AWS re:Invent

NOV. 28 – DEC. 2, 2022 | LAS VEGAS, NV

CMP301

Powering Amazon EC2: Deep dive on the AWS Nitro System

Ravi Murty

Sr. Principal Engineer
EC2

Agenda

Nitro Overview

Continuous innovation

Security

Amazon EC2 bare metal instances

Nitro: Five years later

AWS Nitro

Launched in November 2017

In development since 2013

Purpose-built hardware and software

Hypervisor built for AWS

All modern EC2 instances use the Nitro system

Nitro system in three parts

Nitro Cards

VPC Networking

Amazon Elastic Block Store

(Amazon EBS)

Instance Storage

System Controller

Nitro Security Chip

Integrated into motherboard

Protects hardware resources

Hardware Root of Trust

Nitro Hypervisor

Lightweight hypervisor

Memory and CPU allocation

Bare Metal-like performance

Nitro Cards

ENA PCIe Controller

VPC Data Plane

Amazon VPC

NVMe PCIe Controller

EBS Data Plane

Amazon Elastic Block Store

NVMe PCIe Controller

Transparent Encryption

Instance Storage

System Control

Root of Trust

Nitro Control

Nitro Card for VPC

Amazon VPC

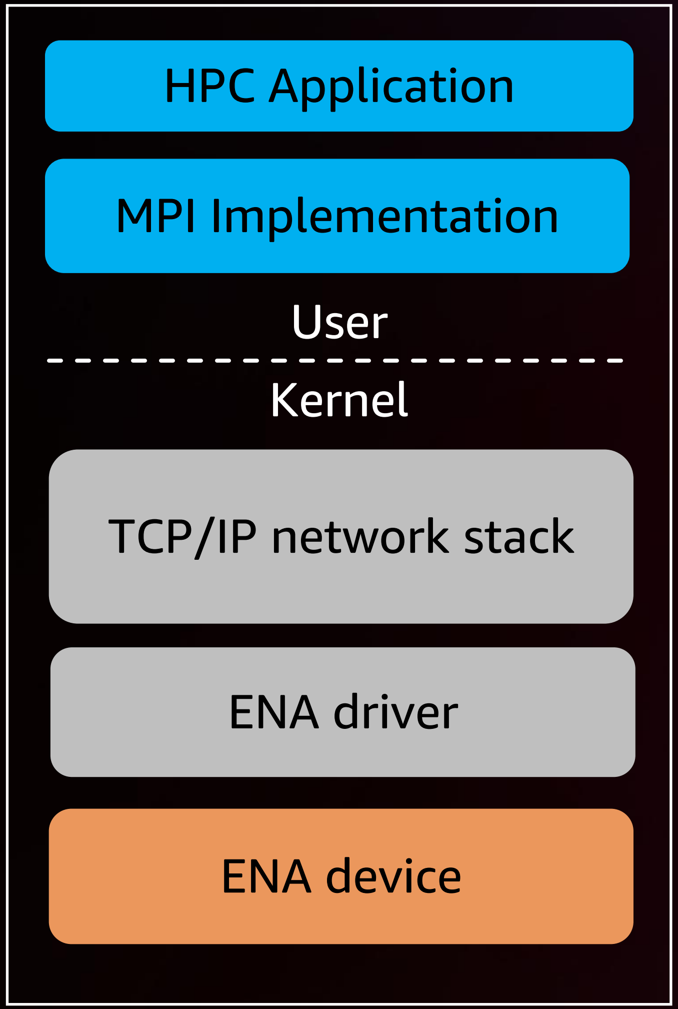
ENA Controller

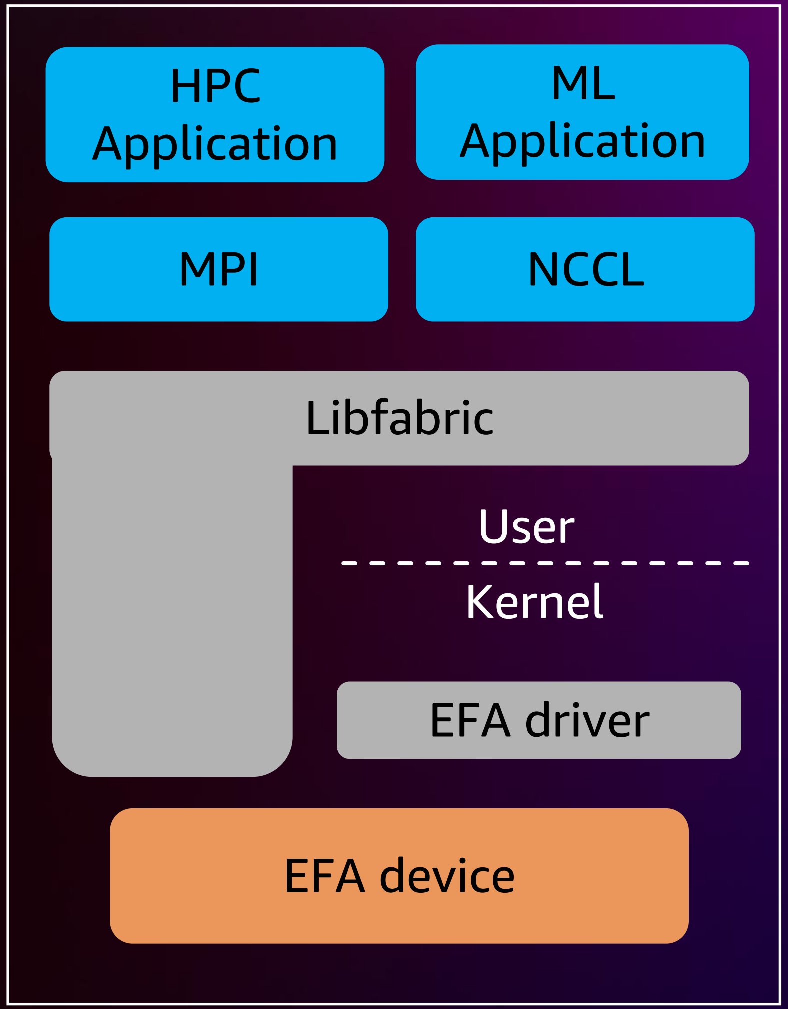
- Drivers available for all major operating systems
- Independent of fabric

VPC Data Plane

- Encapsulation
- Security Groups
- Limiters
- Routing

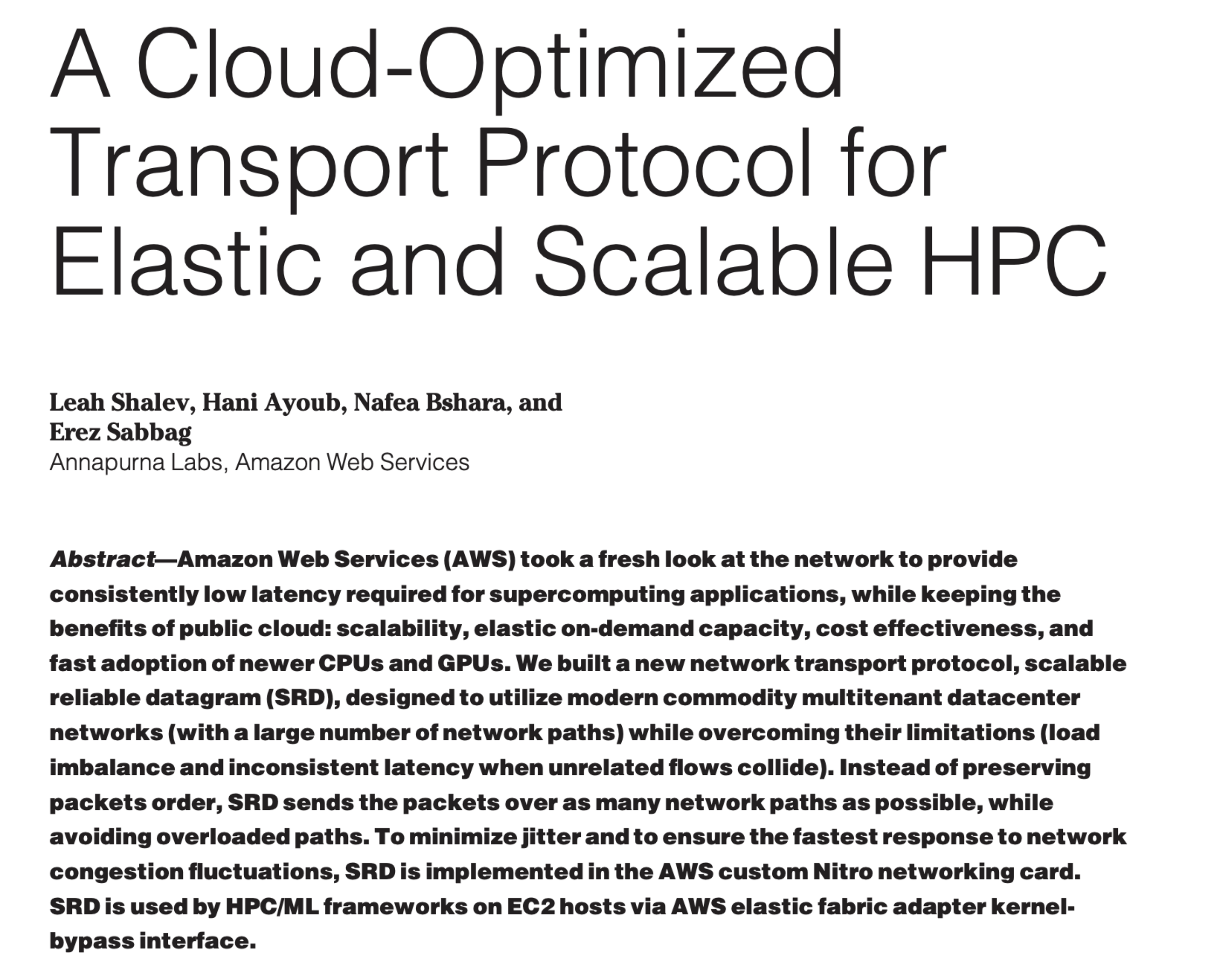
Elastic Fabric Adapter (EFA)





SRD: https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=9167399

Scalable Reliable Datagram (SRD)



