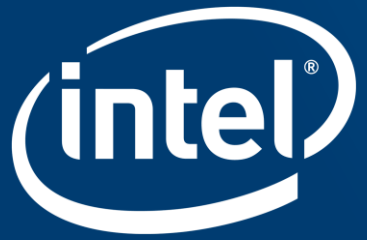


Cloudify policy-driven Orchestration for Multi-Cloud Services with Kubernetes-managed containerized F5 Networks functions

Shay Naeh, Senior Solution Architect, Cloudify
Philip Klatte, Senior Product Manager, F5 Networks
Petar Torre, Principal Engineer, Intel

14 October 2019

Enabling Telco Cloud in layered stack with Kubernetes

Orchestration Policy Driven for Multi-Cloud Services — CLOUDIFY

TOSCA Blueprint

Crypto

Apps

APACHE HTTP SERVER PROJECT Bench → HTTPS → f5 NGINX

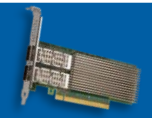
SW Platform

- Multiple network interfaces for VNFs
- High performance Data Plane (E-W, N-S)
- **Node Feature Discovery - platform capabilities**
- **Device Plugin (SR-IOV, Intel® QAT, FPGA…)**
- CPU Core-Pinning and isolation for K8s pods
- Huge Page allocation
- Topology Manager - Guarantee NUMA alignment
- Platform Telemetry
- Installation playbook

