Introduction to Amazon Cloud

Amazon EC2 Overview

Eren Akbaba

Solutions Architect, AWS
Intstructor @Bilkent University

CS 443 – Cloud Computing Syllabus

- Week 1 - Introduction to Cloud Computing
- Week 2 - Service Models: IaaS, PaaS, SaaS and examples from enterprise for each
- Week 3 - Virtualization techniques (VMs vs Containers) HW 1
- Week 4 - Cloud compute services (VM, Containers, Kubernetes, Lambda, Microservices)
- Week 5 - Cloud Networking (VPC) HW 2
- Week 6 - Data Centers and Datacenter Networks + Midterm
- Week 7 - Cloud Storage - Replication; Distributed File Systems; Structured and unstructured storage
- Week 8 - Cloud Databases - SQL/NoSQL databases; HW 3
- Week 9 - Cloud Security
- Week 10 - Cloud Endpoints – APIs, API Gateway, Load Balancing HW 4
- Week 11 - Classical 3 Tier Web Applications
- Week 12 - DevOPS HW 5
- Week 13 - Machine learning and Generative AI
- Week 14 - Project Presentations – Demos – Make Up Week

aws

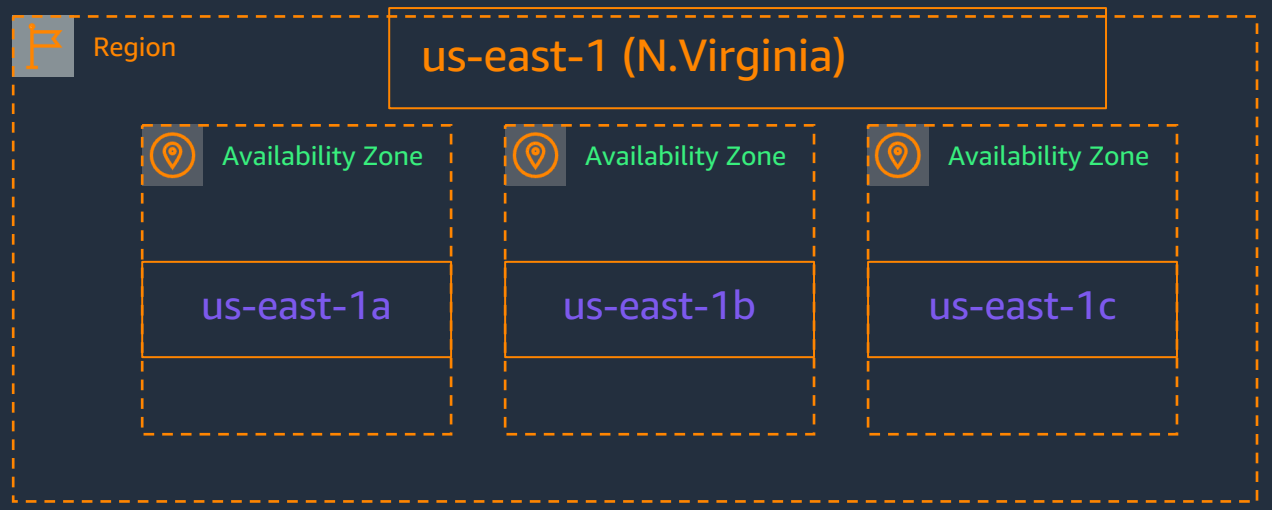
Agenda

- Introduction to Amazon Cloud
- AWS Global Reach
- Amazon EC2 Overview
- Amazon EC2 Design