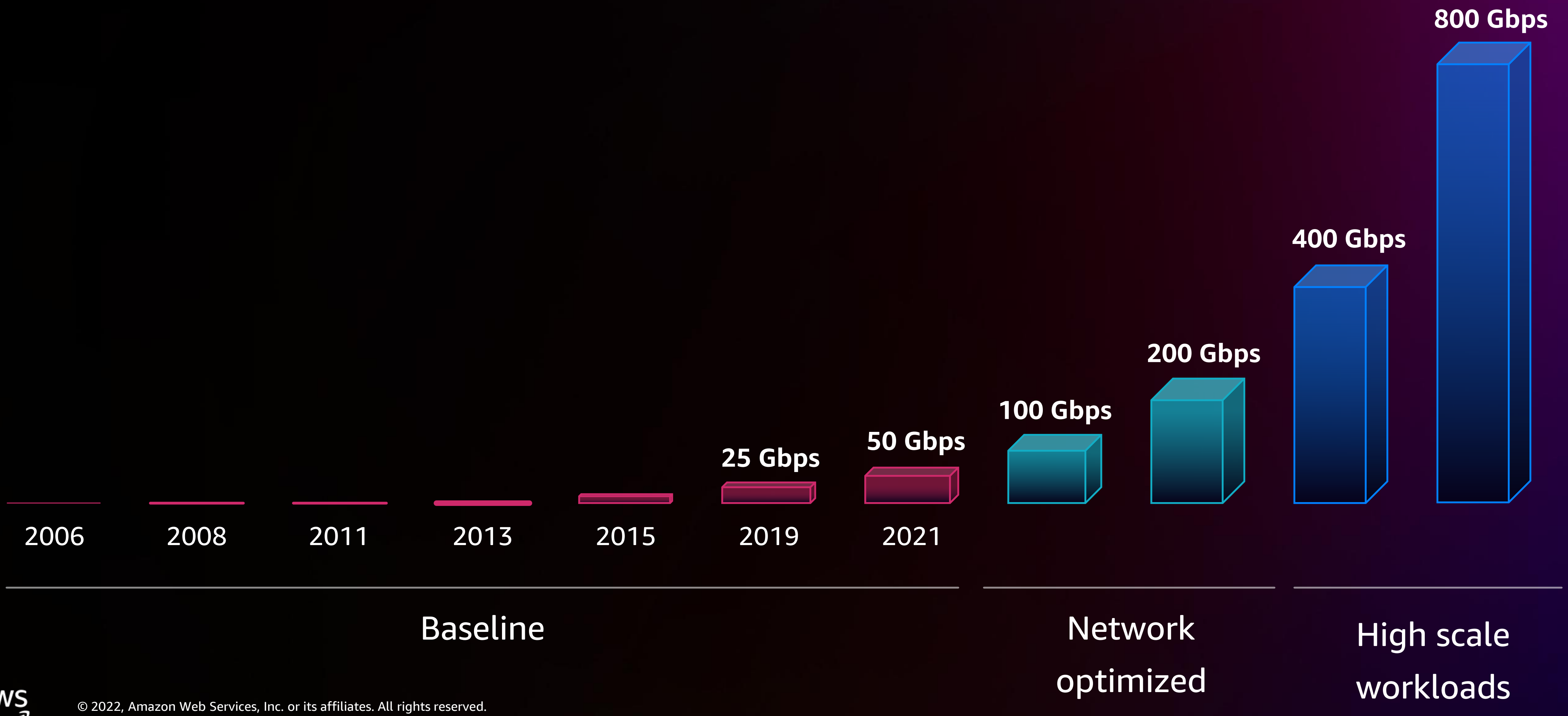
A Cloud-Optimized Transport Protocol for Elastic and Scalable HPC

Leah Shalev, Hani Ayoub, Nafea Bshara, and Erez Sabbag
Annapurna Labs, Amazon Web Services

***Abstract*—Amazon Web Services (AWS) took a fresh look at the network to provide consistently low latency required for supercomputing applications, while keeping the benefits of public cloud: scalability, elastic on-demand capacity, cost effectiveness, and fast adoption of newer CPUs and GPUs. We built a new network transport protocol, scalable reliable datagram (SRD), designed to utilize modern commodity multitenant datacenter networks (with a large number of network paths) while overcoming their limitations (load imbalance and inconsistent latency when unrelated flows collide). Instead of preserving packets order, SRD sends the packets over as many network paths as possible, while avoiding overloaded paths. To minimize jitter and to ensure the fastest response to network congestion fluctuations, SRD is implemented in the AWS custom Nitro networking card. SRD is used by HPC/ML frameworks on EC2 hosts via AWS elastic fabric adapter kernel-bypass interface.**

Amazon VPC

Instance network bandwidth



Nitro Card for EBS

Amazon Elastic Block Store

NVMe Controller

- Standard drivers broadly available

EBS Data Plane

- Encryption support
- NVM to remote storage protocol

Nitro Card for Instance Storage

Instance Storage

NVMe Controller

- Standard drivers broadly available

Instance Storage Data Plane

- Transparent Encryption
- Limiters
- Drive monitoring

Nitro Controller

Nitro
Controller

System Control

- Provides passive API endpoint
- Coordinates all other Nitro Cards
- Coordinates with Nitro Hypervisor
- Coordinates with Nitro Security Chip

Nitro Security Chip

Custom microcontroller that traps all I/O to non-volatile storage

Controllable from the Nitro Controller to hold system boot

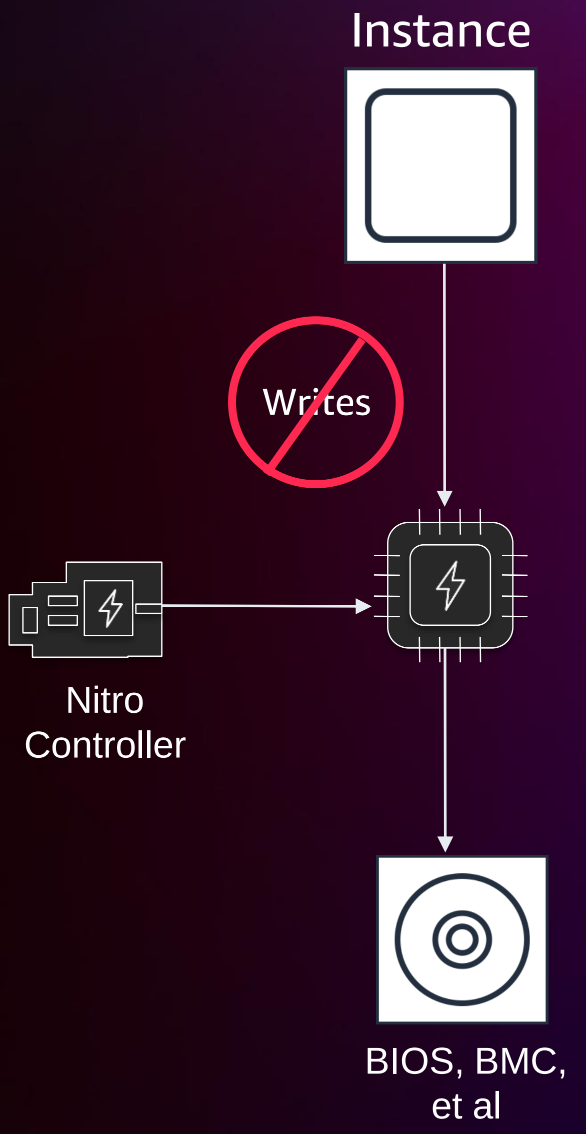
Provides a simple, hardware-based root of trust

Nitro Hardware Root of Trust

Radical simplification enabled by Nitro Cards

All write access to non-volatile storage is blocked in hardware

Simple to understand security due to lack of legacy



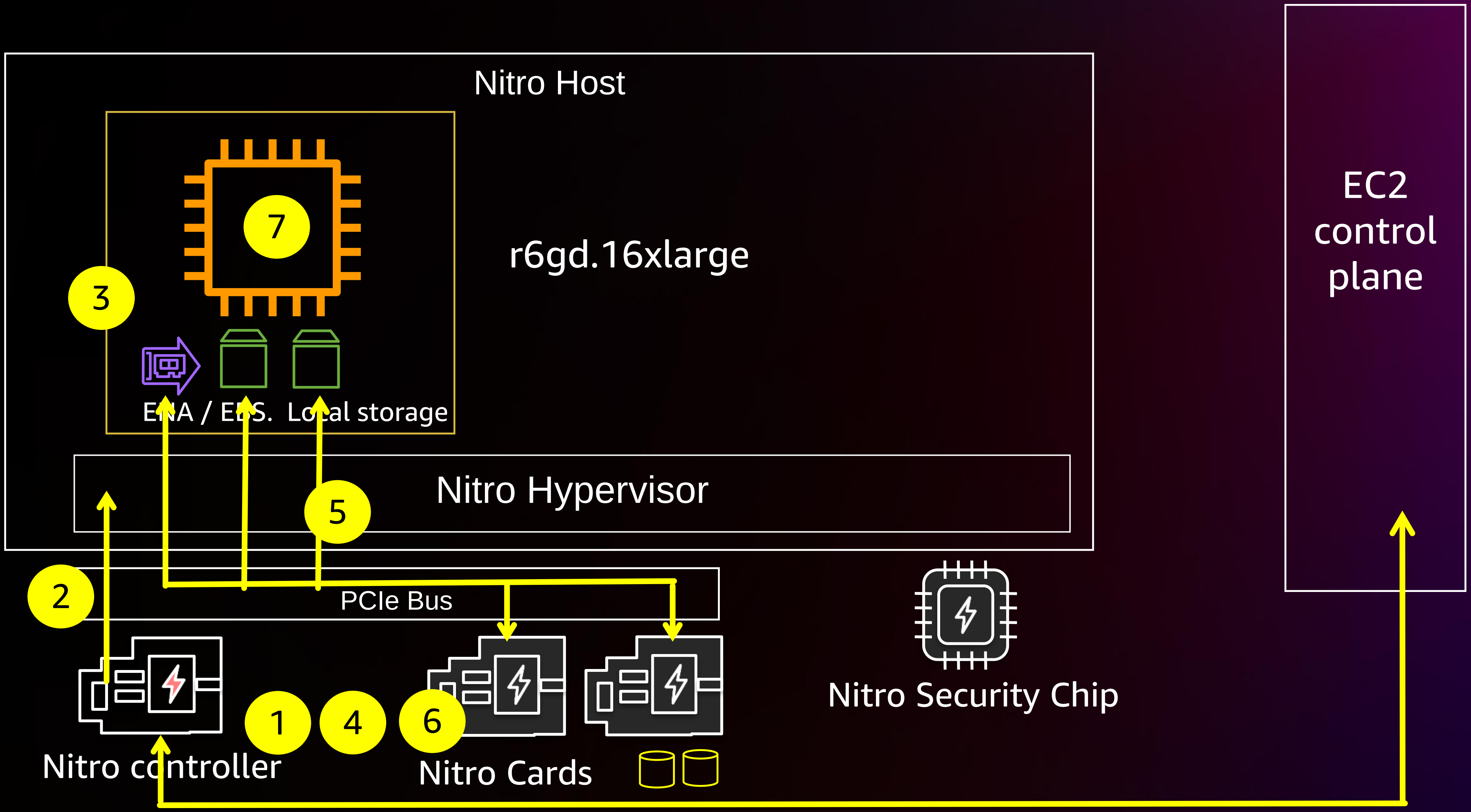
Nitro Hypervisor

KVM-based hypervisor with custom MM and small userspace

Only executes on behalf of instance, quiescent.