kubernetes, MULTUS, CNI, DPDK, GitHub, HELM, Ansible

Crypto

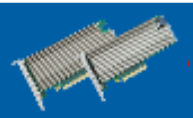
HW Platform



Intel® Ethernet Network Adapter

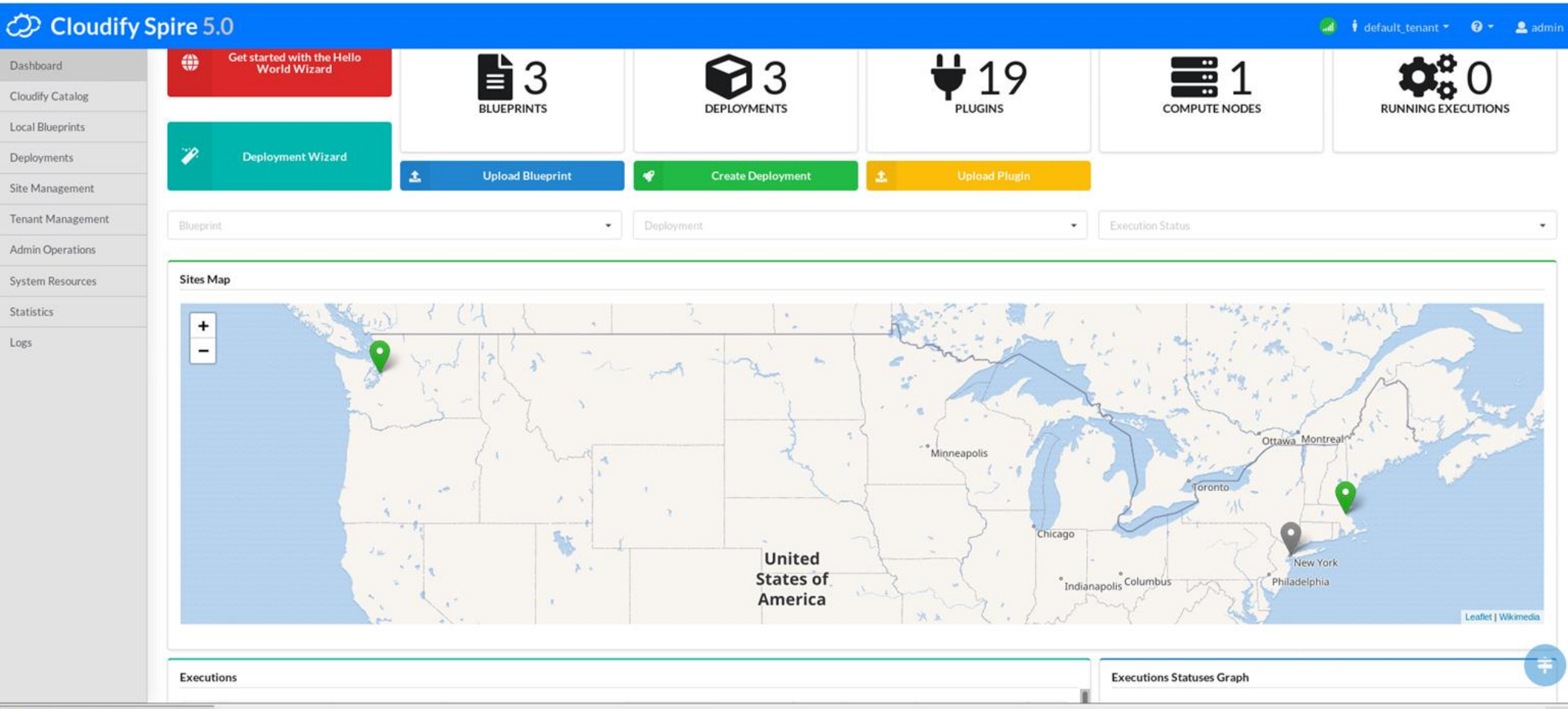
Intel® DC SSD

Intel® Programmable Acceleration Card

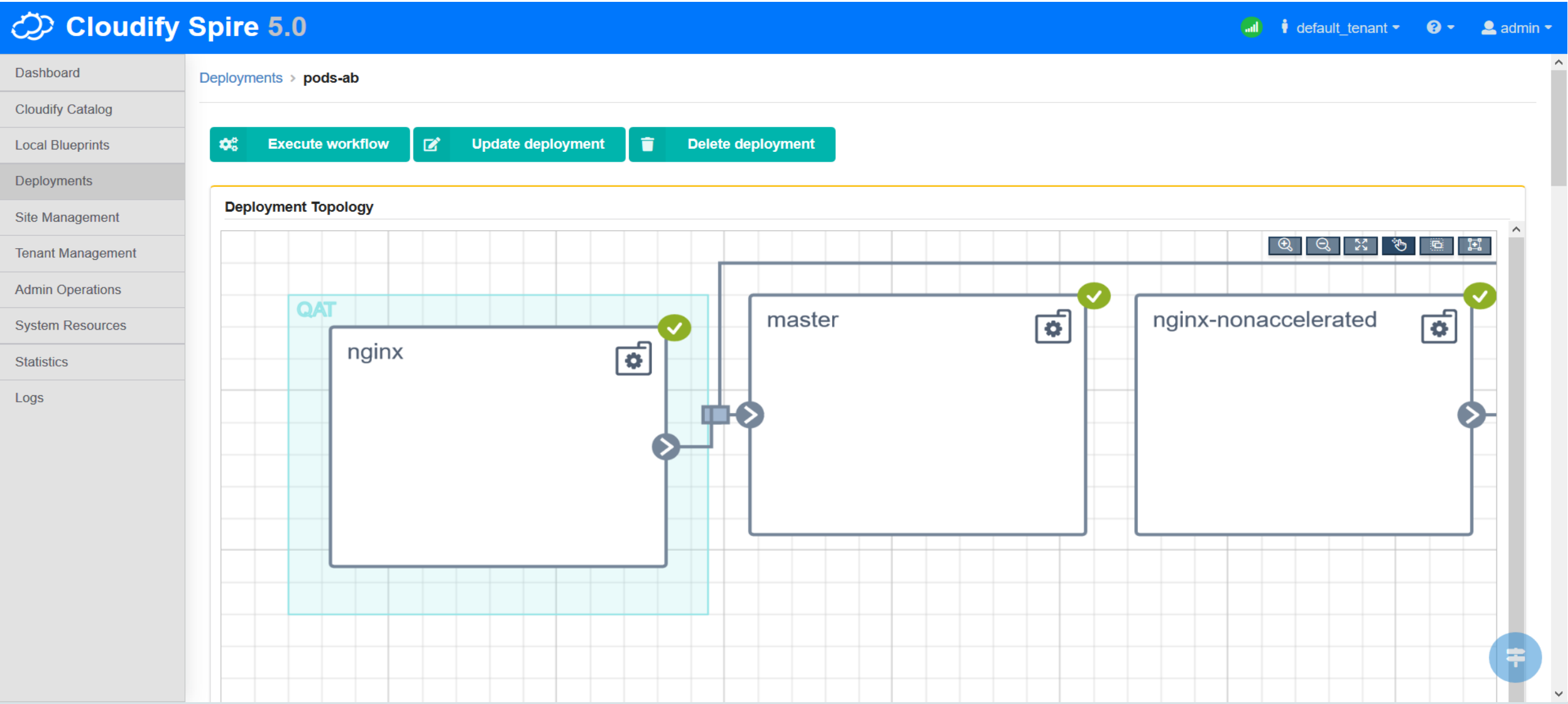


Intel® QuickAssist Technology

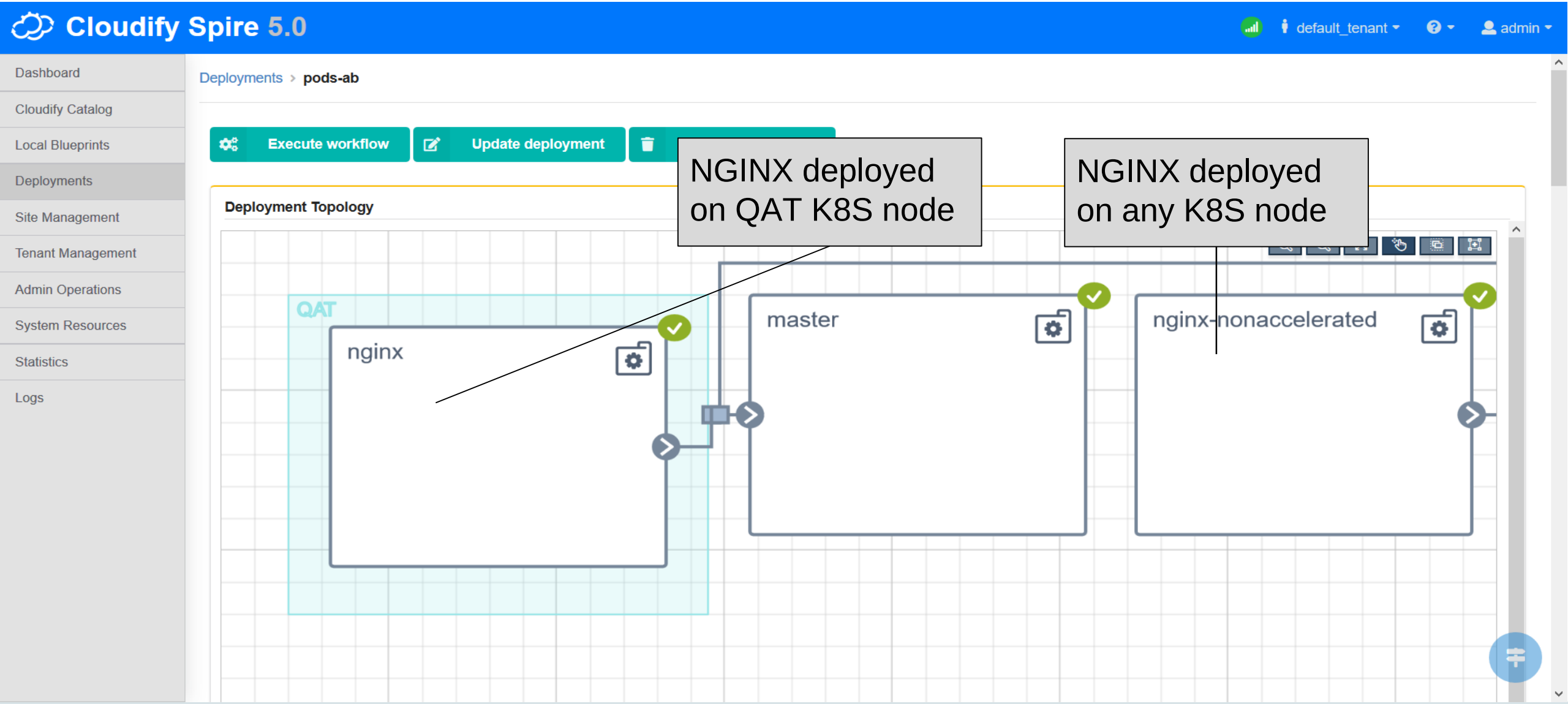
Cloudify Site Map



Deployments



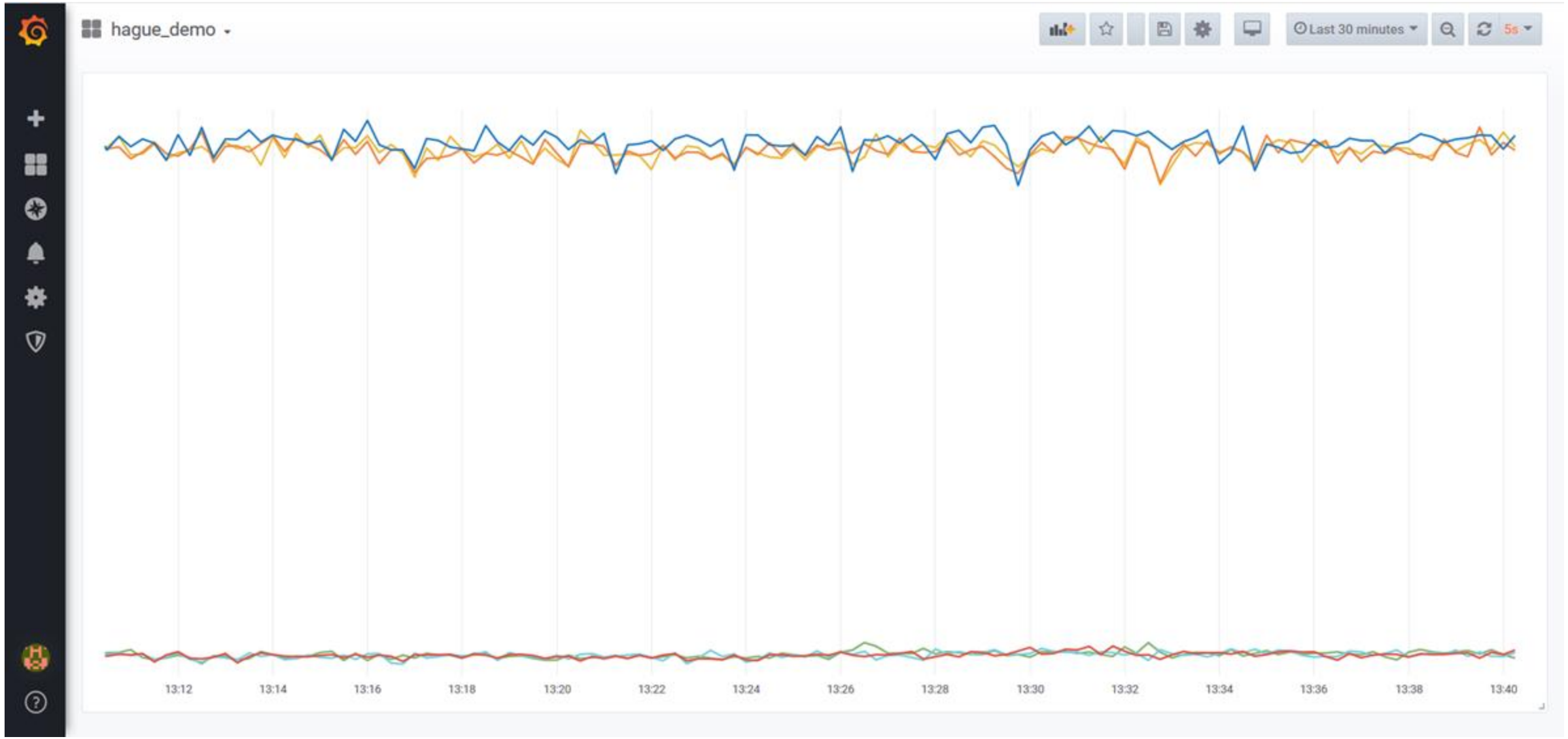
Placement with Intel® QuickAssist Technology (QAT)



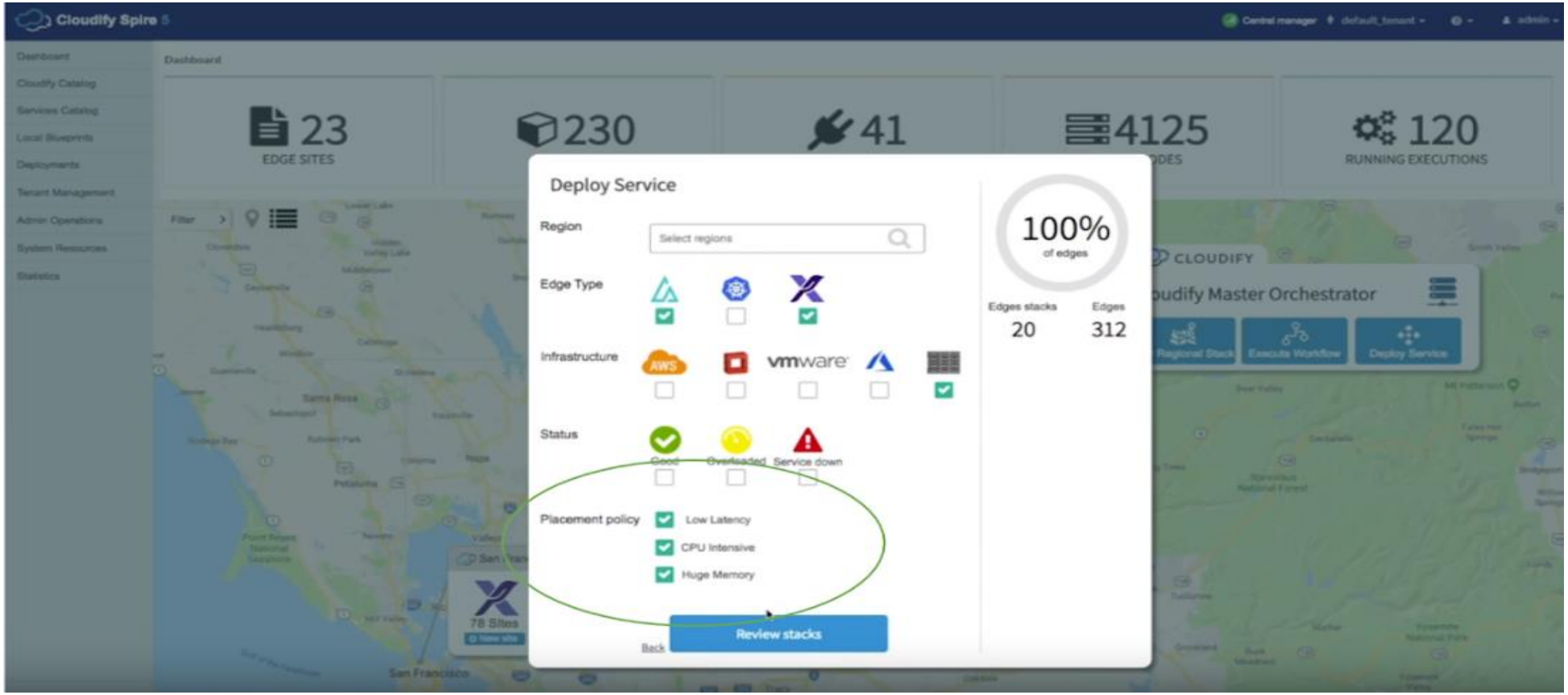
```
Listing labels for Node./node2:
 node.alpha.kubernetes-incubator.io/nfd-network-sriov=true
 node.alpha.kubernetes-incubator.io/nfd-cpuid-HTT=true
 node.alpha.kubernetes-incubator.io/nfd-cpuid-SSSE3=true
 node.alpha.kubernetes-incubator.io/nfd-cpuid-HLE=true
 node.alpha.kubernetes-incubator.io/nfd-cpuid-FMA3=true
 node.alpha.kubernetes-incubator.io/nfd-cpuid-AVX2=true
 kubernetes.io/hostname=node2
 node.alpha.kubernetes-incubator.io/nfd-cpuid-MMXEXT=true
 node.alpha.kubernetes-incubator.io/nfd-cpuid-AVX512DQ=true
 node.alpha.kubernetes-incubator.io/nfd-cpuid-MMX=true
 node.alpha.kubernetes-incubator.io/nfd-rdt-RDTMBA=true
 node.alpha.kubernetes-incubator.io/nfd-cpuid-RTM=true
 qat=true
 node.alpha.kubernetes-incubator.io/nfd-cpuid-F16C=true
 node.alpha.kubernetes-incubator.io/nfd-cpuid-BMI2=true
 node.alpha.kubernetes-incubator.io/nfd-rdt-RDTCMT=true
 node.alpha.kubernetes-incubator.io/nfd-cpuid-ADX=true
 node-role.kubernetes.io/node=
 node.alpha.kubernetes-incubator.io/nfd-memory-numa=true
 node.alpha.kubernetes-incubator.io/nfd-rdt-RDTL3CA=true
 node.alpha.kubernetes-incubator.io/nfd-cpuid-SSE=true
 node.alpha.kubernetes-incubator.io/nfd-rdt-RDTMON=true
 beta.kubernetes.io/arch=amd64
 node.alpha.kubernetes-incubator.io/nfd-cpuid-NX=true
 node.alpha.kubernetes-incubator.io/nfd-cpuid-SSE4.2=true
 node.alpha.kubernetes-incubator.io/nfd-pstate-turbo=true
 node.alpha.kubernetes-incubator.io/nfd-rdt-RDTMBM=true
 node.alpha.kubernetes-incubator.io/nfd-cpuid-RDTSCP=true
 node.alpha.kubernetes-incubator.io/nfd-iommu-enabled=true
 node.alpha.kubernetes-incubator.io/nfd-cpuid-LZCNT=true
 node.alpha.kubernetes-incubator.io/nfd-cpuid-SSE2=true
 node.alpha.kubernetes-incubator.io/nfd-cpuid-AVX512VL=true
 beta.kubernetes.io/os=linux
 node.alpha.kubernetes-incubator.io/nfd-cpuid-AESNI=true
 node.alpha.kubernetes-incubator.io/nfd-cpuid-AVX512CD=true
 node.alpha.kubernetes-incubator.io/nfd-storage-nonrotationaldisk=true
 node.alpha.kubernetes-incubator.io/node-feature-discovery.version=v0.3.0
```

```
[root@master1 ~]# kubectl describe pod demo-g2xln
Name:               demo-g2xln
Namespace:          default
Priority:           0
PriorityClassName:  <none>
Node:               node2/192.168.0.237
Start Time:         Thu, 10 Oct 2019 13:13:23 +0300
Labels:             <none>
Annotations:        k8s.v1.cni.cncf.io/networks-status:
                      [{
                          "name": "cni0",
                          "interface": "eth0",
                          "ips": [
                              "10.244.2.23"
                          ],
                          "mac": "0a:58:0a:f4:02:17",
                          "default": true,
                          "dns": {}
                      }]
Status:             Running
IP:                 10.244.2.23
Containers:
  demonginx:
Conditions:
  Type              Status
  Initialized       True
  Ready             True
  ContainersReady   True
  PodScheduled      True
Volumes:
  default-token-mljgc:
    Type:        Secret (a volume populated by a Secret)
    SecretName:  default-token-mljgc
    Optional:    false
QoS Class:       Burstable
Node-Selectors:  qat=true
Tolerations:     node.kubernetes.io/not-ready:NoExecute for 300s
                 node.kubernetes.io/unreachable:NoExecute for 300s
Events:          <none>
```