32 Regions

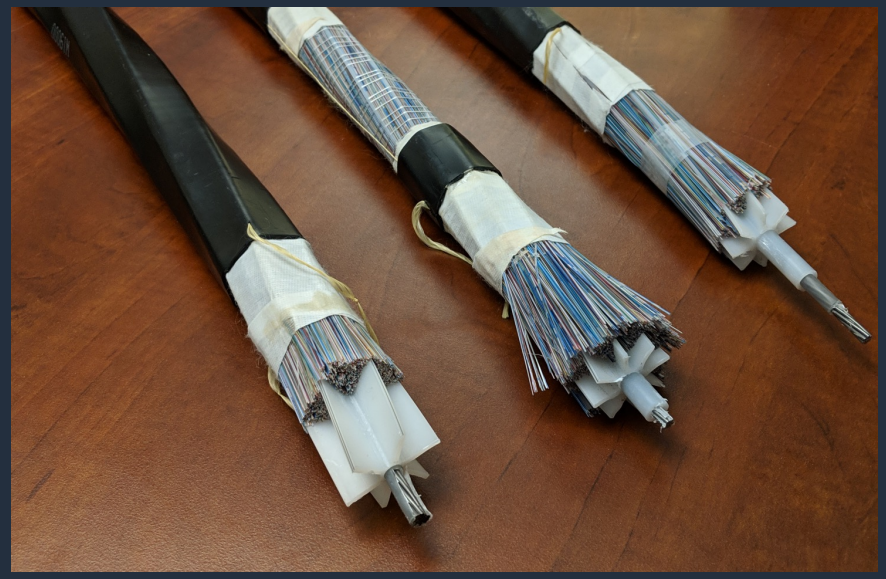
Availability Zones

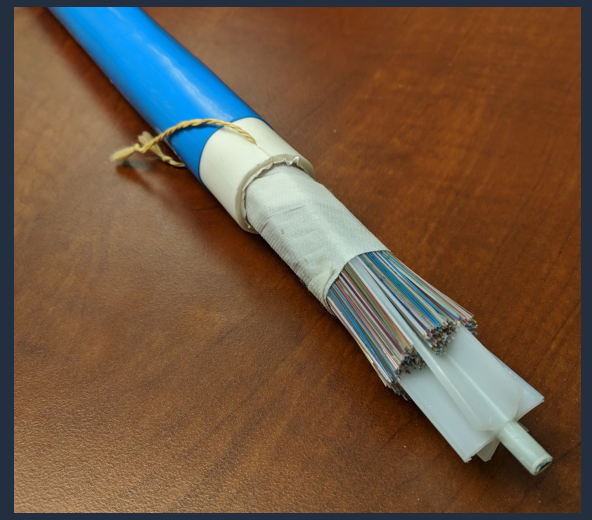
- Each AWS Region consists of multiple, isolated, and physically separate AZs within a geographic area
- An Availability Zone (AZ) is one or more discrete data centers with redundant power, networking, and connectivity in an AWS Region
- High throughput, low latency (< 10 ms) network between Availability Zones
- All traffic between AZs is encrypted
- Physical separation with 100 km (60 miles)



