potential bug fixes. | November 2025 onward |

### **9. Future Considerations**

* **Enhanced Analytics:**
  + Develop deeper analytics features that provide predictive insights and trend analysis on endpoint slice behavior and network policy effectiveness.
* **User Interface Enhancements:**
  + Expand Azure Portal’s capabilities to visually display endpoint slice metrics and detailed network policy maps.
* **Multi-Cloud Integration:**
  + Explore the possibility of extending the feature to support multi-cloud or hybrid environments, enhancing flexibility for customers.
* **Performance Tuning:**
  + Investigate further optimizations for large-scale deployments and consider community feedback to fine-tune synchronization intervals and error handling processes.
* **Community Contributions:**
  + Engage with the open-source community to integrate additional use cases and enhancements that could benefit a broader audience.

### **10. Appendix**

#### **Glossary**

* **CiliumEndpointSlices:** A scalable method for managing and grouping network endpoints, enabling efficient service discovery and load balancing in Kubernetes.
* **Azure CNI:** Azure Container Networking Interface that manages networking for containers deployed on Azure Kubernetes Service.
* **Endpoint Slices:** Kubernetes objects that serve as a scalable alternative to traditional Endpoints for managing service traffic.
* **RBAC:** Role-Based Access Control used for managing permissions within the cluster.
* **CI/CD:** Continuous Integration/Continuous Deployment processes that streamline application development and delivery.

#### **References**

* [Azure CNI Documentation](https://docs.microsoft.com/azure/aks/)
* Cilium Official Documentation
* Kubernetes Networking Best Practices

#### **Contacts**

* **Product Manager:** Jamie Rivera – jamie.rivera@example.com
* **Technical Lead:** Alex Chen – alex.chen@example.com
* **Support & Feedback:** aks-support@example.com