With Nitro, the hypervisor is minimal and performant

The Nitro system: all together



Broadest and deepest platform choice

Categories

General purpose + burstable

Compute-optimized

Memory-optimized

Storage-optimized

Accelerated computing

HPC-optimized

+

Capabilities

Choice of processor
(AWS, Intel, AMD, Apple)

Fast processors
(up to 4.5 GHz)

High memory footprint
(up to 24 TiB)

Instance storage
(HDD and NVMe)

Accelerated computing
(GPUs, FPGA, ASIC)

Networking
(up to 1600 Gbps)

Bare Metal

Size
(<1vCPU to 448 vCPU)

+

Options

Amazon EBS

Windows, Linux, UNIX, macOS

=

More than

575

Instance types

for virtually every workload and business need

New launches

- C6in, M6in, R6in – 6th generation network optimized instances, up to 200 Gbps network bandwidth, 80 Gbps of EBS bandwidth
- C7gn – Featuring Nitro v5 to enable best perf for Network-enabled performance, 200 Gbps, 2x per vCPU compared to current C6gn
- R7iz – Intel Sapphire Rapids, all core turbo freq up to 3.9 GHz and up to 15% better compute perf vs. previous generation instances
- Hpc6id – memory and data-intensive HPC workloads, 200 Gbps with EFA, 2x higher than current generation HPC instances
- Inf2 – 4x higher throughput, up to 10x lower latency vs. Inf1, up to 12 Inferentia2 accelerators and up to 384 GB of HBM2e high speed accelerator memory
- Mac2 – Apple silicon M1 based instances

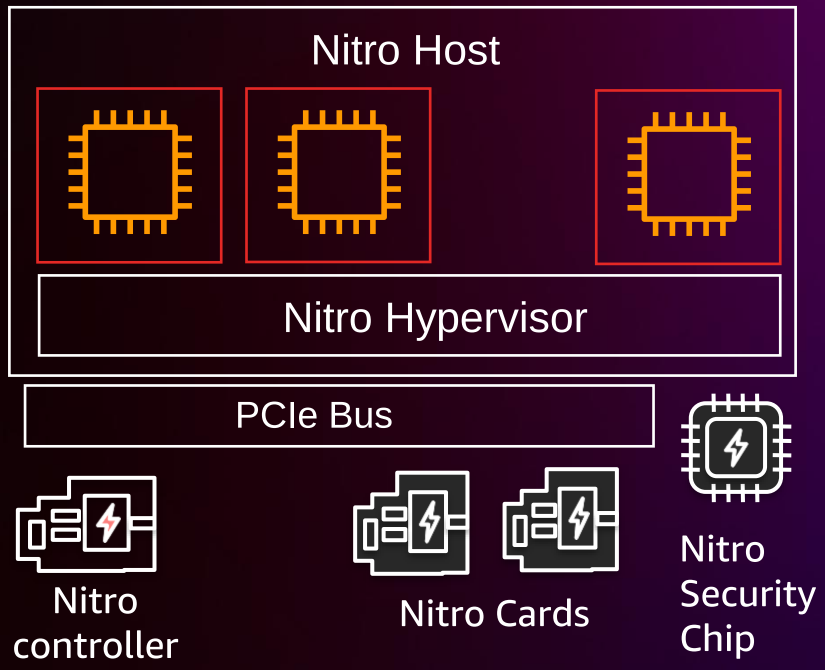
Security

Shared responsibility for security

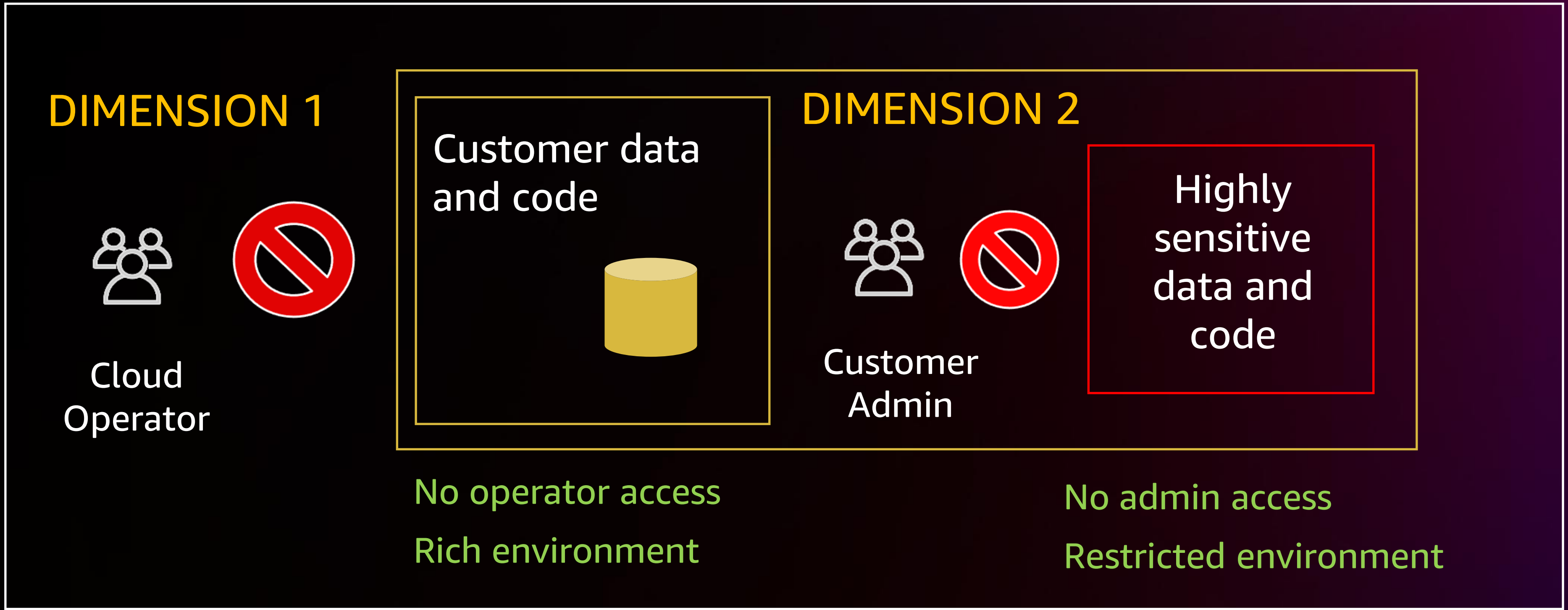
Security is our top priority

It’s a shared responsibility

- Security of the cloud is AWS’s responsibility
- Security in the cloud is customer’s responsibility
    - VPC security groups and encryption, management of credentials, management of guest OS and applications



Confidential computing



Protecting customer code and sensitive data in use

Confidentiality protections of the Nitro system

DIMENSION 1

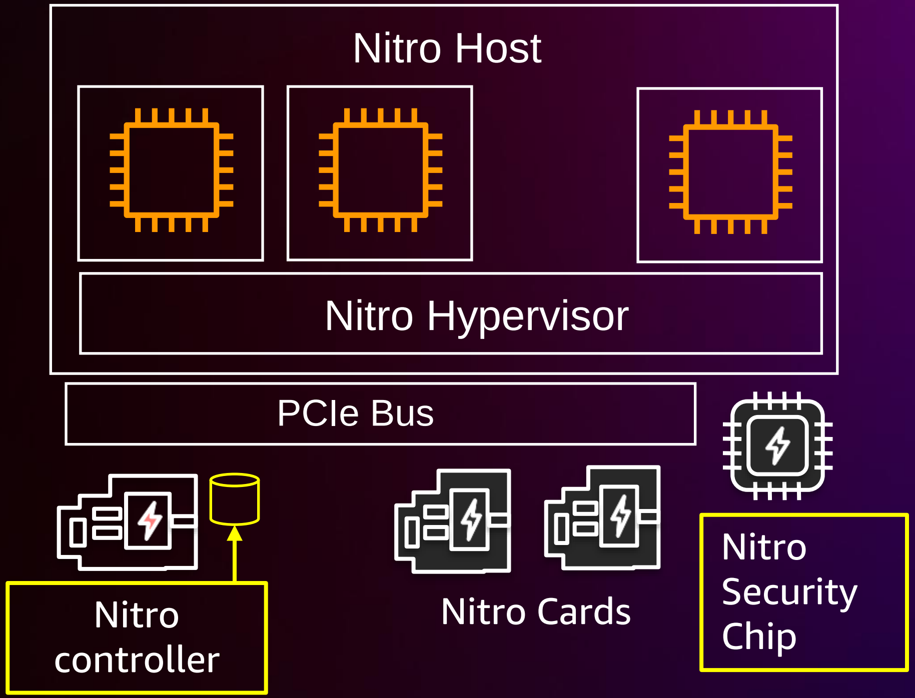
Designed to provide strong isolation between host and Nitro Cards

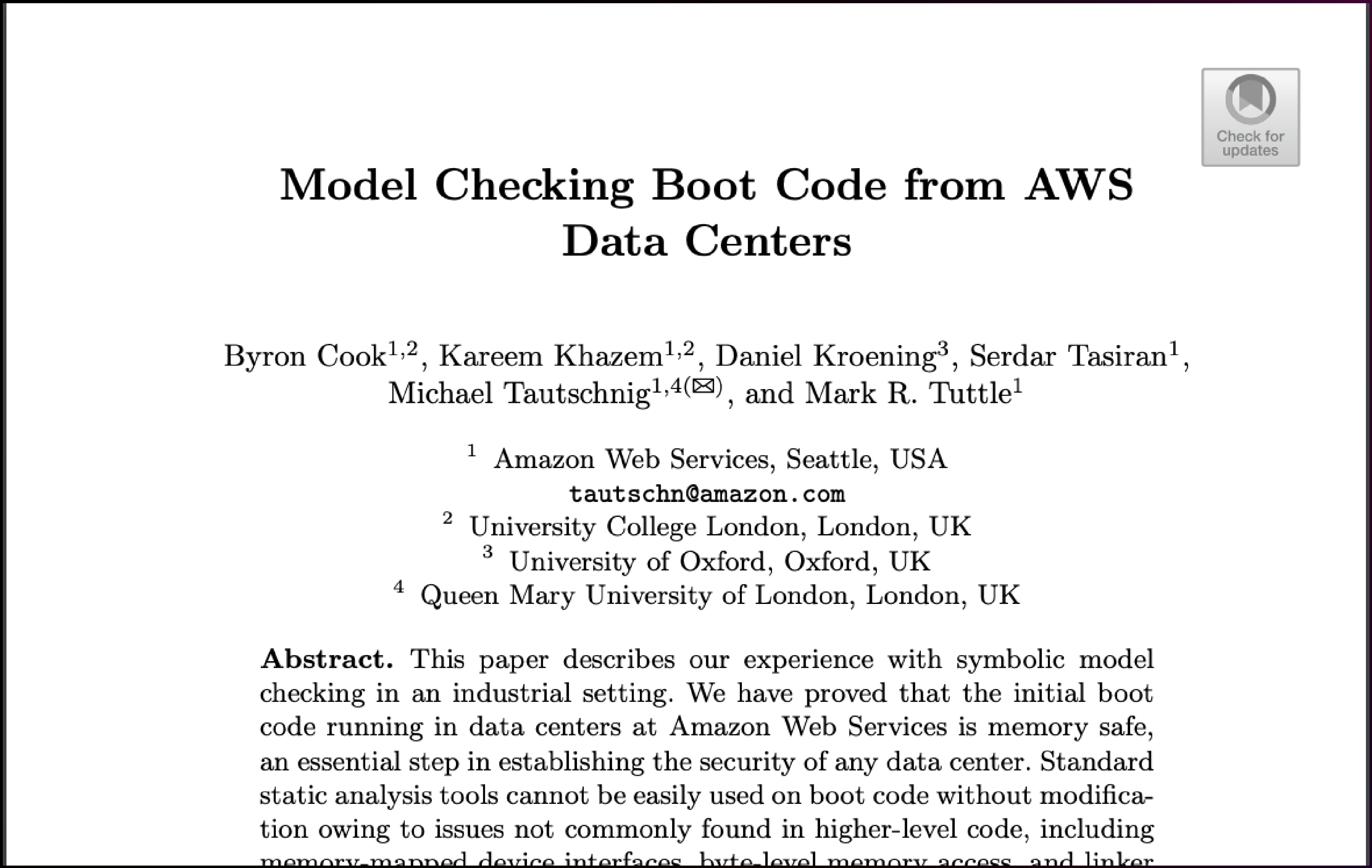
Measured boot process starting from a root of trust