Intra & inter-AZ connectivity

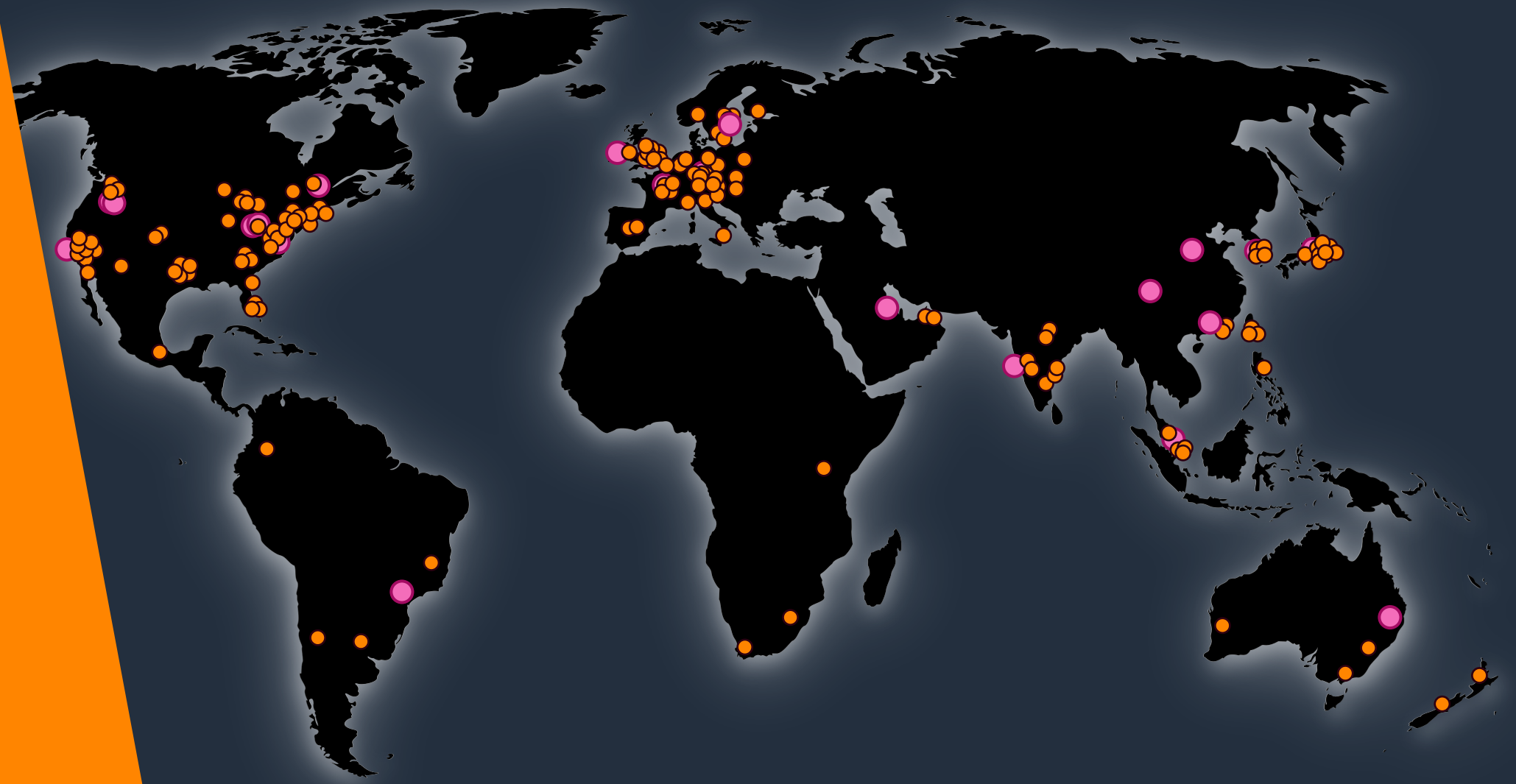
- Dark fiber “spans”
  - Optimized for low-latency & physical diversity
- Amazon controlled infrastructure
- Geospatial coordinates
- Dense wavelength division multiplexing (DWDM)