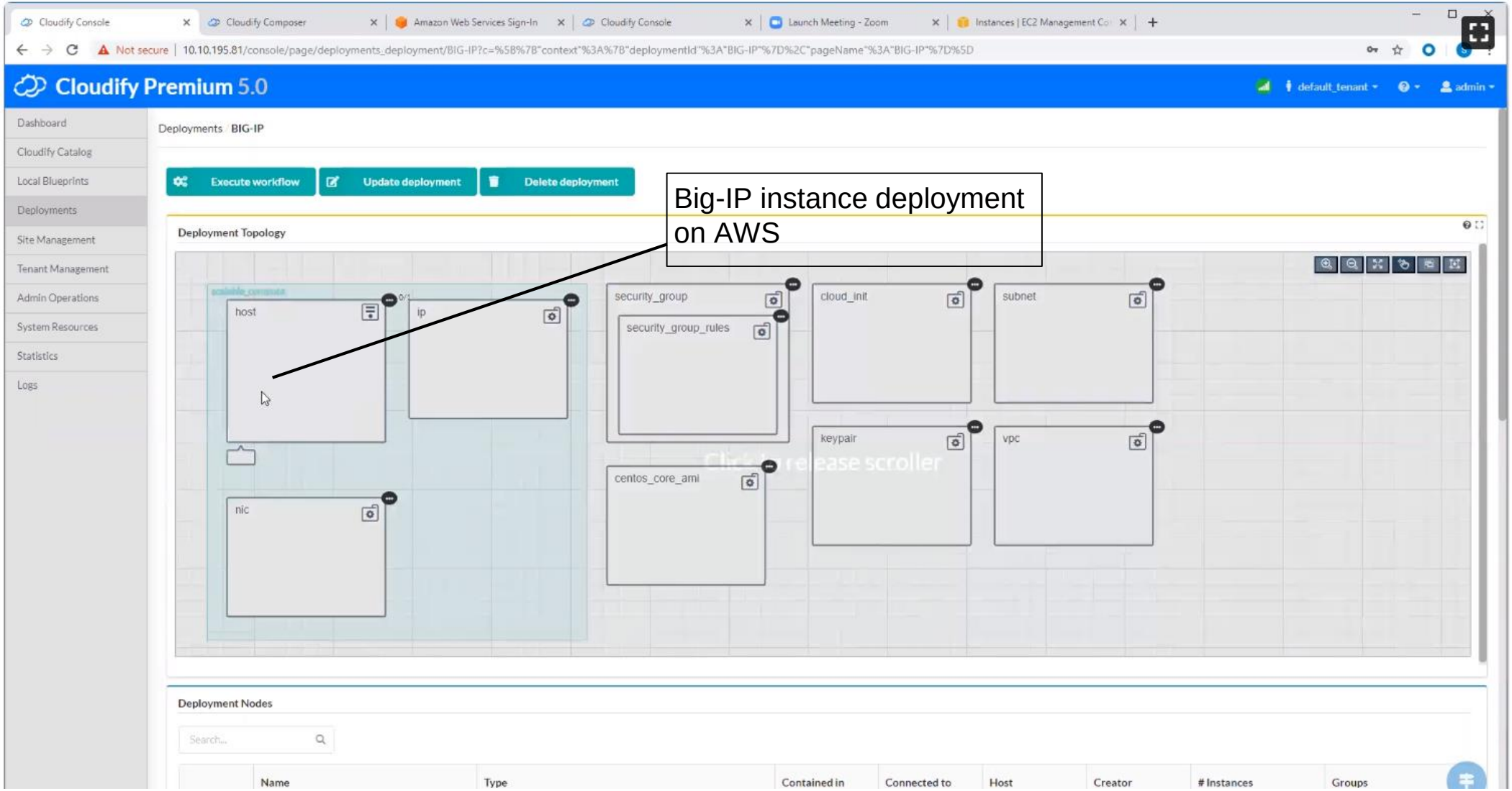
Results


hague_demo
Last 30 minutes
5s
13:12
13:14
13:16
13:18
13:20
13:22
13:24
13:26
13:28
13:30
13:32
13:34
13:36
13:38
13:40

Intent based placement



Cloud Instances – F5 BIG IP on AWS



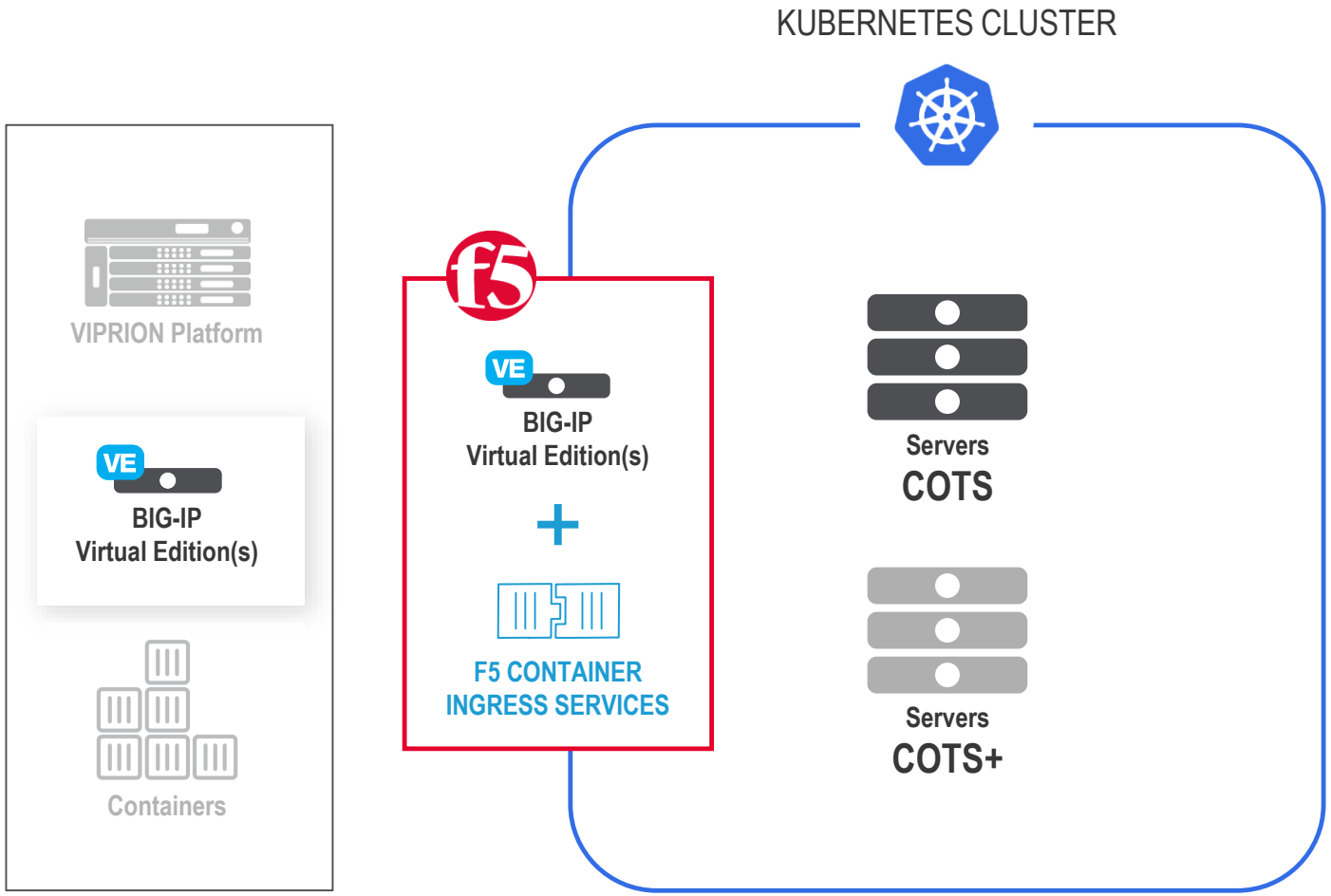
F5 Use Cases: Initial Demo Setup

Cloudify Orchestration Leverages Node Feature Discovery to Optimize NGINX Service Placement