- Measurements extended into PCRs in TPM attached to Nitro controller
- SSD decrypted via key sealed against measurements

Nitro security chip

- Intercepts and moderates all operations to local non-volatile storage
- On reboot, holds platform in reset and verifies integrity of system firmware





Model Checking Boot Code from AWS Data Centers

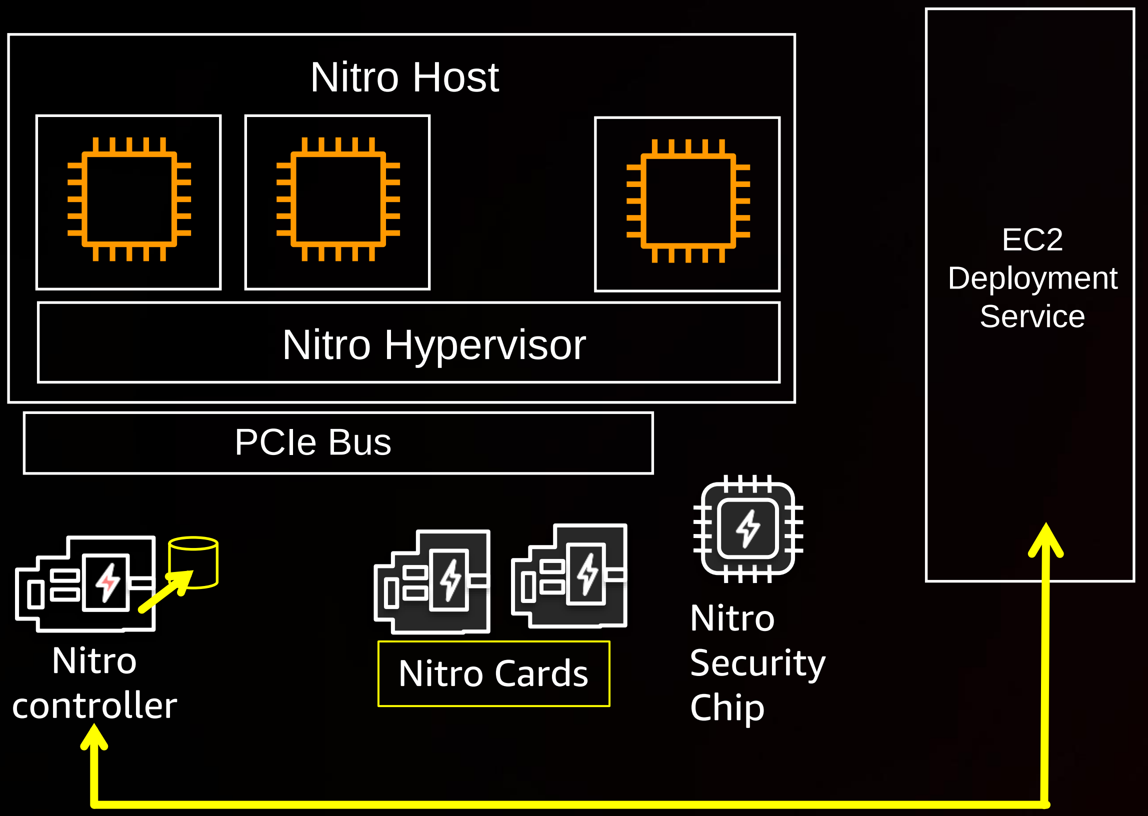
Byron Cook[1,2], Kareem Khazem[1,2], Daniel Kroening[3], Serdar Tasiran[1], Michael Tautschnig[1,4(✉)], and Mark R. Tuttle[1]

[1] Amazon Web Services, Seattle, USA
tautschn@amazon.com
[2] University College London, London, UK
[3] University of Oxford, Oxford, UK
[4] Queen Mary University of London, London, UK

Abstract. This paper describes our experience with symbolic model checking in an industrial setting. We have proved that the initial boot code running in data centers at Amazon Web Services is memory safe, an essential step in establishing the security of any data center. Standard static analysis tools cannot be easily used on boot code without modification owing to issues not commonly found in higher-level code, including memory-mapped device interfaces, byte-level memory access, and linker

Confidentiality protections of the Nitro system

DIMENSION 1



Only production signed software and firmware is deployed to the EC2 fleet

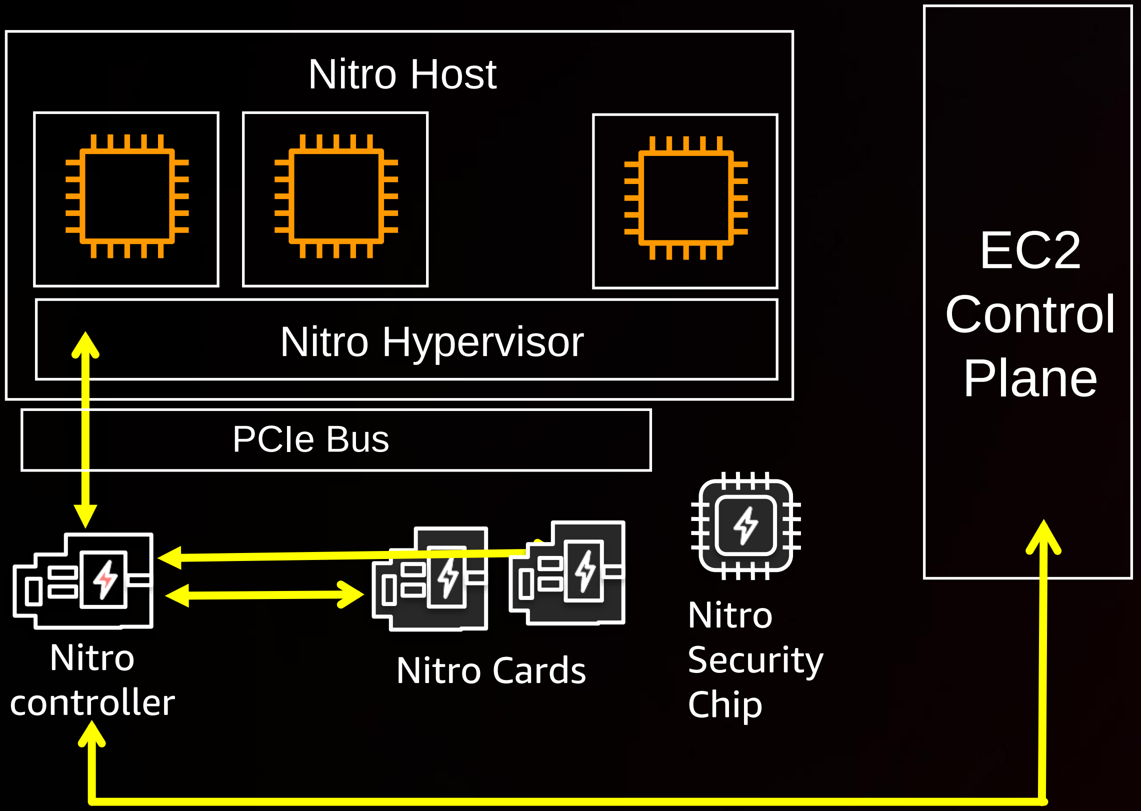
Software and firmware components can be live-updated

Specialized Nitro cards for IO implement data encryption for network and storage with secure storage integrated in the SoC

For additional defense-in-depth against physical attacks at the memory interface level, we offer memory encryption on various EC2 instances