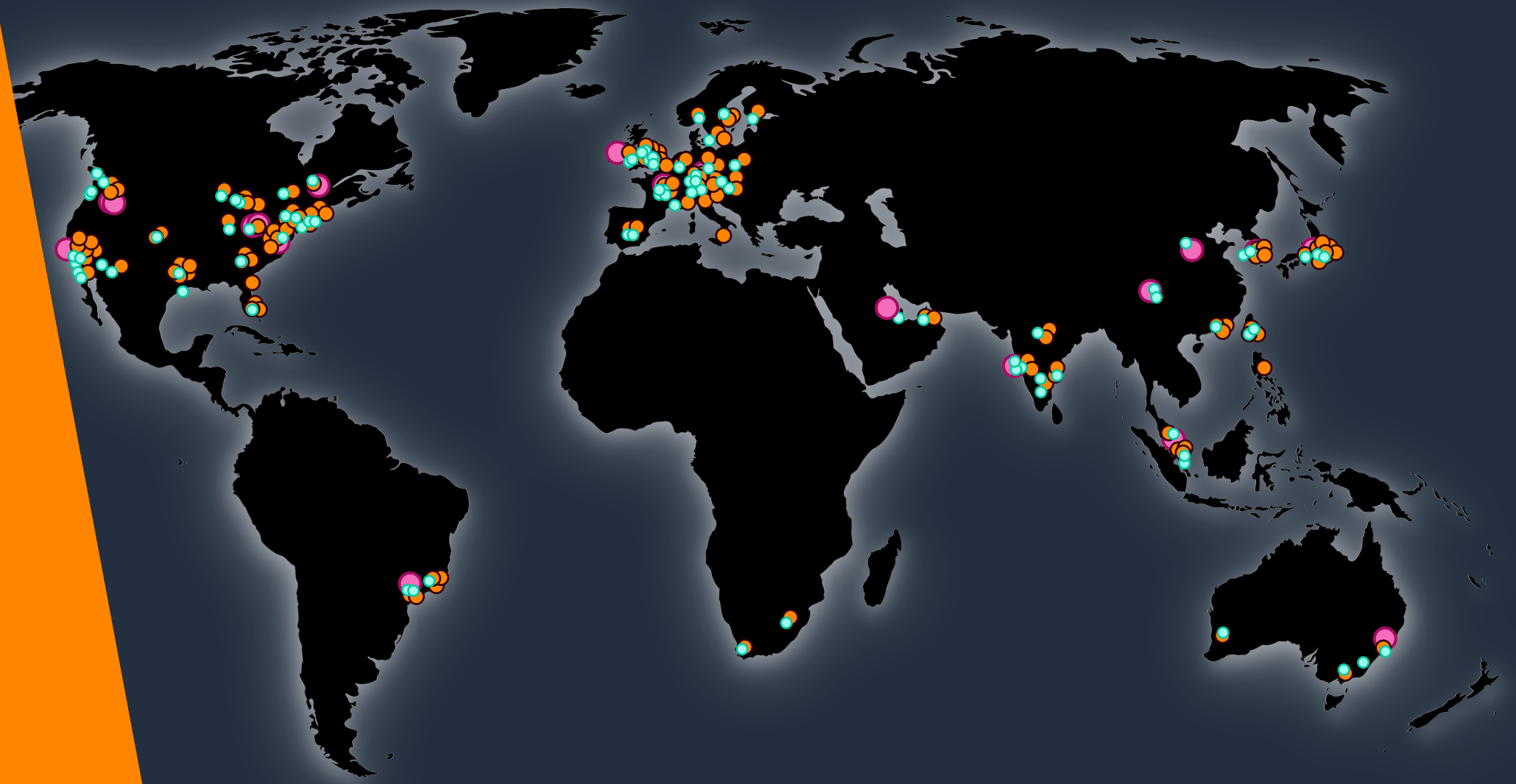
550+

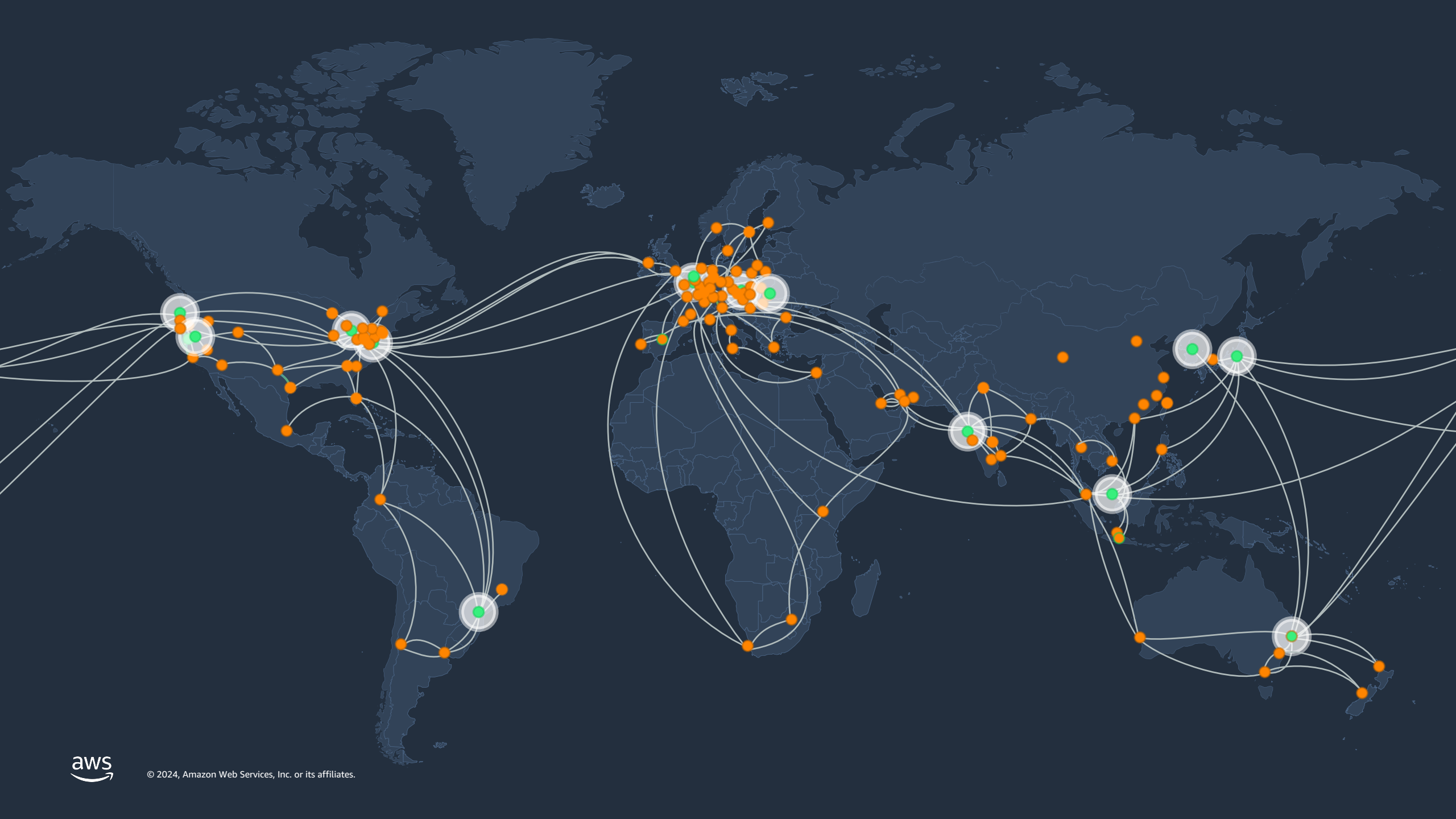
Amazon CloudFront Points of Presence