Intel® EPA and Cloudify® Orchestration – Setup

Ingress examples

Security (PAC, DDOS)
Encrypt/Decrypt
Traffic steering



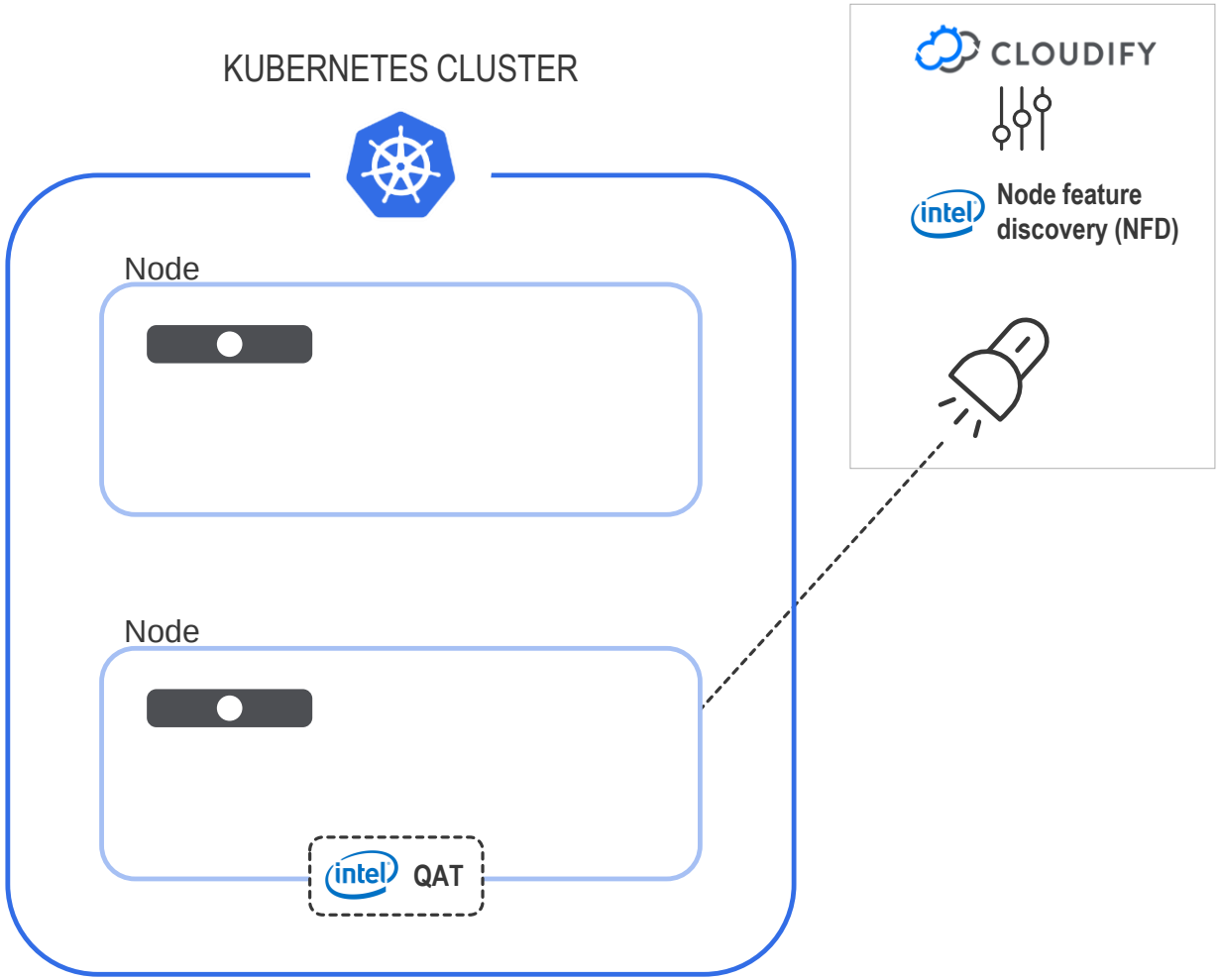
F5 Container Ingress Service can be deployed using virtual or physical models. For the demo, we used BIG-IP Virtual Edition.

Intel® EPA and Cloudify® Orchestration – Setup

Ingress examples

Security (PAC, DDOS)
Encrypt/Decrypt
Traffic steering

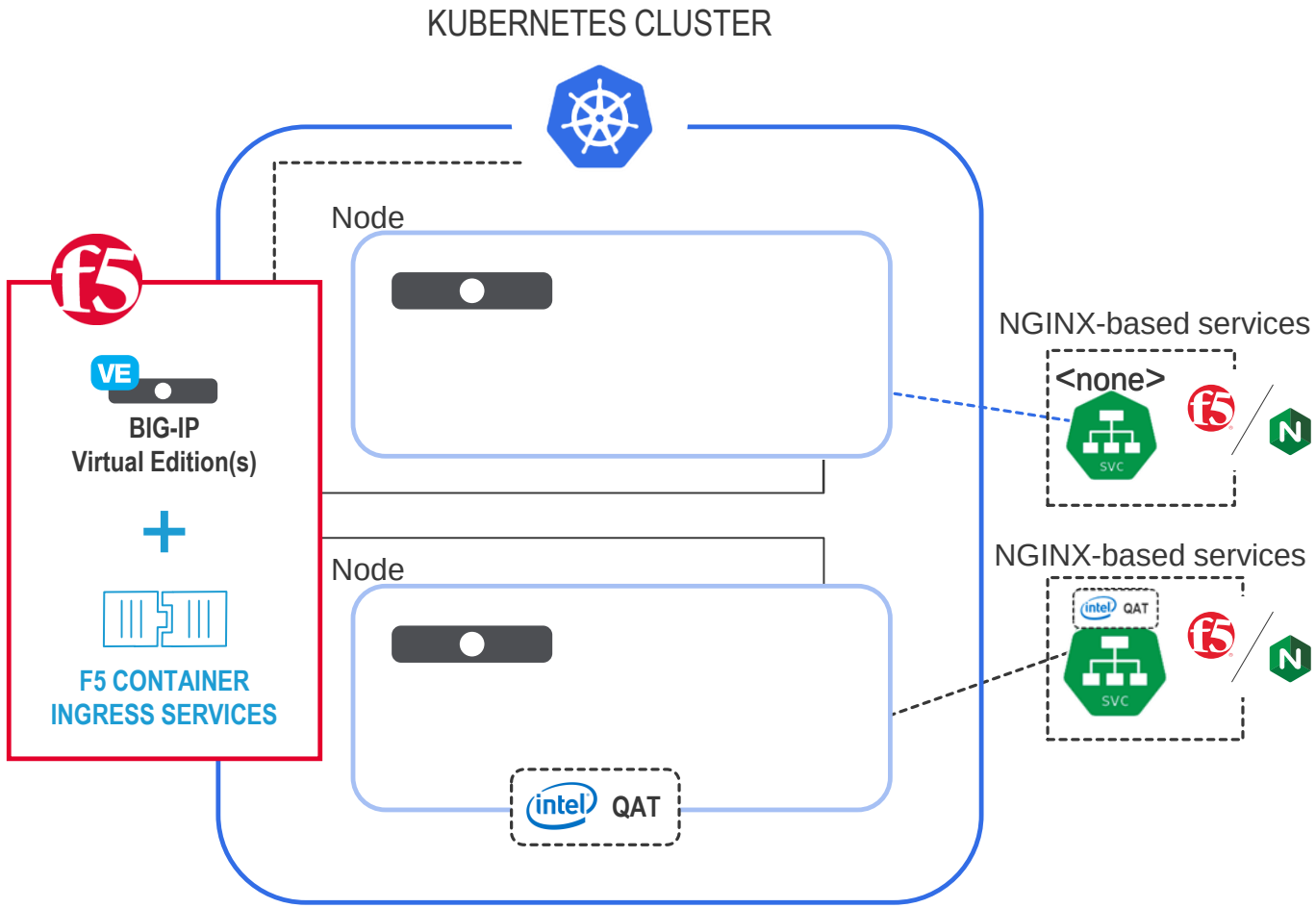
Cloudify orchestrates labeling nodes, using Intel Node Feature Discovery ex. Intel® QuickAssist Technology, (Intel® QAT)



Intel® EPA and Cloudify® Orchestration – Setup

Ingress examples

Security (PAC, DDOS)
Encrypt/Decrypt
Traffic steering



Pods are deployed based on node affinity, with some optimized for Intel® QAT offload.

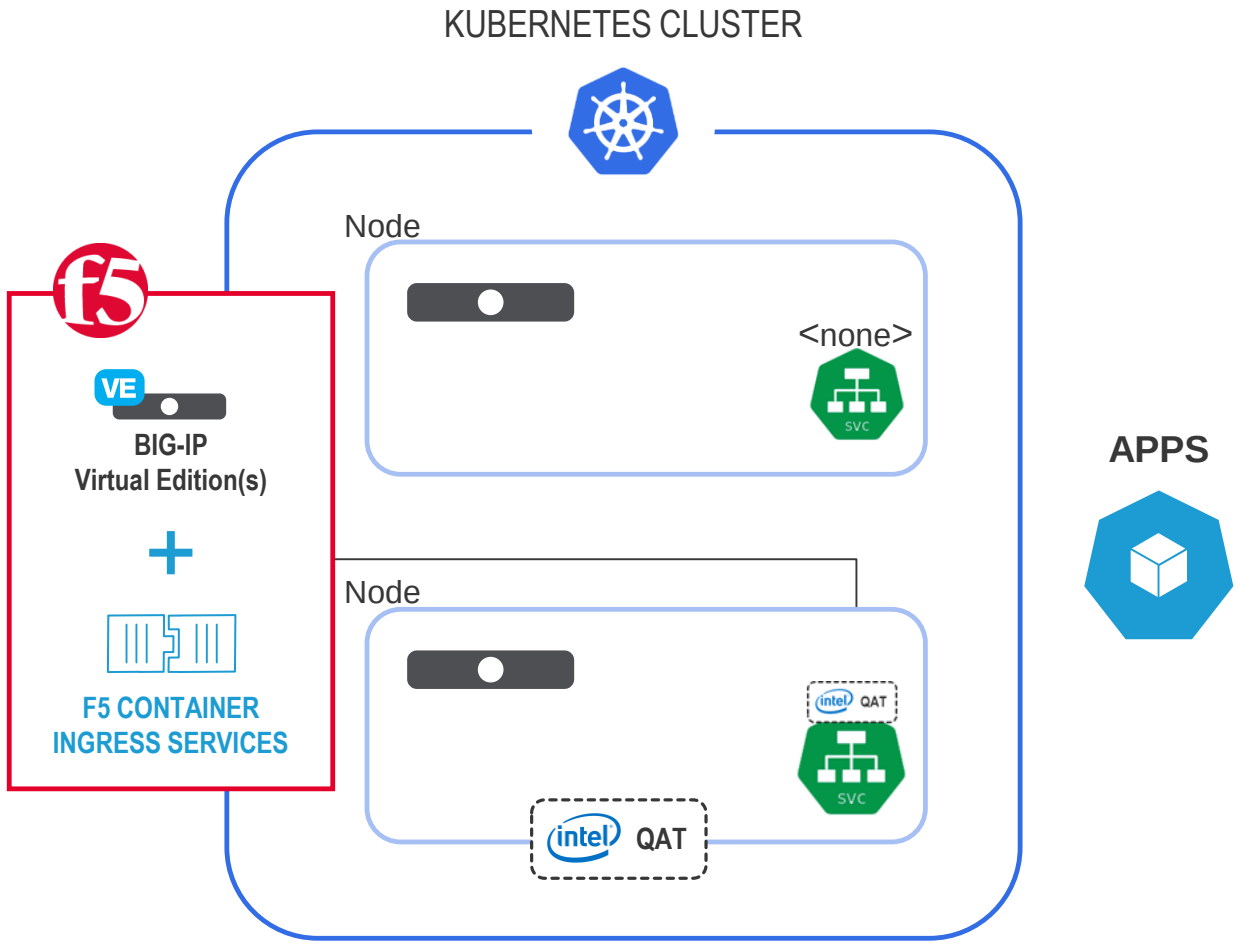
F5 CIS establishes connections to the new services, utilizing the K8s API.

