**Audit Document for Container Architecture Rollout Process**

**2.1 Document Accountability, Roles, and Process Overview**

**Accountable Parties:**

* **NEF (Container Platform Team): Accountable for managing and maintaining EKS/AKS clusters, providing base container images, and ensuring compliance with security baselines for containerized workloads.**
* **Security Architect: Oversees container security architecture, ensures configurations align with CSAM guidelines, and reviews changes for security compliance.**
* **Application Security Team (AppSec): Manages vulnerability scanning tools (e.g., Sneak for static scans, CrowdStrike Falcon for runtime scans) and ensures identified vulnerabilities are remediated.**
* **CPS Team: Provides hardened AMIs for EKS/AKS nodes and baseline security configurations for cluster nodes.**

**High-Level Process Description:**

1. **Input Activities:**
   * **NEF publishes base container images for application teams.**
   * **CPS ensures hardened AMIs for nodes are compliant with baseline standards.**
   * **Security guidance is based on CSAM policies for EKS and AKS platforms.**
2. **Execution:**
   * **Containers are created using NEF-provided base images; however, these images are not aligned with industry standards (e.g., CIS or NIST benchmarks).**
   * **CI/CD pipelines integrate static vulnerability scans during the build phase and runtime scans during container operation.**
   * **Logging and monitoring systems (FluentD, Cribble, Splunk) collect container-level logs for analysis by the SoC.**
3. **Output:**
   * **Containerized workloads deployed on EKS/AKS clusters.**
   * **Security vulnerabilities addressed during scanning stages.**
   * **Logs monitored for security alerts to support incident detection and response.**

**2.2 Granular Risks and Key Controls Identified**

**Risk Identified: P1.R1**

* **Risk: Containers are not securely designed in alignment with industry standards such as CIS benchmarks or NIST guidelines.**
* **Relevance: The absence of adherence to recognized security standards increases the risk of misconfigurations and vulnerabilities in containerized environments. This could result in exposure to attacks that exploit insecure configurations.**

**Control Identified: P1.C1**

* **Control: Implement industry-standard baselines (e.g., CIS benchmarks, NIST guidelines) for container security configurations. Require application teams to build containers using NEF-provided hardened base images that meet these standards.**

**Risk Identified: P1.R2**

* **Risk: Limited governance and inconsistency in container configuration management due to the decentralized model in ECS workloads.**
* **Relevance: Inconsistent application of security standards can lead to exploitable security gaps across containerized applications in the organization.**

**Control Identified: P1.C2**

* **Control: Transition from a decentralized ECS model to a centralized governance approach. Establish mandatory pipelines with integrated security configurations and ensure all container configurations comply with industry standards.**

**Risk Identified: P1.R3**

* **Risk: Lack of detailed documentation and visibility into logging parameters for container workloads.**
* **Relevance: Undefined logging parameters may result in missed critical security events, impairing incident detection and response capabilities.**

**Control Identified: P1.C3**

* **Control: Define and standardize logging parameters for all containerized workloads. Ensure all logs are forwarded to Splunk and monitored by the SoC, with documented guidelines for effective log management.**

**Risk Identified: P1.R4**

* **Risk: Absence of periodic pen testing focused on containerized applications and pods.**
* **Relevance: Without regular assessments, vulnerabilities within containerized workloads may remain undetected and exploitable.**

**Control Identified: P1.C4**

* **Control: Conduct periodic pod-level pen tests to identify and mitigate vulnerabilities in containerized applications. Document and track remediation efforts to address findings.**

**Overview:**

**The conversation focused on understanding the end-to-end container lifecycle, deployment processes, and related security and operational controls**

**4o**

**You said:**

**seee the transcript and then give summary**

**ChatGPT said:**

**ChatGPT**

**Based on the transcript provided, here is the summary of the Zoom call regarding the container architecture rollout:**

**Summary of the Audit Process Call on Container Architecture Rollout**

**Participants:**

* **Stakeholders: Audit team, Container Team, Security Architect, and supporting teams.**

**Key Topics Discussed:**

**1. Container Lifecycle Overview:**

* **The lifecycle starts with creating base Docker images sourced from CPS or Artifactory.**
* **Configuration changes in images are made at the Dockerfile level, followed by a CI/CD pipeline process for image builds and deployment.**
* **Base images undergo vulnerability scanning using tools like Sneak and SonarQube, ensuring compliance before deployment.**

**2. Pipeline and Deployment Management:**

* **Pipelines in the Harness platform are used for continuous deployment.**
* **Projects are tied to specific clusters within landing zones, with configurations controlled centrally.**
* **Helm charts are standardized and control application-specific deployments; tenants can only modify limited parameters.**

**3. Cluster Configuration and Management:**

* **Clusters are created using Terraform and Jenkins pipelines with configurations stored in GitHub repositories.**
* **Resource allocation is based on standardized node sizes and limits (e.g., min size, max size).**
* **Clusters are auto-scaled based on deployment needs and workload demand.**
* **Nodes, pods, and containers are managed centrally by the containers team, with tenants having limited configuration access.**

**4. Security and Access Control:**

* **Security configurations are built into Helm charts and validated through the security architect's approval process.**
* **Tenants have read-only access to logs but cannot modify cluster-level configurations.**
* **Centralized tools and policies, such as Open Policy Agent (OPA), enforce runtime security and policy compliance.**

**5. Incident Management and Monitoring:**

* **Recent incidents were attributed to pipeline load issues due to throttling on interconnects.**
* **Monitoring tools and automation ensure configurations remain aligned with best practices.**

**6. Audit Scope and Next Steps:**

* **The audit will focus on end-to-end processes, from image creation to deployment.**
* **The team seeks documentation on processes, including pipeline configurations, Helm charts, and tenant-specific YAML files.**
* **Further walkthroughs and collaboration are planned to finalize the audit scope and identify key controls.**

**Key Takeaways:**

* **A robust process exists for container creation and deployment with clear controls and standardized configurations.**
* **Security and compliance are integral at each step, with limited scope for tenant-level deviations.**
* **The audit aims to ensure adherence to security, operational standards, and identify any potential risks.**