Multi-Tenant Secure Cloud Native Platform

There have been security concerns when deploying untrusted workloads using bare-metal containers, which utilize shared kernel from the host and only use cgroups and namespaces for isolation. Kata Containers addresses these concerns by using HW virtualization to isolate each container.

This additional security layer would allow the support of multiple tenant container workloads in one Kubernetes cluster.

A telco provider should be able to launch its trusted workloads (e.g. telemetry) using bare-metal containers. In the same Kubernetes cluster, customers would be allowed to bring their application workloads using Kata Containers.

This will eliminate the need of having multiple VM based Kubernetes clusters for different tenants or the need to assign different compute nodes for each tenant, which should improve the resource utilization of the environment.

Case Attributes	Description	Informational
Type	New	
Blueprint Family - Proposed Name	ICN	
Use Case	uCPE Edge Computing (described below)	
Blueprint proposed Name	Multi-Tenant Secure Cloud Native Platform	
Initial POD Cost (capex)	Same as ICN, no additional cost.	
Scale & Type	Same as ICN. Minimum of 4 Xeon Servers + 1 Xeon server as bootstrap node.	
Applications	Telco trusted workloads and customer untrusted workloads. E.g. SDEWAN, EDGX Foundry	
Power Restrictions	Same as ICN.	
Infrastructure orchestration	Bare Metal Provisioning Kubernetes provisioning : KuD. Centralized provisioning : Cluster-API + Provisioning controller (Explore Regional controller) Containerd for runc and Kata containers. Virtlet for VMs. Service Orchestration : EMCO MEC framework: OpenNESS Site orchestrator : Kubernetes upstream Traffic Orchestration within a cluster: ISTIO Traffic orchestration with external entities : ISTIO-ingress Knative for function orchestration	
SDN	OVN, Multus, Flannel	
Workload Type	Containers, VMs and functions. Manageability of Bare-metal containers for trusted workloads and Kata Containers (VM based) for untrusted workloads.	
Additional Details	Kata Containers should be deployable across existing Kubernetes clusters using containerd/cri. Kubernetes RuntimeClass (from k8s v1.14) and PodOverhead (from k8s v1.16) are features that allow Kata Containers to be selected, managed and monitored with existing Kubernetes tools. Kata Containers will not work when used with docker-shim runtime interface.	

Contributors	Intel: Adams, Eric (eric.adams@intel.com), Fuentes, Salvador (salvador.fuentes@intel.com), Shinde, Archana (archana.m.shinde@intel.com), Sterrett, Craig (craig.sterrett@intel.com) Verizon: Ravi (ravi.chunduru@verizon.com) Aarna Networks: Sandeep (ssharma@aarnanetworks.com), Sriram (srupanagunta@aarnanetworks.com)	

Use case: uCPE Edge Computing

Attributes	**Description**
Type	New
Industry Sector	Edge, Cloud, Enterprise, Telco
Business driver	There have been security concerns when deploying untrusted workloads using bare-metal containers, which utilize shared kernel from the host and only use cgroups and namespaces for isolation. Kata Containers addresses these concerns by using a lightweight VM to isolate each container. This additional security layer would allow the support of multiple tenant container workloads in one bare-metal Kubernetes cluster.
Business use cases	Kata Containers will add hard multi-tenancy capabilities. By adding Kata Containers and containerd into ICN, a Kubernetes cluster would be able to launch containers using both runtimes: runc for trusted workloads and Kata Containers for untrusted workloads. A telecommunications provider deploys CNFs such as SD-EWAN and NGFW under a Kubernetes Cluster in an Edge Location. The provider would like to deploy its trusted workloads using runc. A Customer of the provider deploys normal applications in the same Kubernetes cluster. Even with customer, there could be multiple departments deploying different workloads which would need isolation. For these different workloads, provider could allow to run them using Kata Containers in the same Kubernetes cluster. Using EMCO, which provides multitenancy capabilities, provider could define edge locations where Kata Containers is available and will use EMCO to redirect and isolate with respect to logical clouds and users.
Business Cost - Initial Build Cost Target Objective	Kata Containers should be able to run in existing infrastructure as long as it has hardware virtualization enabled. No additional cost from current ICN infrastructure.
Business Cost – Target Operational Objective	Same as ICN.
Security need	Kata Containers runs a lightweight VM, a minimal Kernel and a rootfs for launching containers. This provides more isolation against attacks from one untrusted container to other trusted/untrusted containers or to the host. This solution will allow the support of multiple tenant container workloads in a single bare-metal Kubernetes cluster.
Regulations	
Other restrictions	
Additional details	Kubernetes RuntimeClass (from k8s v1.14) and PodOverhead (from k8s) are features that allow Kata Containers to be selected, managed and monitored with existing Kubernetes tools.

Presentation:

Akraino ICN Mult...ve Platform.pptx

PTL & Committers:

Salvador Fuentes was elected PTL in Feb 2021

Committer	Committer Company	Committer Contact Info	Committer Bio	Committer Picture	Self Nominate for PTL (Y/N)
Salvador Fuentes	Intel	salvador.fuentes@intel.com	Salvador is the engineering manager for the Kata Containers project. Since he joined Intel in 2014, he has contributed to different open source projects for the cloud.		Y
Eric Adams	Intel	eric.adams@intel.com			
Archana Shinde	Intel	archana.m.shinde@intel.com			
Ravi Chunduru	Verizon	ravi.chunduru@verizon.com			
Amar Kapadia	Aarna Networks	akapadia@aarnanetworks.com			
Sandeep Sharma	Aarna Networks	ssharma@aarnanetworks.com			
Sriram Rupanagunta	Aarna Networks	srupanagunta@aarnanetworks.com			
Kuralamudhan Ramakrishnan	Intel	kuralamudhan.ramakrishnan@intel.com			
Todd Malsbary	Intel	todd.malsbary@intel.com			