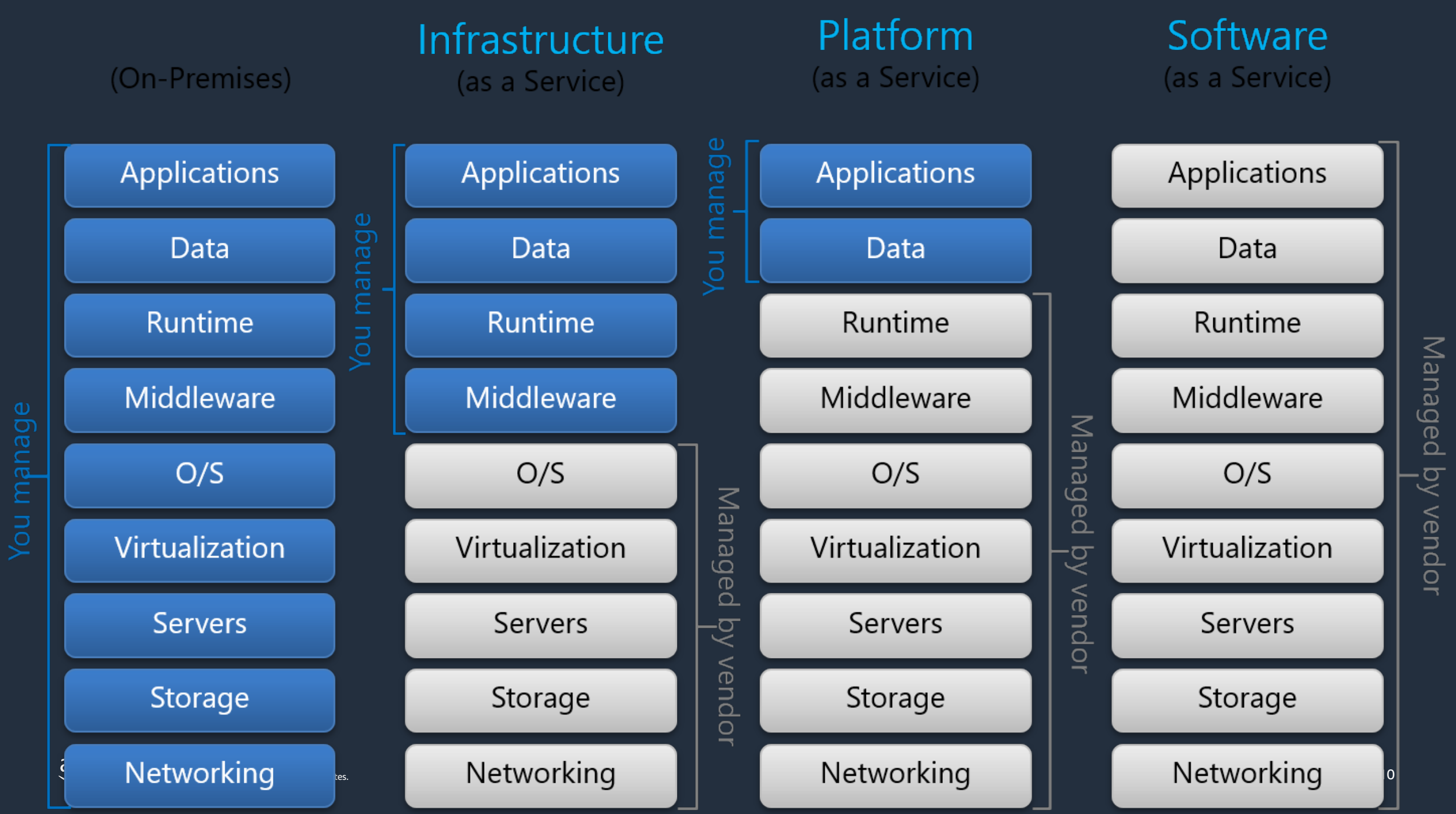


115

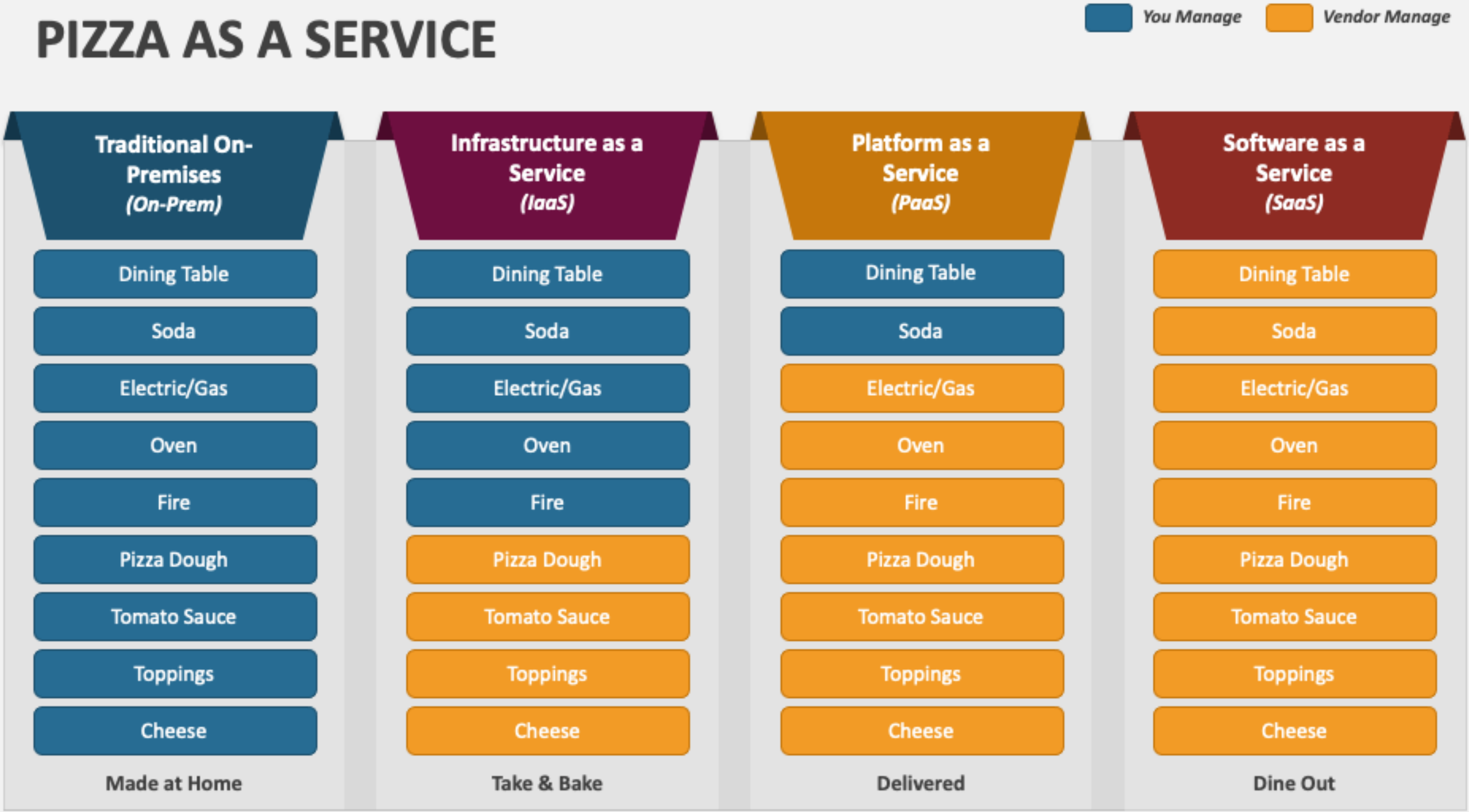
AWS Direct Connect locations