No AWS operator access

DIMENSION 1



API: Encrypted, authenticated, authorized, and logged

No operator access to the Nitro System

All Nitro operations are done via secure, authenticated APIs

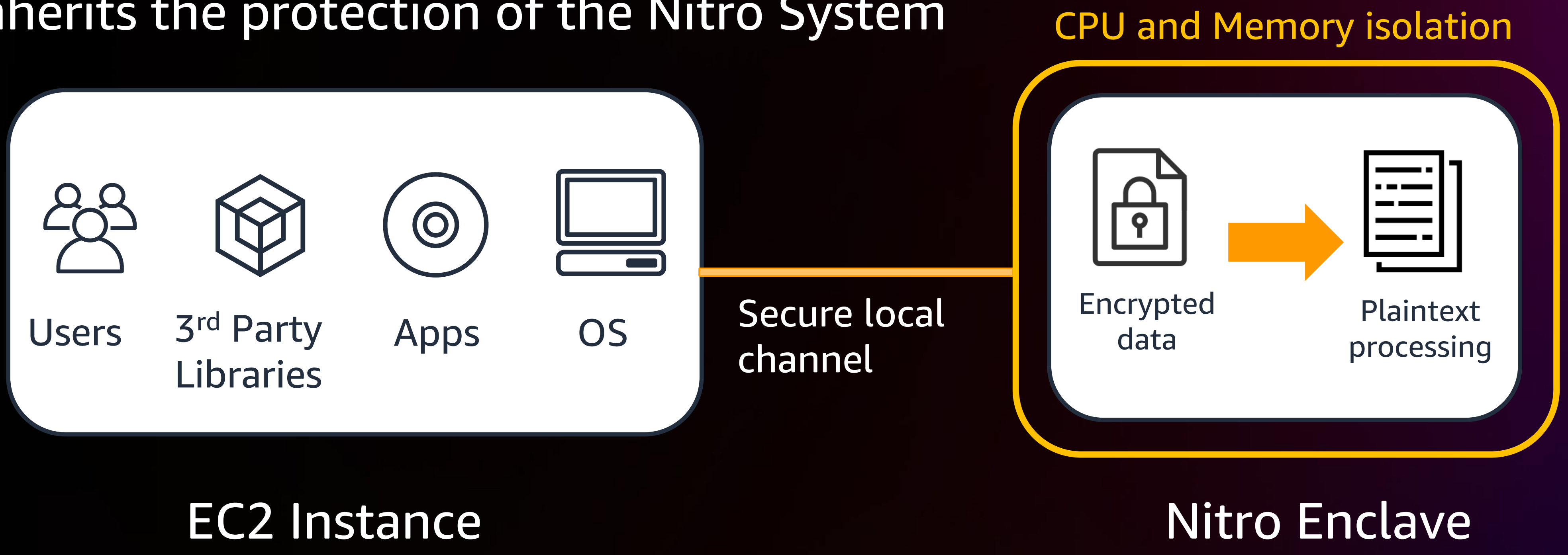
Uses the principle of least privilege

None of these APIs have the ability to access customer data on the EC2 server

AWS Nitro Enclaves

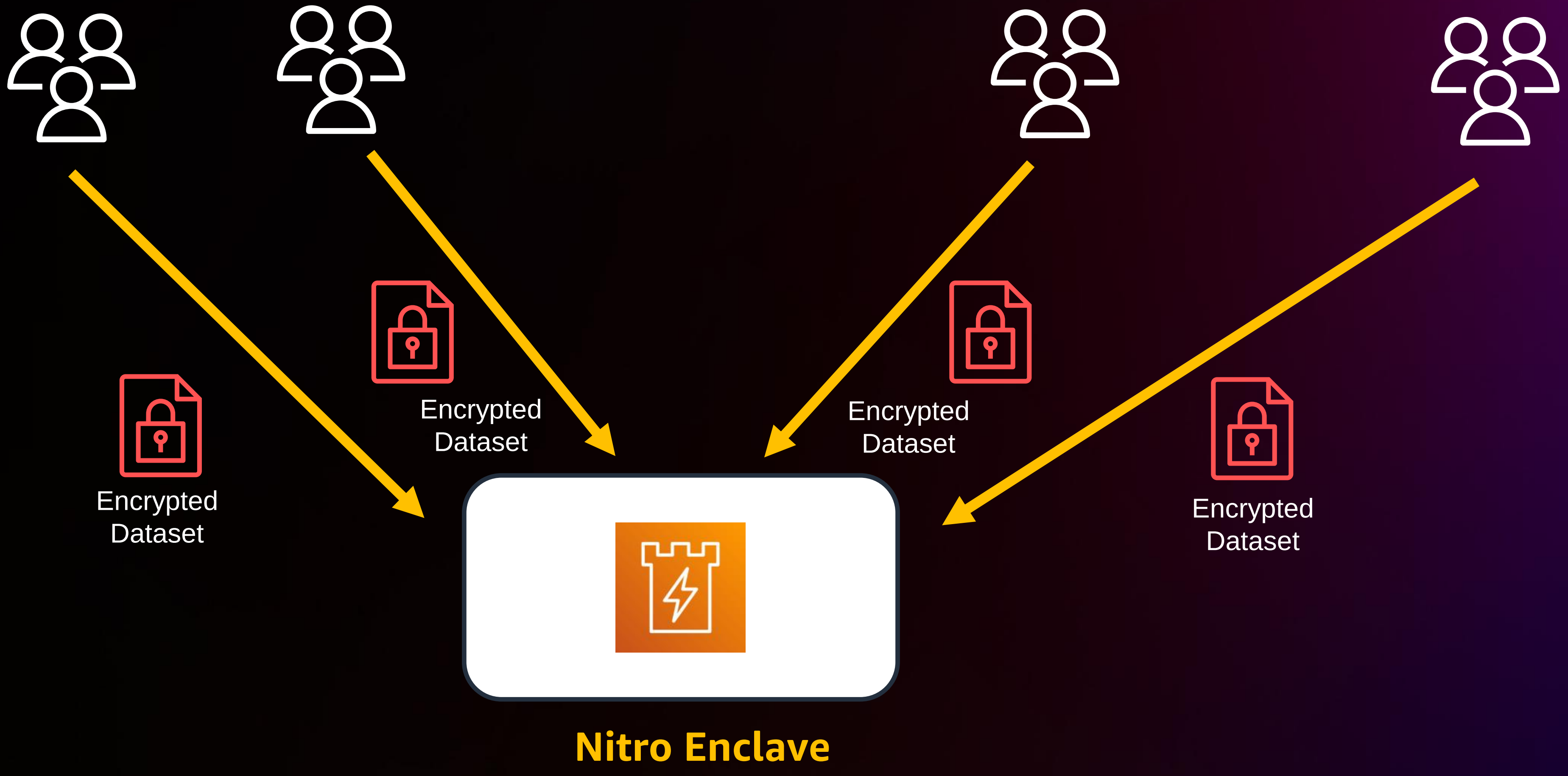
DIMENSION 2

- For customers who want to take confidentiality a step further and isolate their highly sensitive data from the users, applications, and libraries on their EC2 instance
- Inherits the protection of the Nitro System



Multi-party collaboration

TWO OR MORE PARTIES PROCESS SENSITIVE DATA WITHOUT GIVING ACCESS TO EACH OTHER



UEFI Secure Boot

UEFI Secure Boot flow ensures that the bootloader is properly signed by a known authority

Validate the signed bootloader (for example, Grub2) against certificates stored in UEFI

Fall back to backup bootloader or stop if validation fails

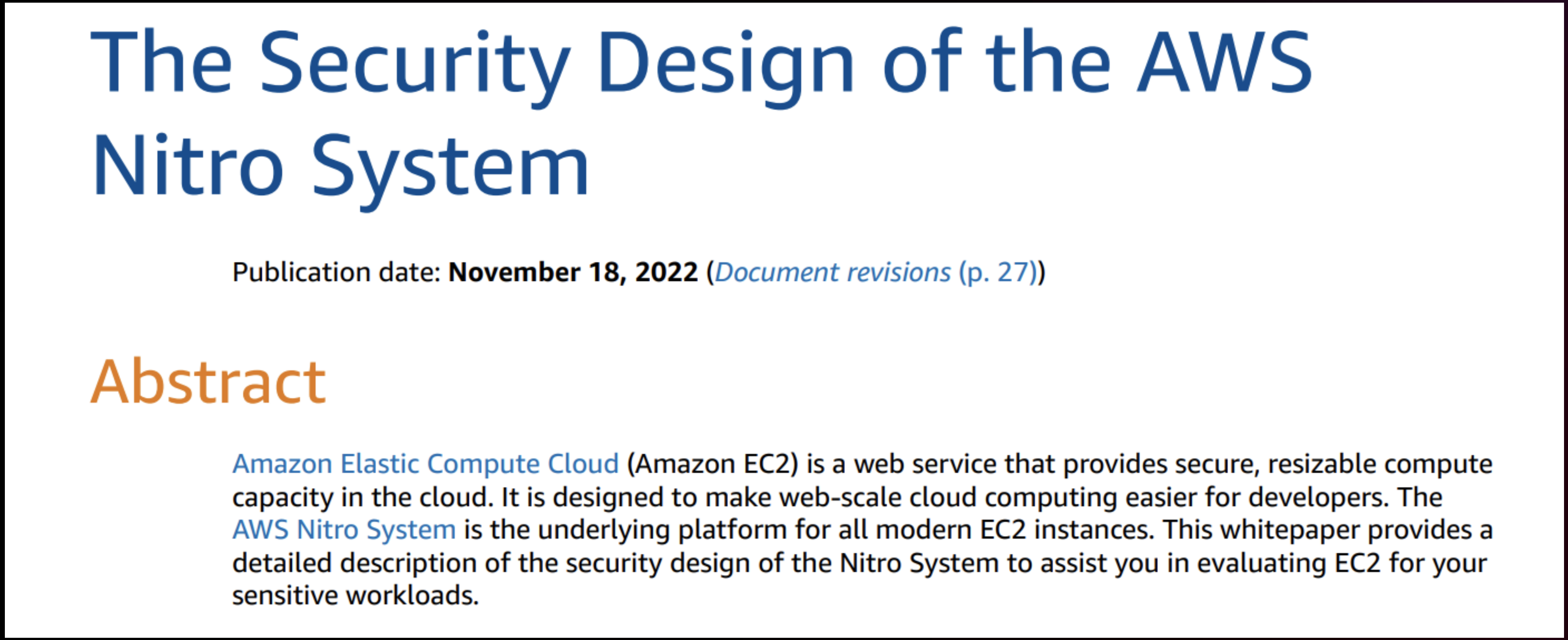
NitroTPM

A trusted platform module

Conforms to the industry standard TPM 2.0 specification

Software compatibility. Makes it easier for customers to migrates applications to rely on TPM to EC2.

Provides capabilities like attestation system state, store and generate cryptographic data, and prove platform identity to your EC2 instance



The Security Design of the AWS Nitro System

Publication date: **November 18, 2022** (*Document revisions* (p. 27))

Abstract

Amazon Elastic Compute Cloud (Amazon EC2) is a web service that provides secure, resizable compute capacity in the cloud. It is designed to make web-scale cloud computing easier for developers. The AWS Nitro System is the underlying platform for all modern EC2 instances. This whitepaper provides a detailed description of the security design of the Nitro System to assist you in evaluating EC2 for your sensitive workloads.

https://a.co/hYWhsH9

- Detailed review of the security design the three primary components of the AWS Nitro System
- Deep dive on the AWS Nitro System integrity protections, tenant isolation model, and no operator access design