Intel® EPA and Cloudify® Orchestration – Setup

Ingress examples

Security (PAC, DDOS)
Encrypt/Decrypt
Traffic steering

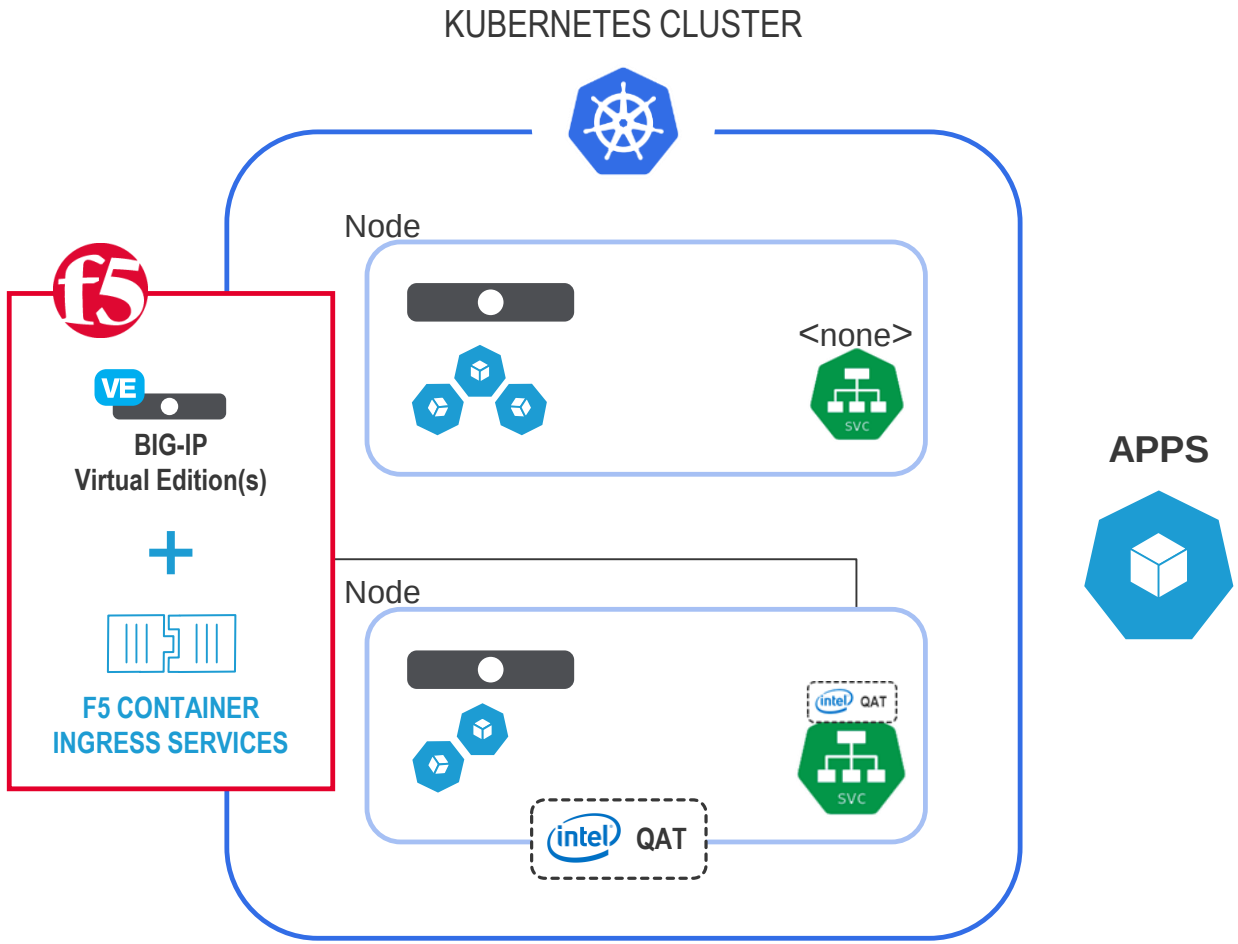
Backend application pods are deployed with no particular affinity.



Intel® EPA and Cloudify® Orchestration – Setup

Ingress examples

Security (PAC, DDOS)
Encrypt/Decrypt
Traffic steering



Services have self-sorted to be on optimized nodes.

Now the cluster is ready for traffic.

F5 Use Cases: Ingress security

F5 Container Ingress Services Leveraging Intel® Programmable Acceleration Card (PAC) to Optimize Security

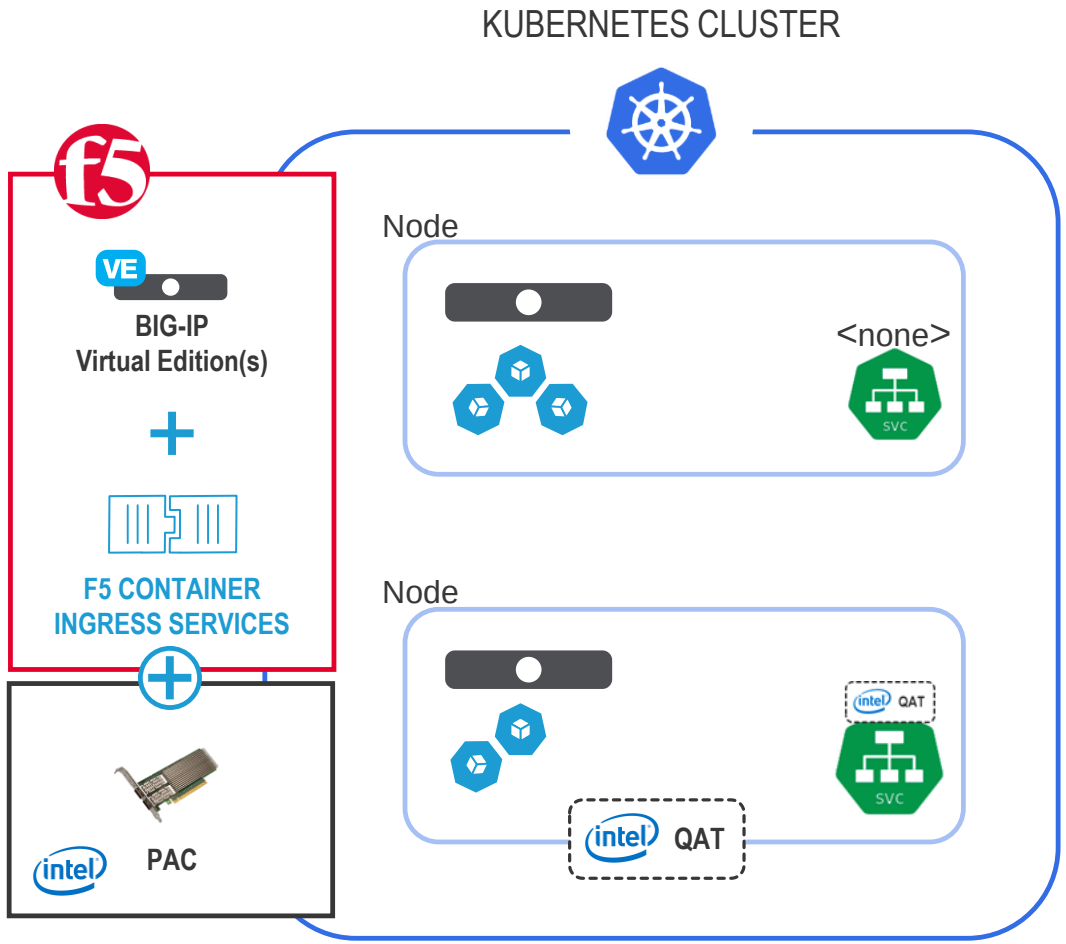
Intel® EPA and Cloudify® Orchestration – Setup

Ingress examples

Security (PAC, DDOS)

Encrypt/Decrypt

Traffic steering



In this scenario, we are using BIG-IP Virtual Edition on a node with a SmartNIC installed.

Intel® EPA and Cloudify® Orchestration – Setup

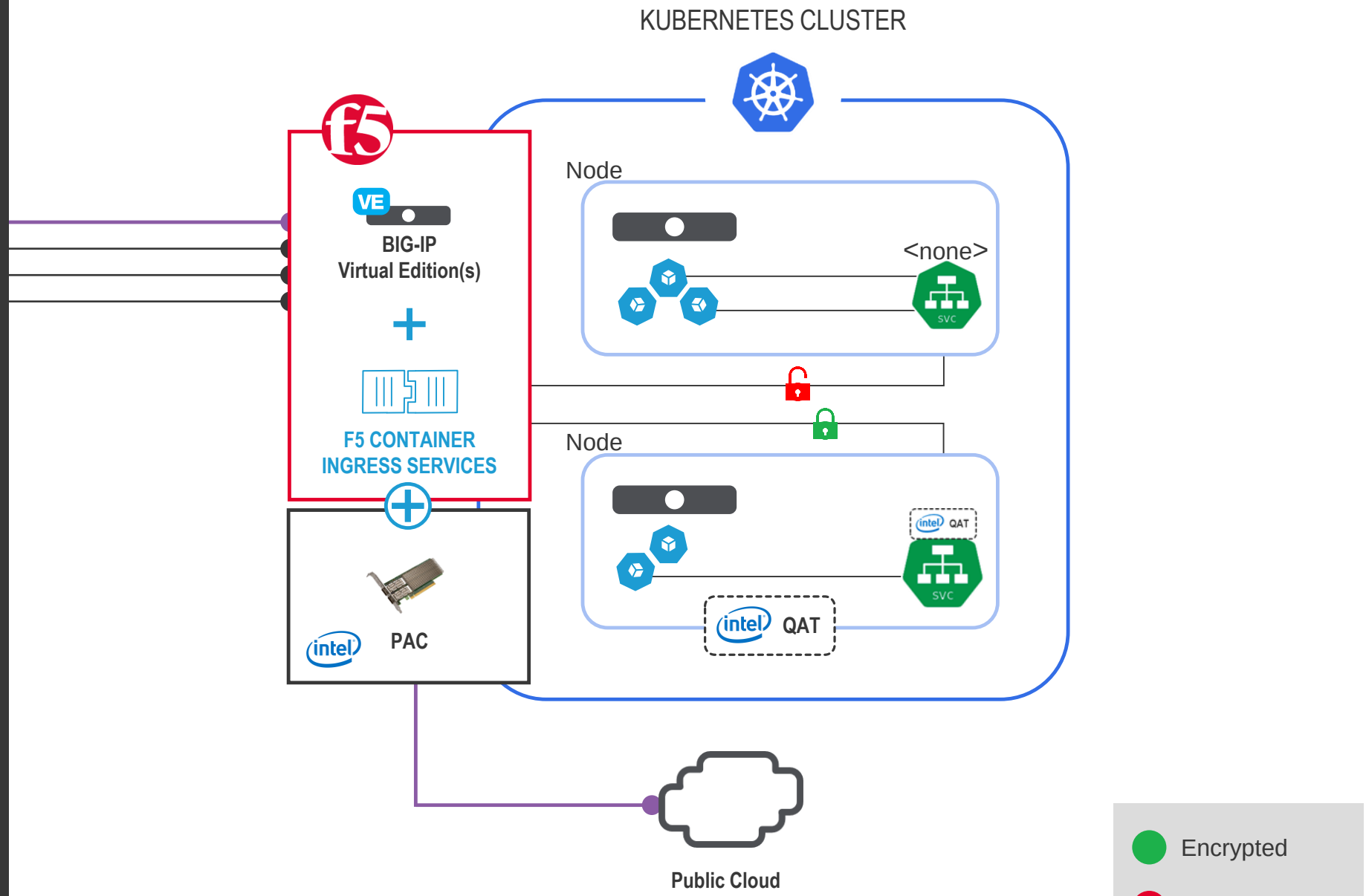
Ingress examples

Security (PAC, DDOS)

Encrypt/Decrypt

Traffic steering

Legitimate application traffic arrives and is directed to the correct service in K8s or in another cloud.