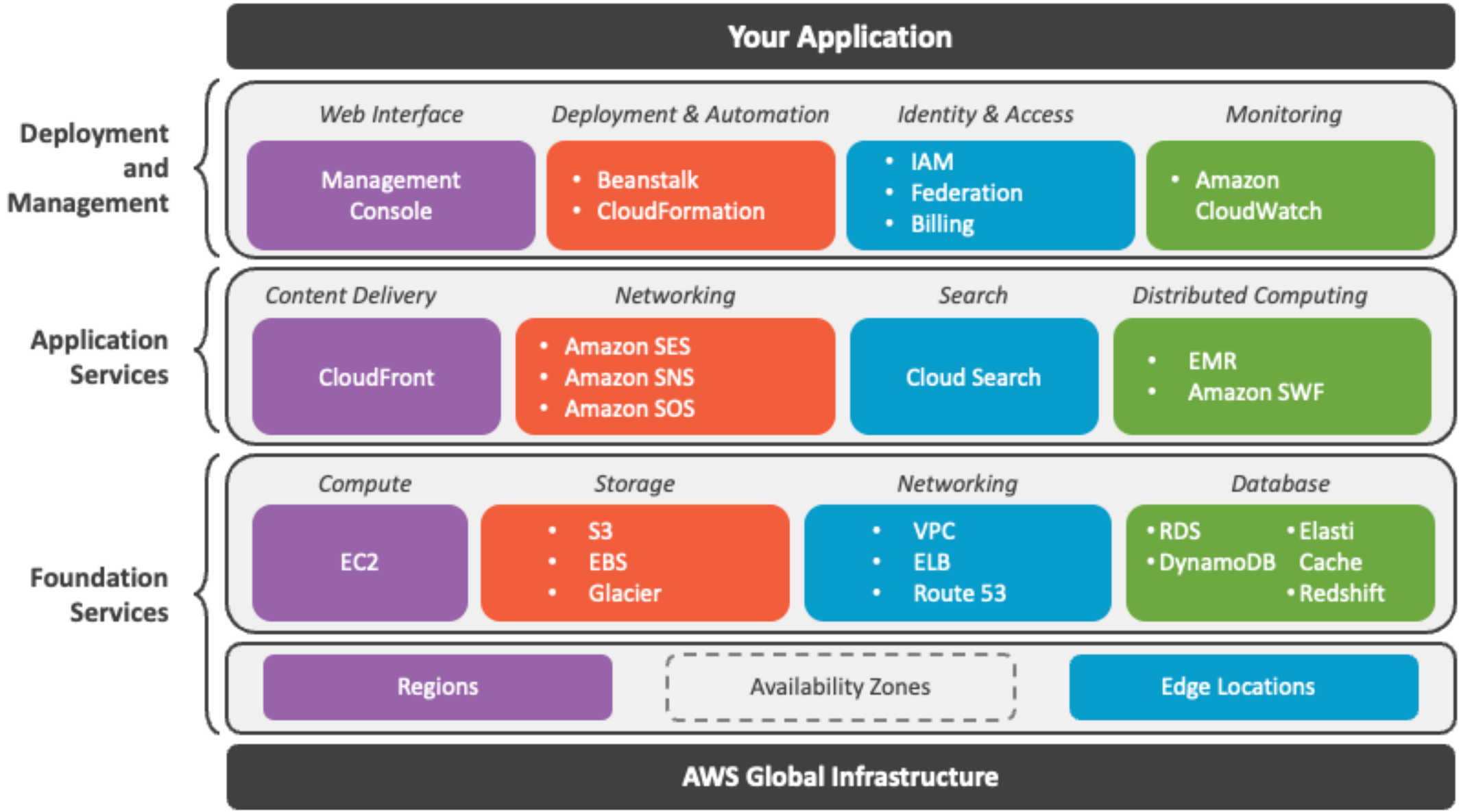



(On-Premises)
Infrastructure
(as a Service)
Platform
(as a Service)
Software
(as a Service)
Applications
Data
Runtime
Middleware
O/S
Virtualization
Servers
Storage
Networking
You manage
Managed by vendor