EC2 bare metal instances

Motivations

Journey started with VMware in 2016

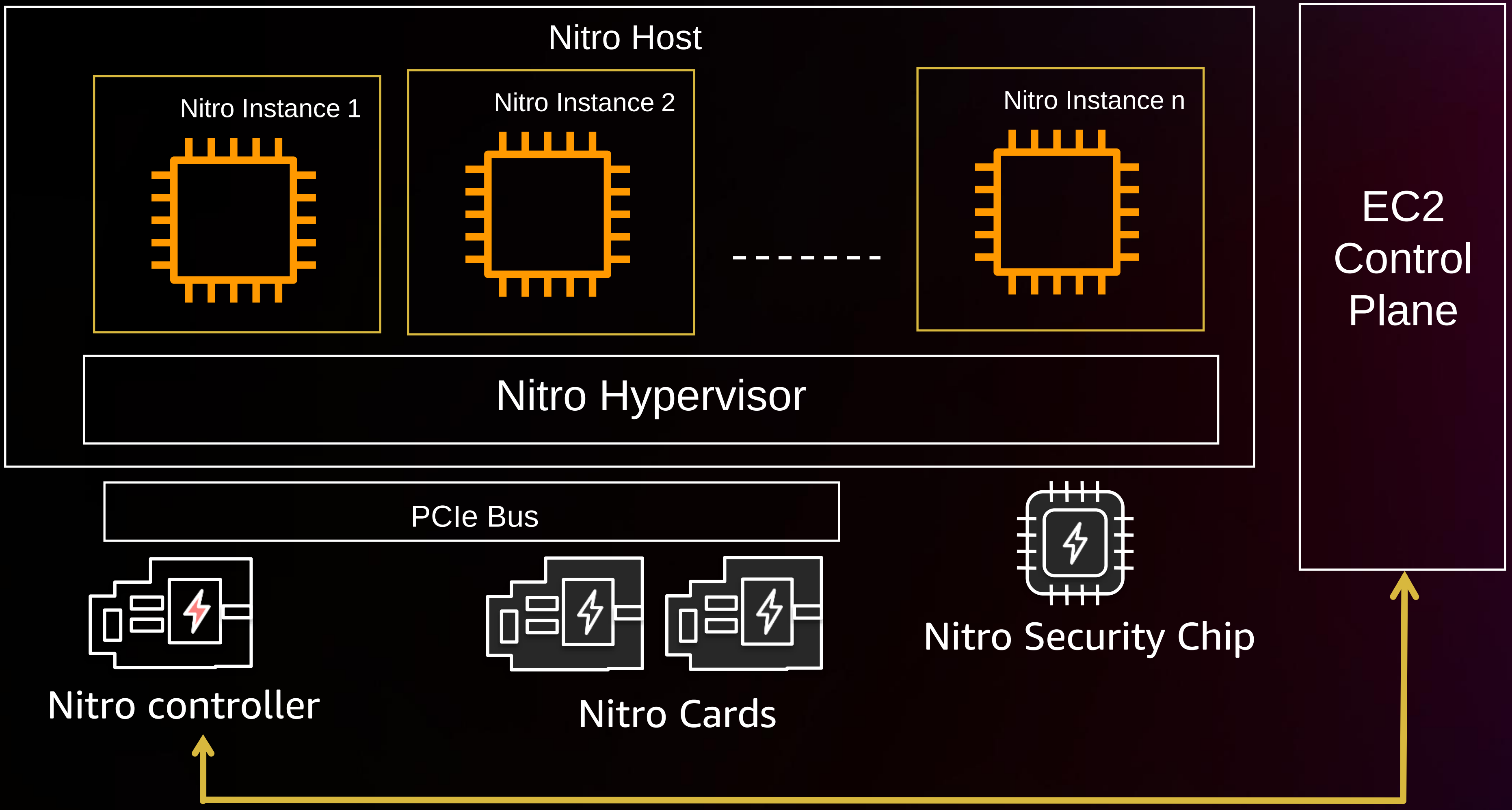
- Vision to build a managed SDDC architecture running on AWS infrastructure
- Customers enjoy AWS benefits while not changing their operational mindset or tooling

Benefits all customers

Use cases

- Custom hypervisors (e.g. ESXi), secure containers (e.g. Clear Linux containers)
- Legacy workloads not supported in virtual environments
- Run applications that benefit from deep performance analysis tools – e.g. Intel VTune profiler
- Licensing-restricted business critical applications

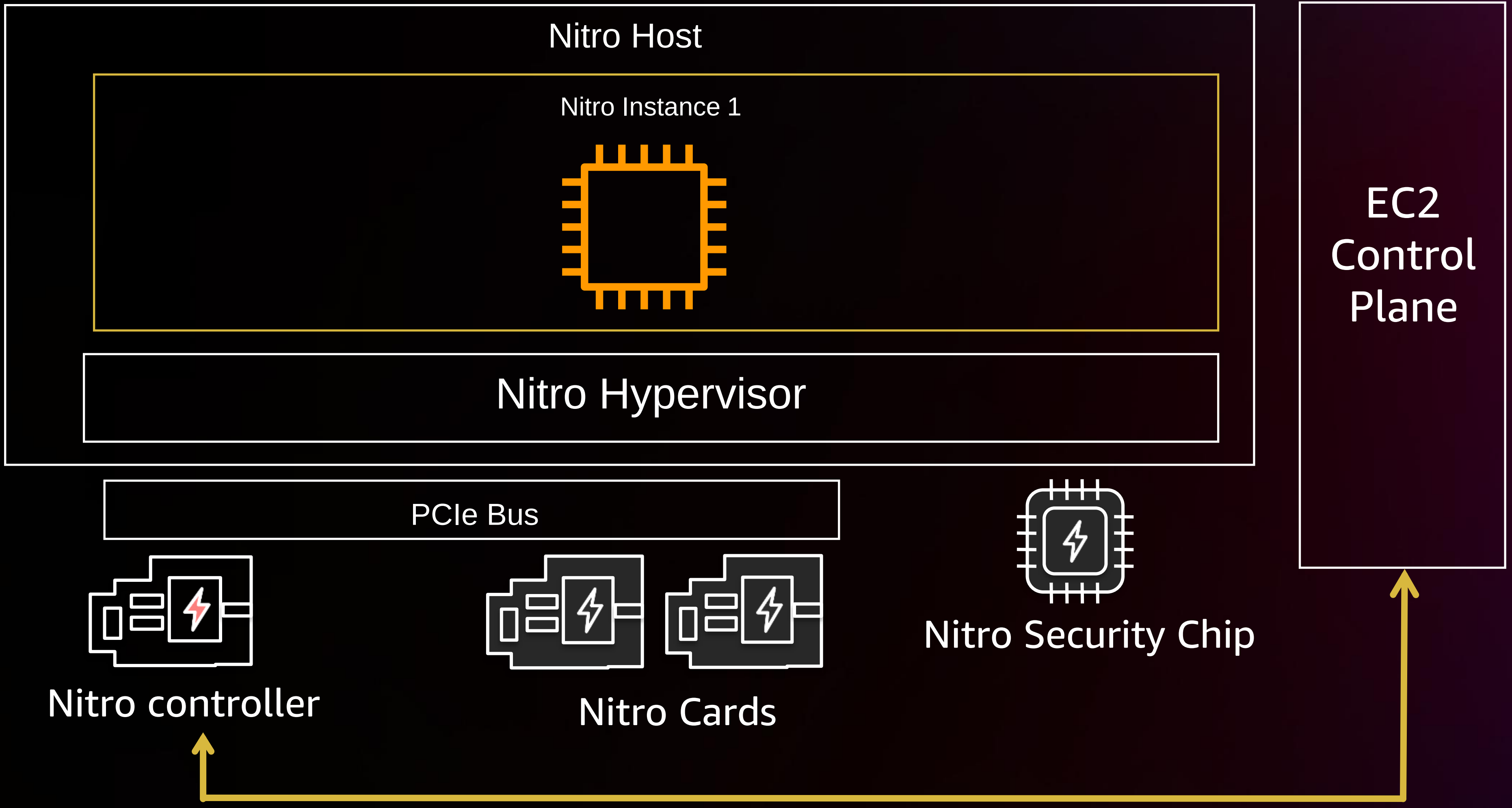
Virtualized instances: c6i.4xlarge



- 16 vCPU
- 32 GB memory
- Up to 12.5 Gbps of network bandwidth
- Up to 10 Gbps of EBS bandwidth

API: Authenticated, authorized, and encrypted

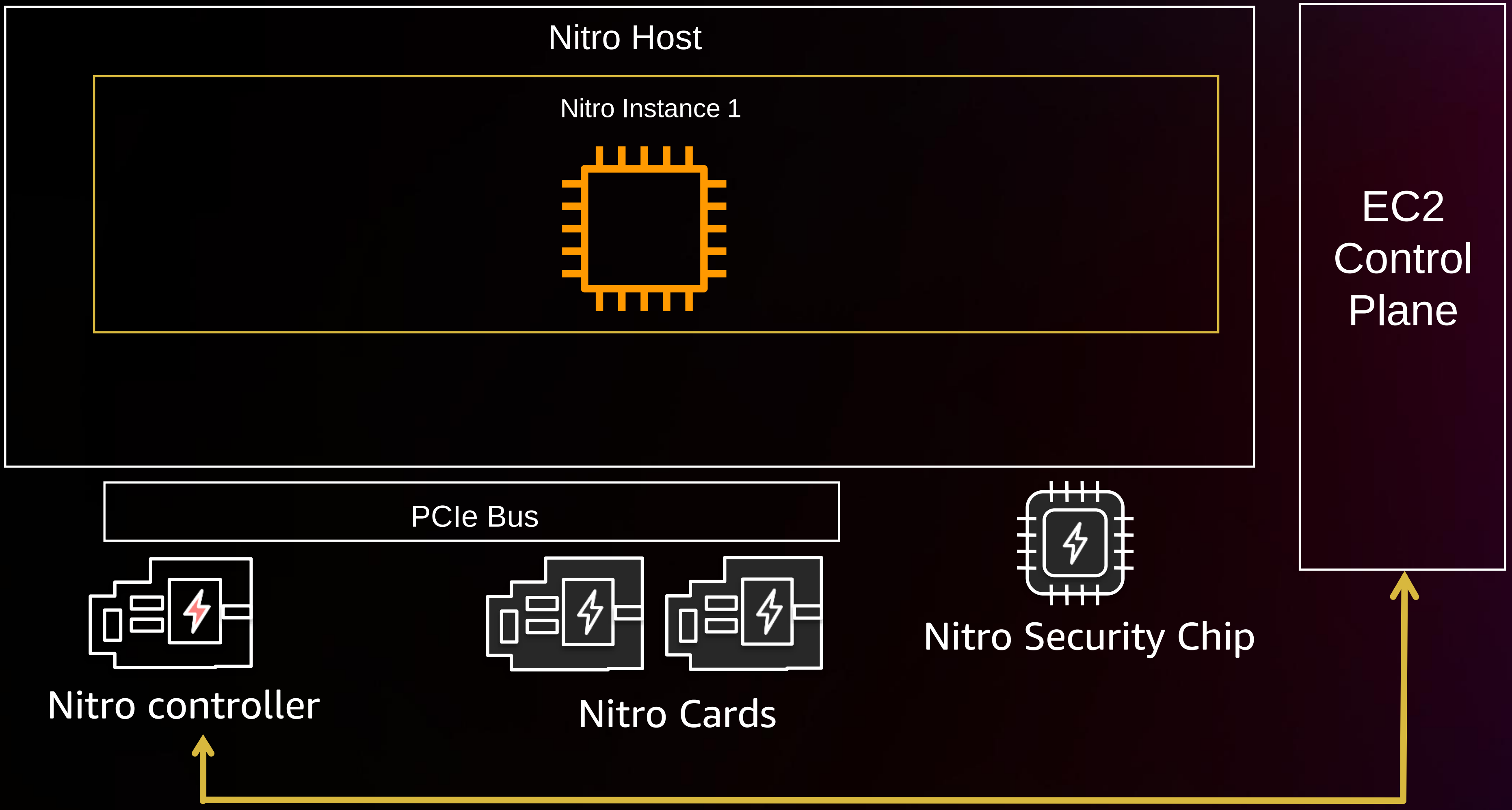
Virtualized instances: c6i.32xlarge



API: Authenticated, authorized and encrypted

- 128 vCPU
- 256 GB memory
- Up to 50 Gbps of network bandwidth
- Up to 40 Gbps of EBS bandwidth