Intel® EPA and Cloudify® Orchestration – Setup

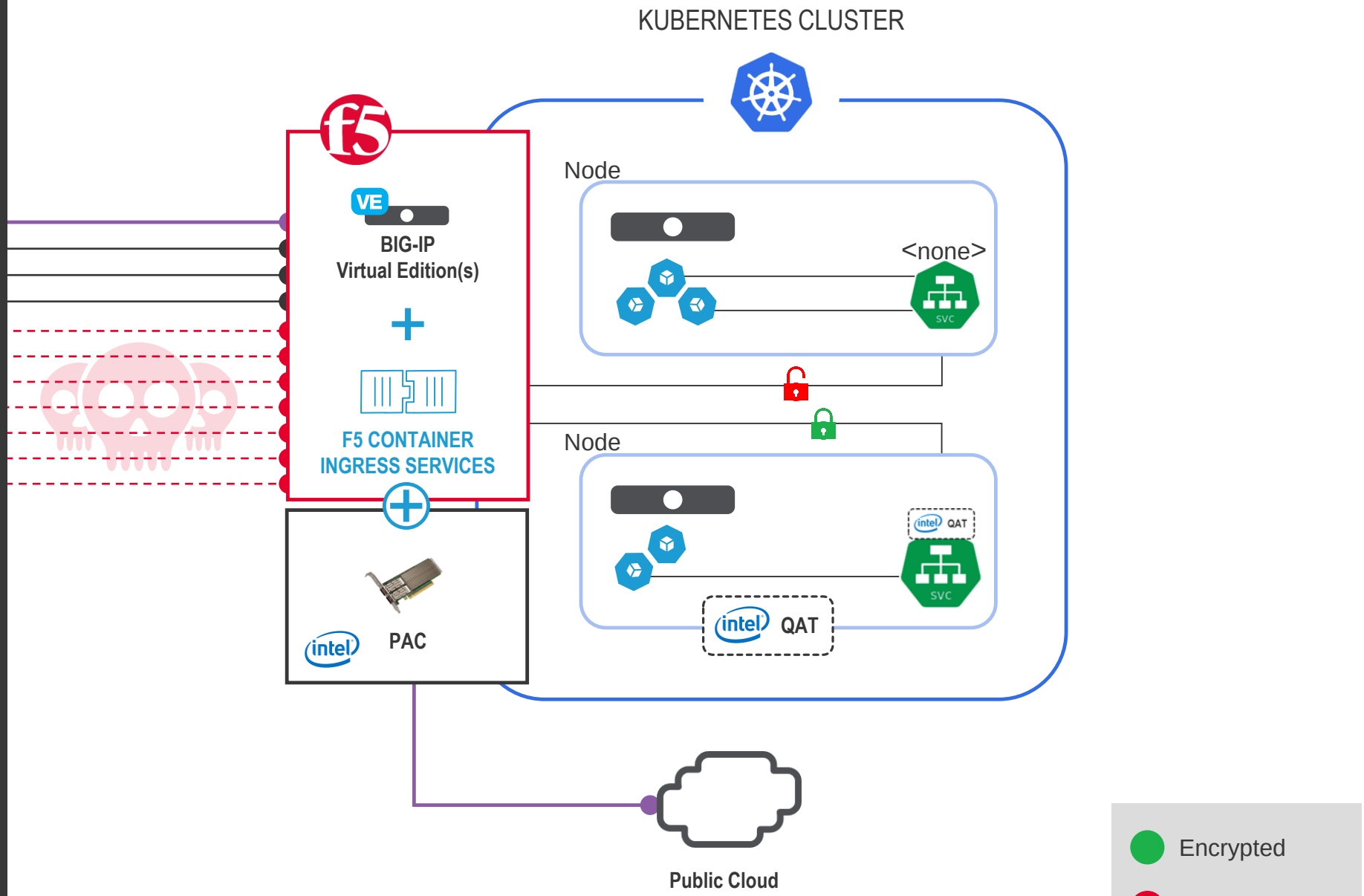
Ingress examples

Security (PAC, DDOS)

Encrypt/Decrypt

Traffic steering

A volumetric/DDoS attack arrives and is identified as an attack.



Intel® EPA and Cloudify® Orchestration – Setup

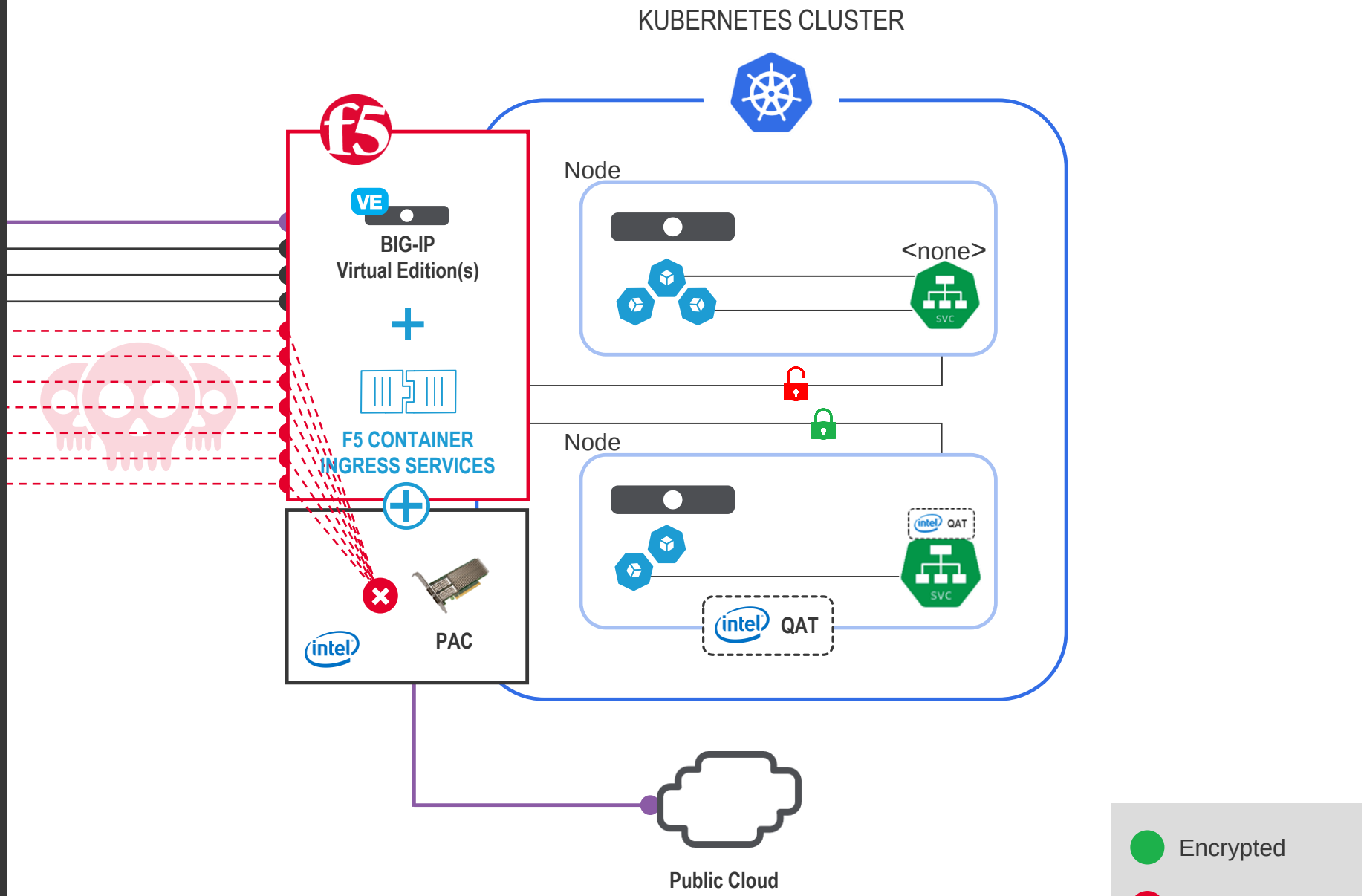
Ingress examples

Security (PAC, DDOS)

Encrypt/Decrypt

Traffic steering

The attack traffic is accelerated onto the Intel® Programmable Acceleration Card (PAC) where it is either dropped or redirected to a scrubber



Intel® EPA and Cloudify® Orchestration – Setup

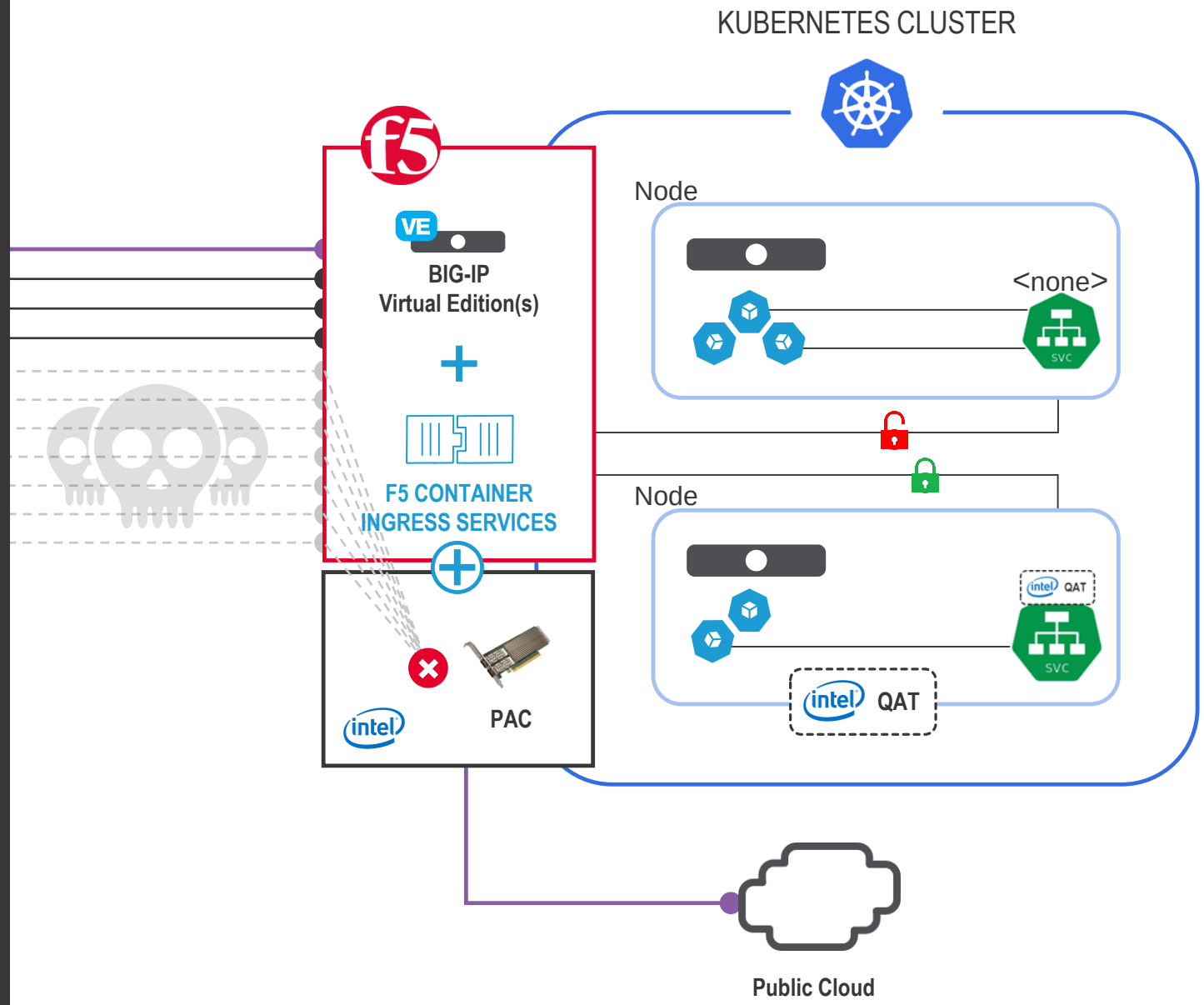
Ingress examples

Security (PAC, DDOS)

Encrypt/Decrypt

Traffic steering

Normal traffic continues to flow.