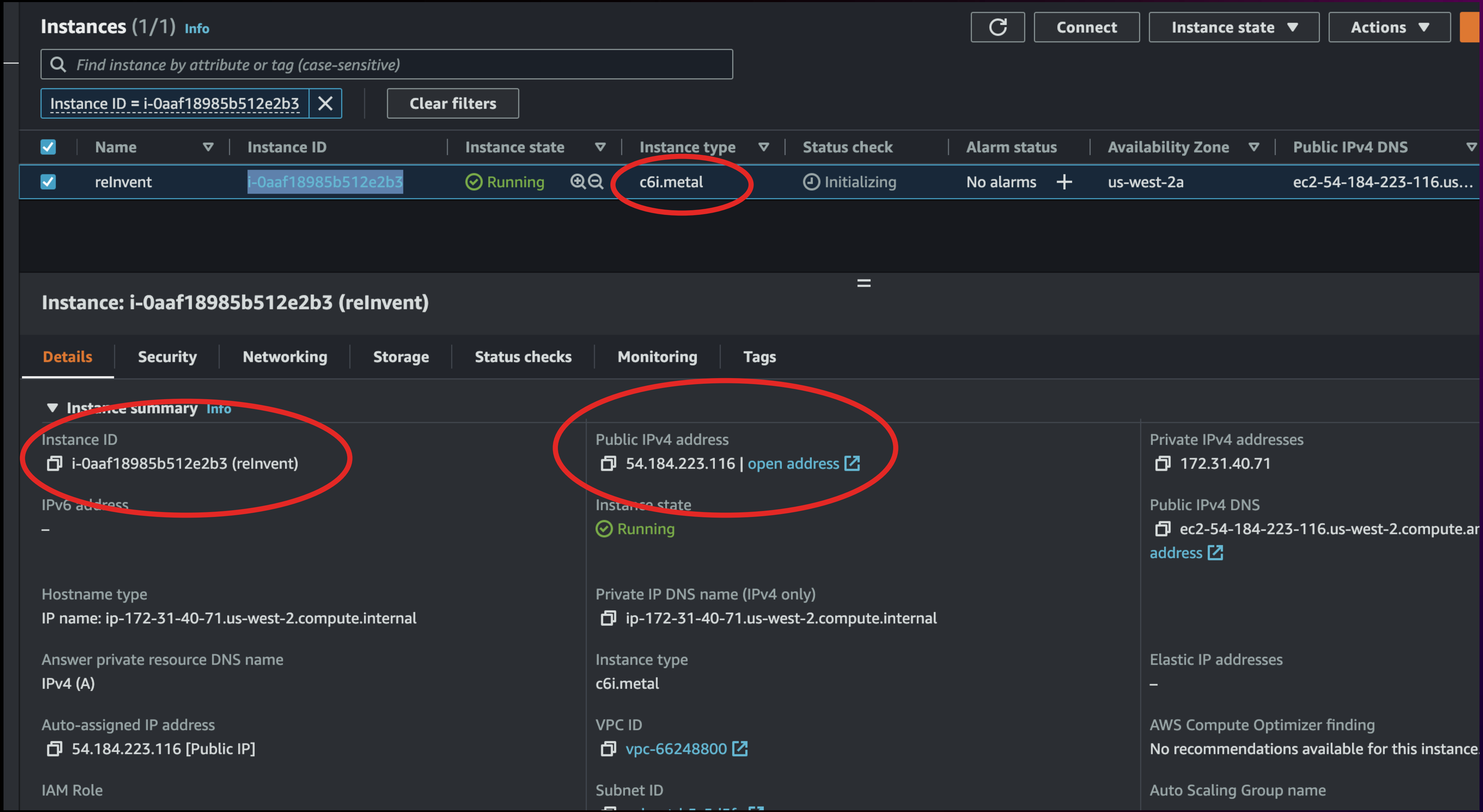
Bare metal instances: c6i.metal



API: Authenticated, authorized and encrypted

- 128 vCPU
- 256 GB memory
- Up to 50 Gbps of network bandwidth
- Up to 40 Gbps of EBS bandwidth

Example: c6i.metal



Instances (1/1) Info

Instance ID = i-0aaf18985b512e2b3

☑	Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IPv4 DNS
☑	reInvent	i-0aaf18985b512e2b3	✓ Running	c6i.metal	Initializing	No alarms +	us-west-2a	ec2-54-184-223-116.us...

Instance: i-0aaf18985b512e2b3 (reInvent)

▼ Instance summary Info

Instance ID
i-0aaf18985b512e2b3 (reInvent)

IPv6 address
–

Hostname type
IP name: ip-172-31-40-71.us-west-2.compute.internal

Answer private resource DNS name
IPv4 (A)

Auto-assigned IP address
54.184.223.116 [Public IP]

IAM Role

Public IPv4 address
54.184.223.116 | open address

Instance state
✓ Running

Private IP DNS name (IPv4 only)
ip-172-31-40-71.us-west-2.compute.internal

Instance type
c6i.metal

VPC ID
vpc-66248800

Subnet ID

Private IPv4 addresses
172.31.40.71

Public IPv4 DNS
ec2-54-184-223-116.us-west-2.compute.ar
address

Elastic IP addresses
–

AWS Compute Optimizer finding
No recommendations available for this instance.

Auto Scaling Group name

Looking around …

```
rmurty@u546449c1a91c51:~$ date
Fri Nov 18 17:43:20 PST 2022
rmurty@u546449c1a91c51:~$ ssh -i ~/.ssh/rmurty-pdx-key.pem ubuntu@54.184.223.116

ubuntu@ip-172-31-40-71:~$ curl -H "X-aws-ec2-metadata-token: $TOKEN" http://169.254.169.254/latest/meta-
data/instance-type
c6i.metal

ubuntu@ip-172-31-40-71:~$ curl -H "X-aws-ec2-metadata-token: $TOKEN" http://169.254.169.254/latest/meta-
data/instance-id
i-0aaf18985b512e2b3

ubuntu@ip-172-31-40-71:~$ cat /proc/cpuinfo | grep processor | wc -l
128

ubuntu@ip-172-31-40-71:~$ free
           total       used       free     shared  buff/cache   available
Mem:   263930804    1301440  262239480       2900      389884   261284852
Swap:          0          0          0
```

PCIe: A lot more devices

```
ubuntu@ip-172-31-40-71:~$ lspci -tv
-+-[0000:ff]-+-00.0  Intel Corporation Device 344c
 |           +-00.1  Intel Corporation Device 344c
 |           +-00.2  Intel Corporation Device 344c
 |           +-00.3  Intel Corporation Device 344c
...
 +-[0000:be]-+-00.0  Intel Corporation Device 09a2
 |           +-00.1  Intel Corporation Device 09a4
 |           +-00.2  Intel Corporation Device 09a3
 |           +-00.4  Intel Corporation Device 0998
 |           \-02.0-[bf-e0]----00.0-[c0-e0]--+-00.0-[c1]----00.0  Amazon.com, Inc. Device 8250
 |                                           +-01.0-[c2]----00.0  Amazon.com, Inc. Device 0061
 |                                           +-02.0-[c3]—
...
 +-[0000:2c]-+-00.0  Intel Corporation Device 09a2
 |           +-00.1  Intel Corporation Device 09a4
 |           +-00.2  Intel Corporation Device 09a3
 |           +-00.4  Intel Corporation Device 0998
 |           \-02.0-[2d-3e]----00.0-[2e-3e]--+-00.0-[2f]--
 |                                           +-01.0-[30]----00.0  Amazon.com, Inc. Elastic Network Adapter (ENA)
 |                                           +-02.0-[31]--
```

NVMe exposed EBS root volume

```
ubuntu@ip-172-31-40-71:~$ sudo apt-get install nvme-cli
Reading package lists... Done
…
Unpacking nvme-cli (1.16-3build1) ...
Setting up nvme-cli (1.16-3build1) ...
…

ubuntu@ip-172-31-40-71:~$ sudo nvme id-ctrl /dev/nvme0n1 -v
NVME Identify Controller:
vid     : 0x1d0f
ssvid   : 0x1d0f
sn      : vol03a4cfccc443ee287
mn      : Amazon Elastic Block Store
fr      : 2.0
…
vs[]:
       0  1  2  3  4  5  6  7  8  9  a  b  c  d  e  f
0000: 73 64 61 31 20 20 20 20 20 20 20 20 20 20 20 20 "sda1............"
```

```
ubuntu@ip-172-31-40-71:~$ lsblk
NAME         MAJ:MIN RM  SIZE RO TYPE MOUNTPOINTS
…
nvme0n1      259:0    0    8G  0 disk
├─nvme0n1p1  259:1    0  7.9G  0 part /
├─nvme0n1p14 259:2    0    4M  0 part
└─nvme0n1p15 259:3    0  106M  0 part /boot/efi
```

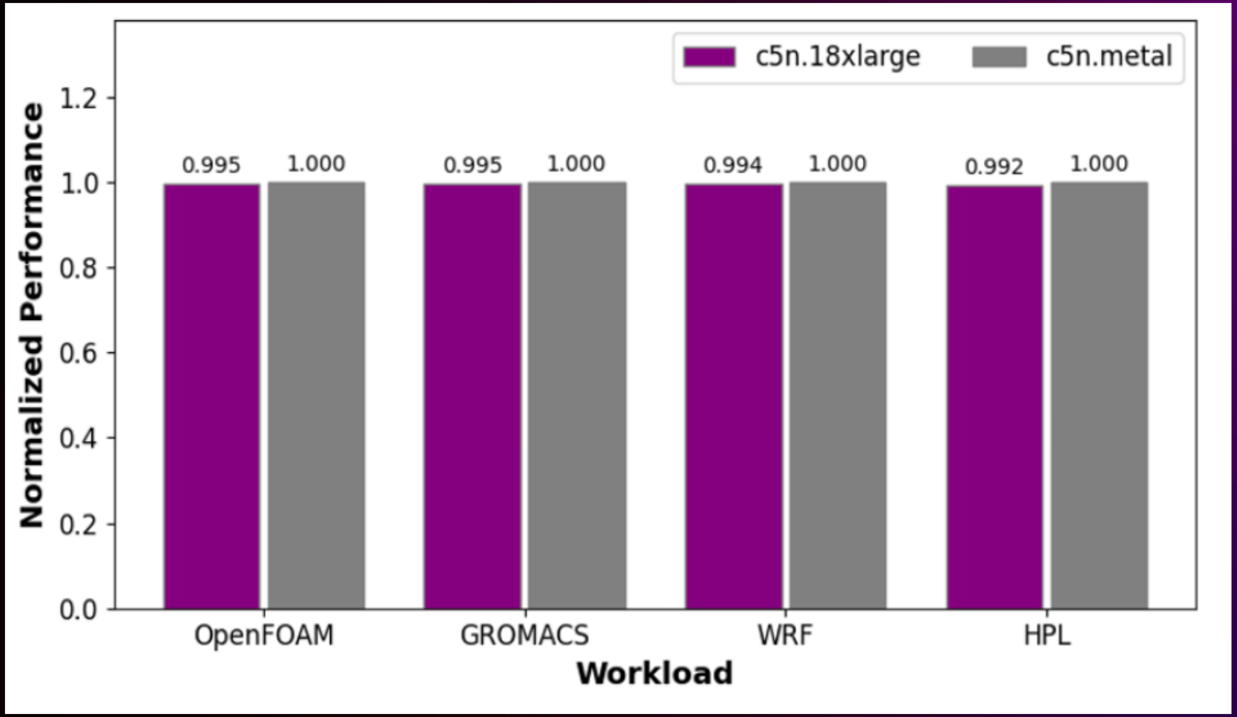
EC2 bare metal or virtualized instances?

Both are enabled by AWS Nitro

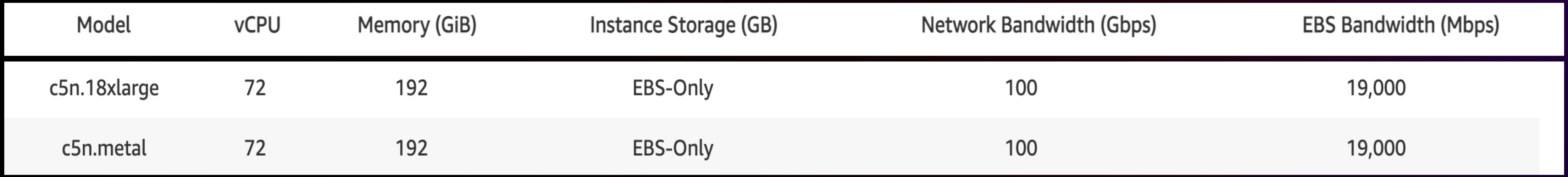
Same API experience, instance lifecycle, scalability, elasticity, reliability, security, SLA, pricing and purchase options

No perceptible performance difference for vast majority of use-cases

Let your use-case and feature requirements guide you



https://aws.amazon.com/blogs/hpc/bare-metal-performance-with-the-aws-nitro-system/



Model	vCPU	Memory (GiB)	Instance Storage (GB)	Network Bandwidth (Gbps)	EBS Bandwidth (Mbps)
c5n.18xlarge	72	192	EBS-Only	100	19,000
c5n.metal	72	192	EBS-Only	100	19,000

Launch time improvements

Customer pain point: Large time to provision bare metal instances

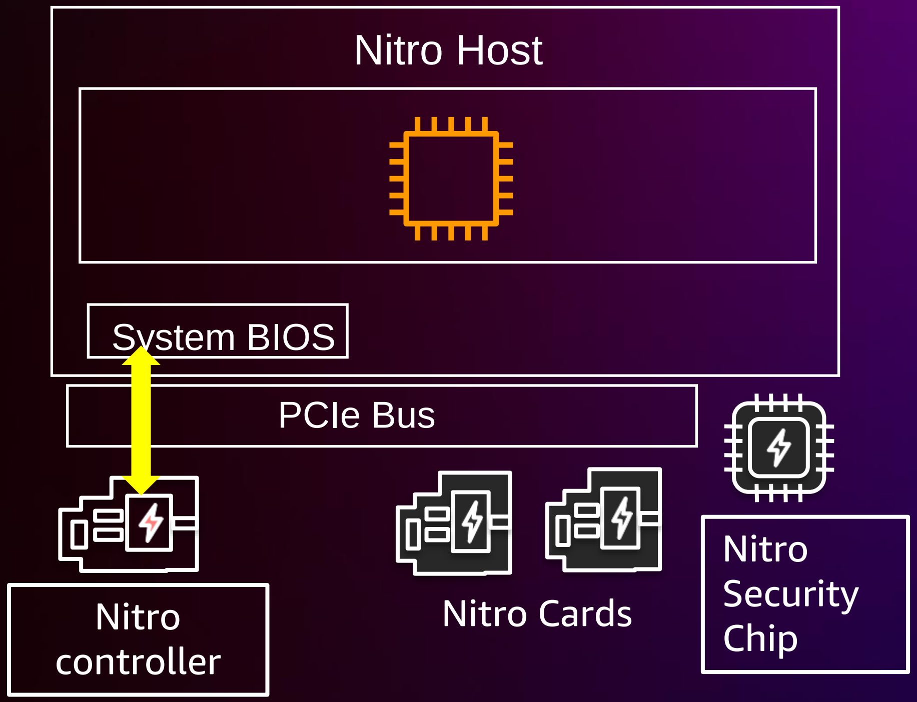
- Minutes vs. seconds
- Maintain warm pool of instances

Tight integration between System BIOS and Nitro controller

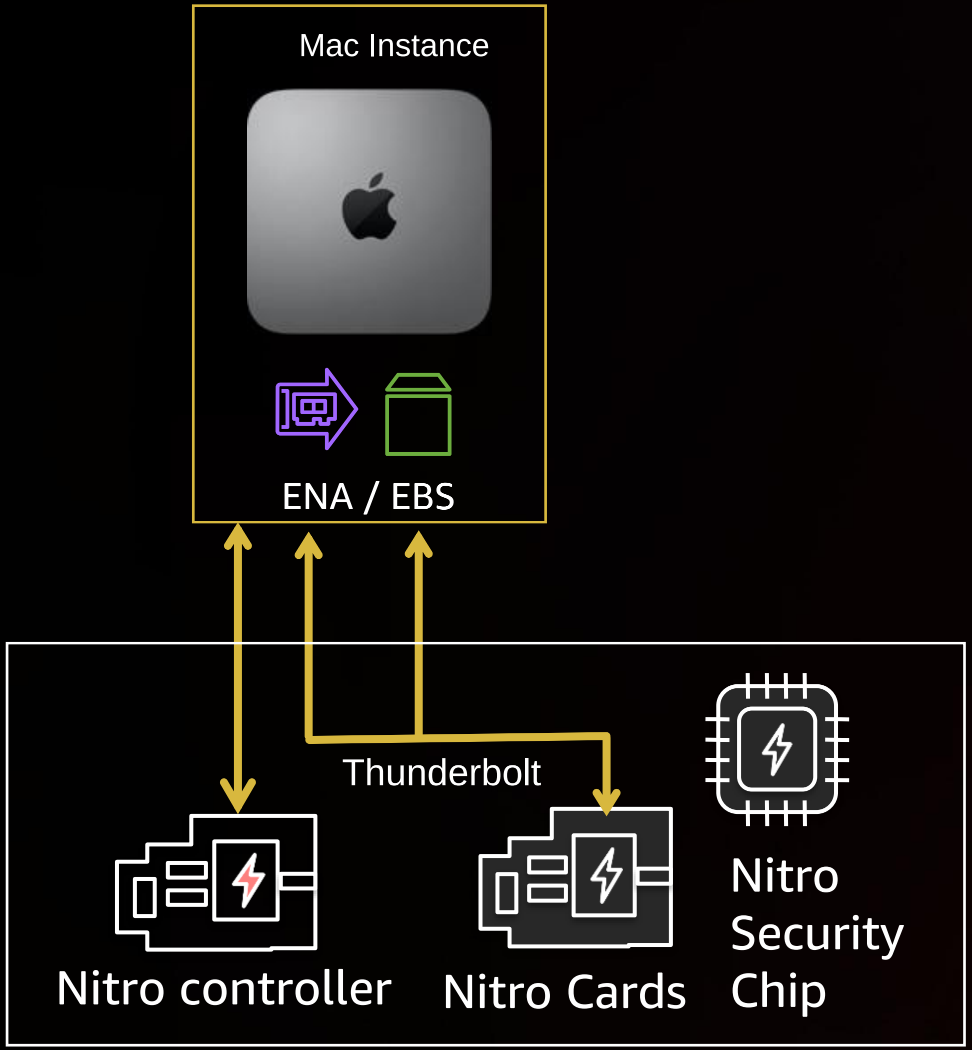
- Allows Nitro controller to “hold” BIOS from probing the PCIe devices, boot from NVMe

90% reduction in instance launch time

- My c6i.metal instance took about 40s to be accessible!



Apple Mac + Nitro = EC2 Mac instances



- Develop, build, test and sign for iOS, tvOS or watchOS applications
  - Need to run on Apple machines running macOS
- Leverage EC2 bare metal capability
- EC2 customers benefit
  - Elasticity, security, reliability
  - macOS AMIs, EBS volume for boot
  - VPC SG, CloudWatch metrics

Bringing macOS to AWS meant bringing Apple Macs into our data centers

EC2 Virtualized vs. EC2 Bare Metal vs. Legacy

<table>
<tr><th></th><th>EC2 Virtualized</th><th>EC2 Bare Metal</th><th>Legacy Bare Metal</th></tr>
<tr><td><b>Elasticity</b></td><td colspan="2">Spin up or down as needed</td><td>Overprovision ahead of time</td></tr>
<tr><td><b>Scalability</b></td><td colspan="2">Scale horizontally or vertically* within seconds</td><td>Wait for weeks or months to order, procure and install new capacity</td></tr>
<tr><td><b>Flexibility</b></td><td colspan="2">Seamlessly move across instance families and generations</td><td>Procure new and decommission old servers</td></tr>
<tr><td><b>Security</b></td><td colspan="2">AWS security, patching, updates</td><td>Customer responsible for security, privacy, isolation boundaries</td></tr>
<tr><td><b>Maintenance</b></td><td colspan="2">AWS maintains hardware, firmware and hypervisor*</td><td>Customer maintains hardware and firmware</td></tr>
<tr><td><b>Managed Services</b></td><td colspan="2">Full suite of AWS services, AMI compatibility</td><td>Only small subset of AWS services</td></tr>
<tr><td><b>Pricing</b></td><td colspan="2">Pay as you go</td><td>Capex and opex</td></tr>
</table>

* = Virtualized

Related breakouts

CMP 404: Enhancing security and future-proofing your instances

CMP 327: AWS Graviton deep dive: The best price performance for your AWS workloads

CMP 312: Run high-performance storage workloads on EC2 storage optimized instances

CMP 306: Building apps to isolate & process sensitive data with AWS Nitro Enclaves

CMP 302: Confidential computing with AWS compute

CMP 225: What’s new in Amazon EC2

CMP 201: Silicon innovation at AWS

SEC 327: Zero-privilege operations: Running services without access to data

Thank you!

Ravi Murty, EC2, AWS

rmurty@amazon.com

Please complete the session survey in the **mobile app**