F5 Use Cases: Ingress encryption/decryption offload

F5 Container Ingress Services Leveraging Intel® Quick Assist Technology (QAT) to Optimize Encryption and Decryption

K8s Node Labels

COTS/COTS+ matchmaking

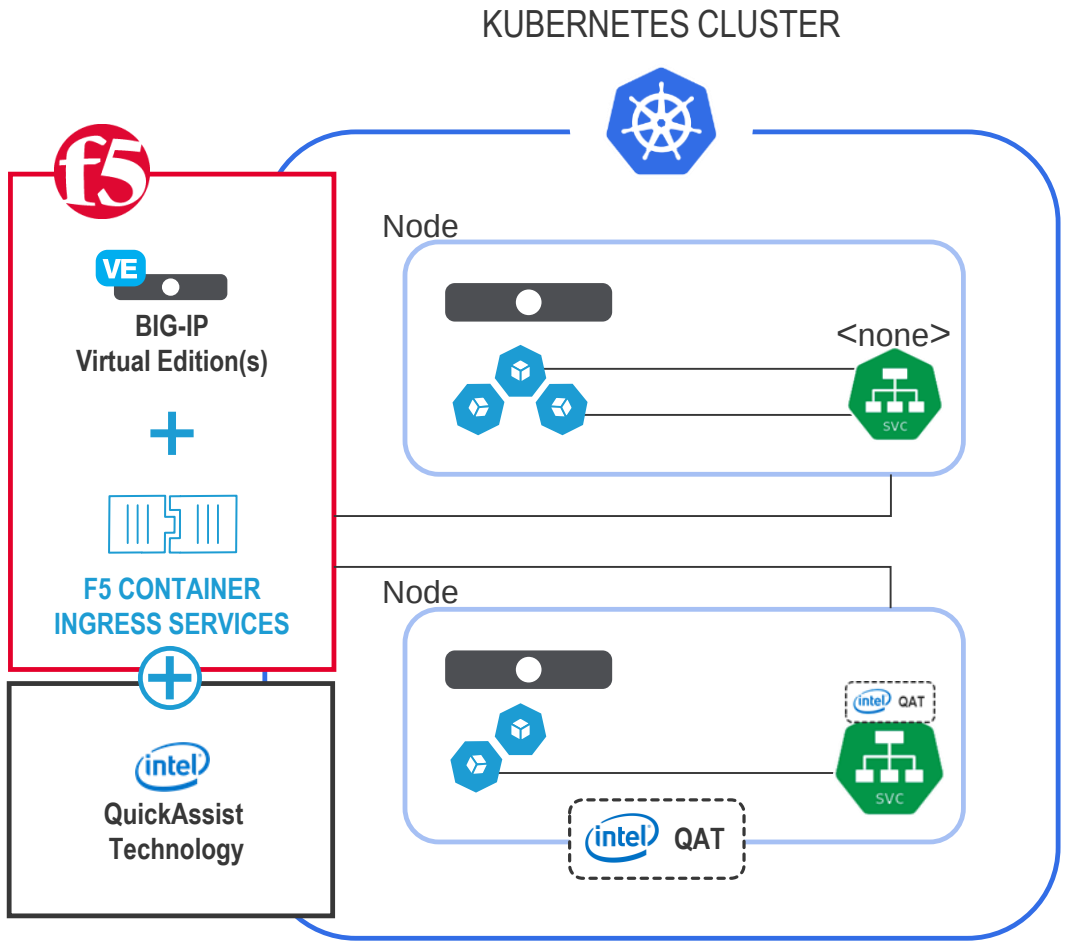
Ingress examples

Security (PAC, DDOS)

Encrypt/Decrypt

Traffic steering

Deployment: virtual F5 BIG-IP running on server with Intel® QuickAssist Technology (QAT)



Encrypted

Unencrypted

K8s Node Labels

COTS/COTS+ matchmaking

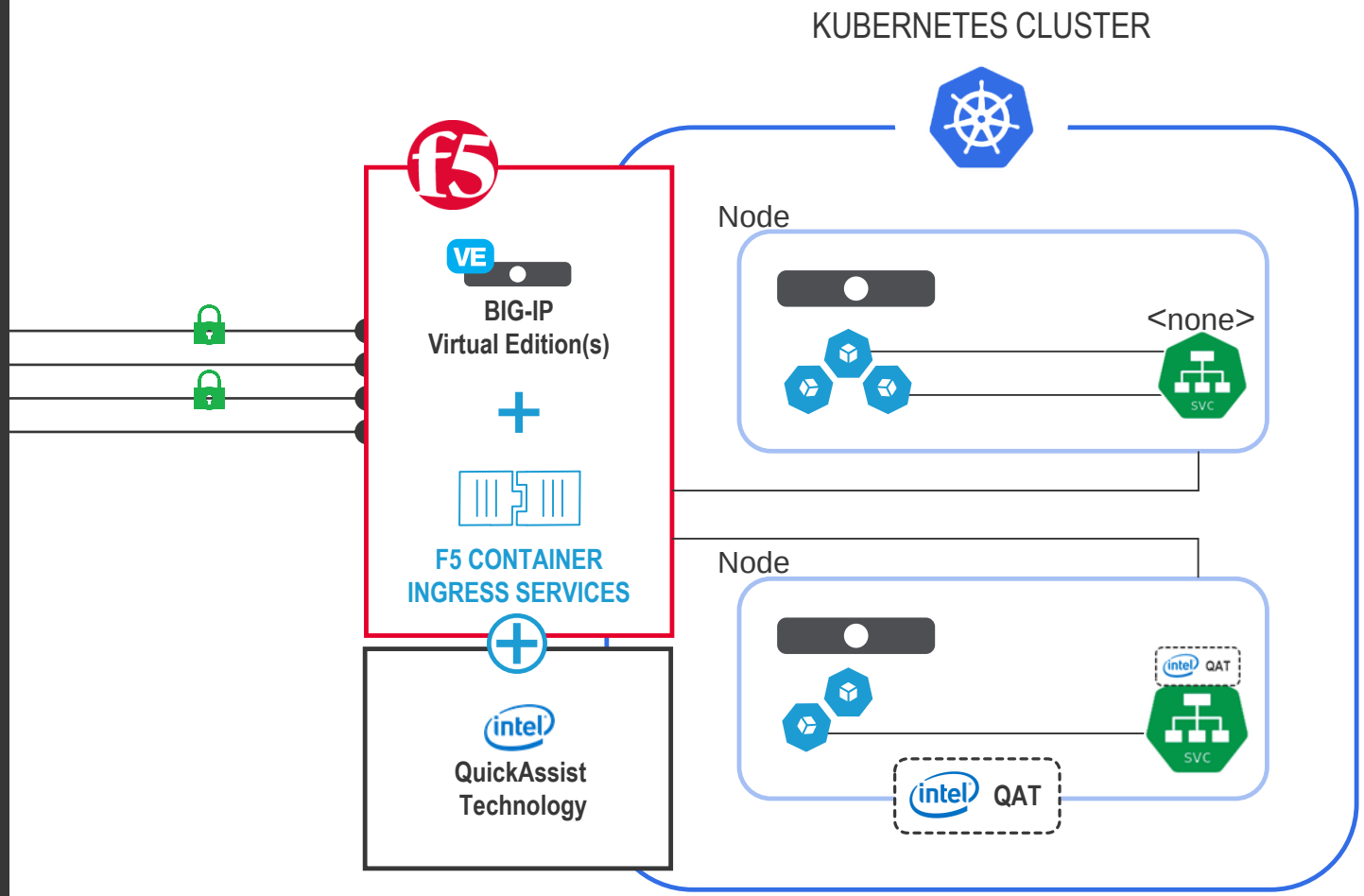
Ingress examples

Security (PAC, DDOS)

Encrypt/Decrypt

Traffic steering

Traffic decrypted using Intel® QAT offload



Encrypted

Unencrypted

K8s Node Labels

COTS/COTS+ matchmaking

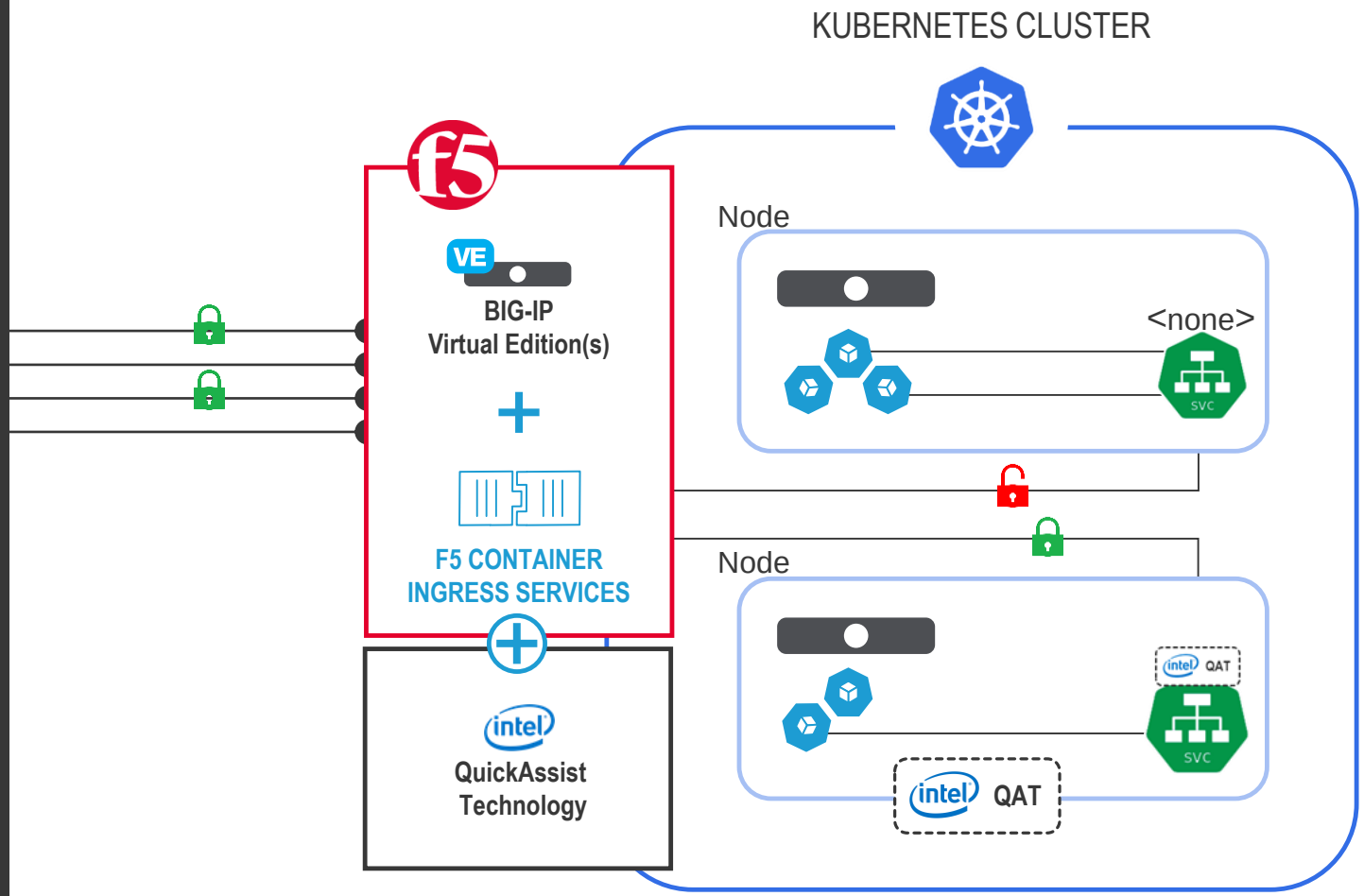
Ingress examples

Security (PAC, DDOS)

Encrypt/Decrypt

Traffic steering

Unencrypted and re-encrypted traffic sent to services within K8s



Encrypted

Unencrypted

K8s Node Labels

COTS/COTS+ matchmaking

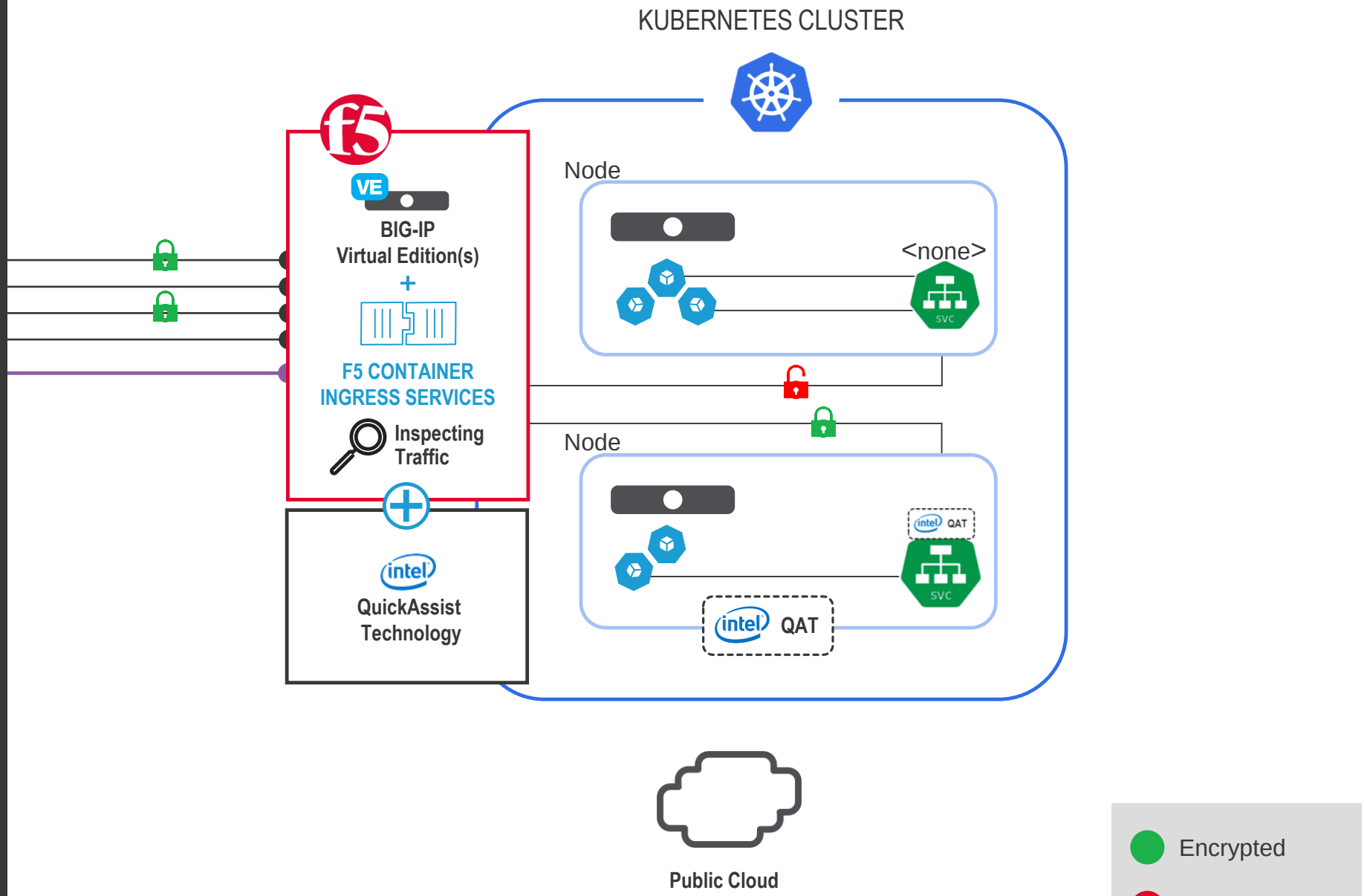
Ingress examples

Security (PAC, DDOS)

Encrypt/Decrypt

Traffic steering

Decrypted traffic identified for steering to another destination in Public Cloud



K8s Node Labels

COTS/COTS+ matchmaking

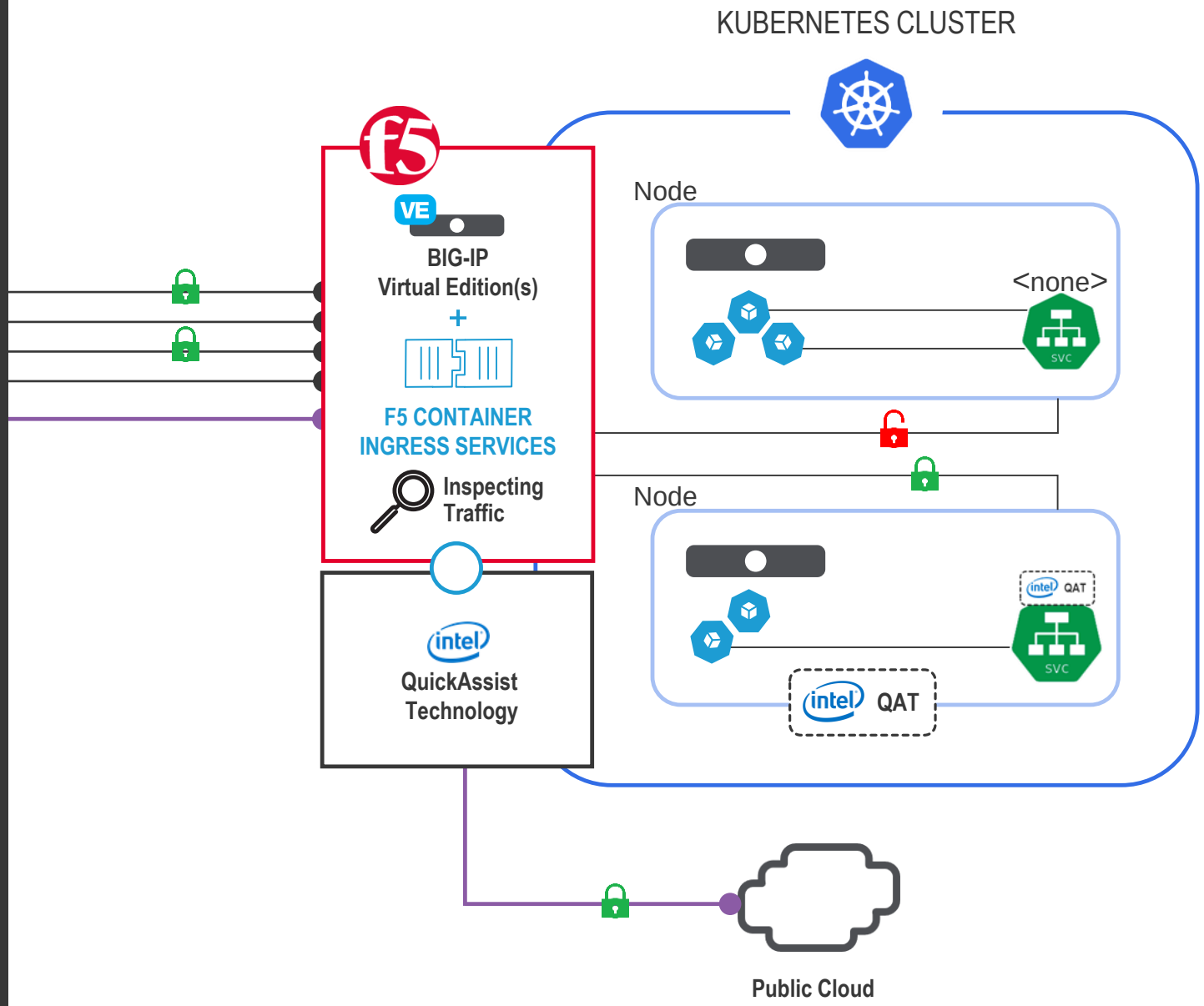
Ingress examples

Security (PAC, DDOS)

Encrypt/Decrypt

Traffic steering

Traffic re-encrypted (using Intel® QAT offload) and sent to another destination in Public Cloud



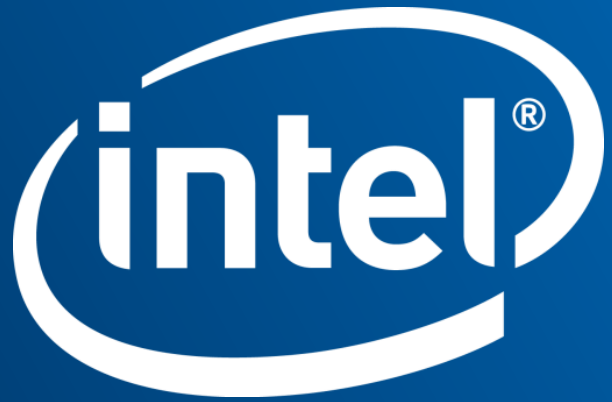
Visit us to see demonstrations at:

Intel booth B14

F5 Networks booth C6

Legal Disclaimers

- Copyright © Intel Corporation. All rights reserved.
- Other names and brands may be claimed as the property of others.
- Intel, the Intel logo and other Intel Marks are trademarks of Intel Corporation or its subsidiaries in the U.S. and/or other countries.
- No license (express or implied, by estoppel or otherwise) to any intellectual property rights is granted by this document.
- Intel disclaims all express and implied warranties, including without limitation, the implied warranties of merchantability, fitness for a particular purpose, and non-infringement, as well as any warranty arising from course of performance, course of dealing, or usage in trade.


intel