PIZZA AS A SERVICE

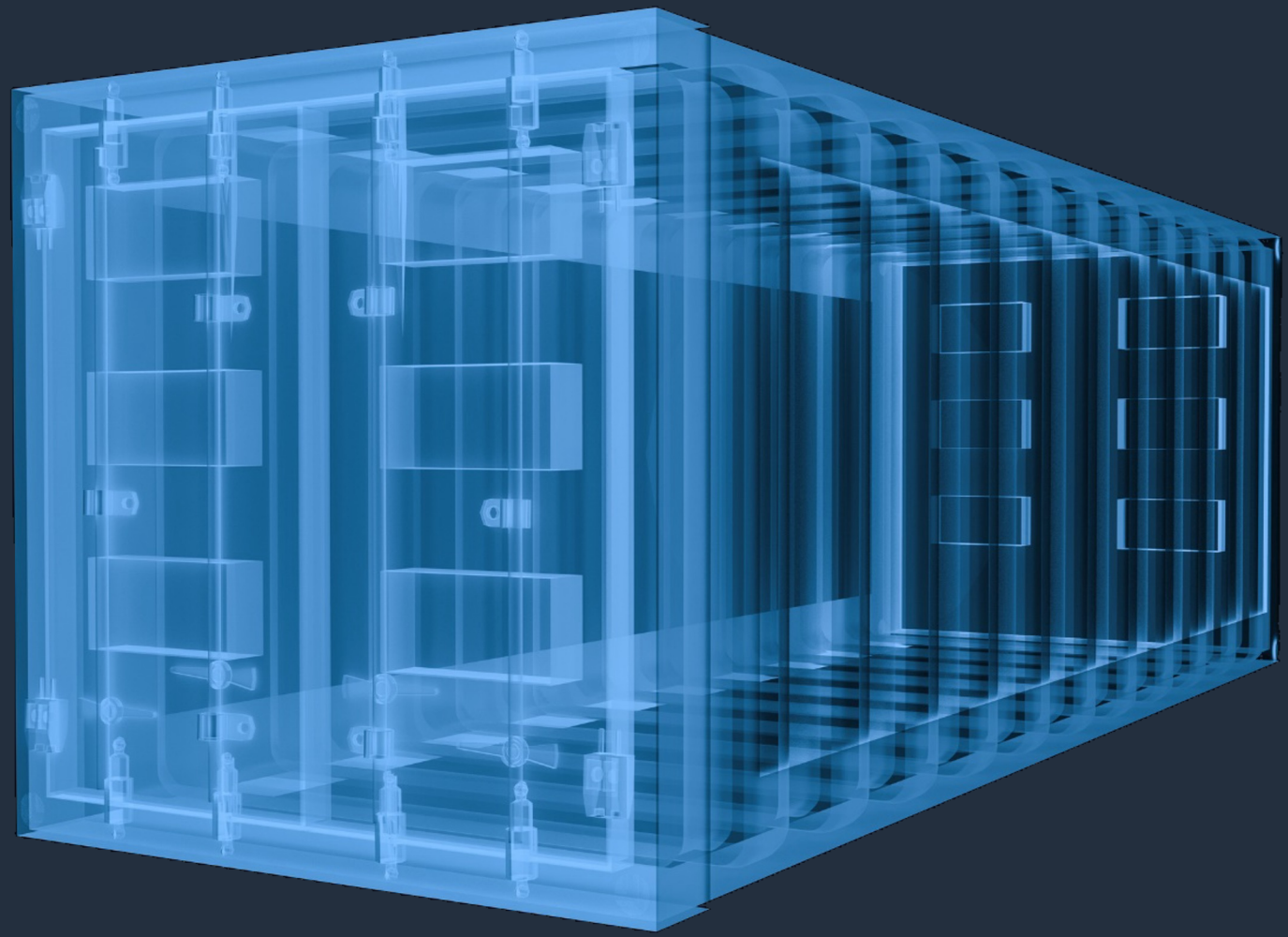


AWS INFRASTRUCTURE



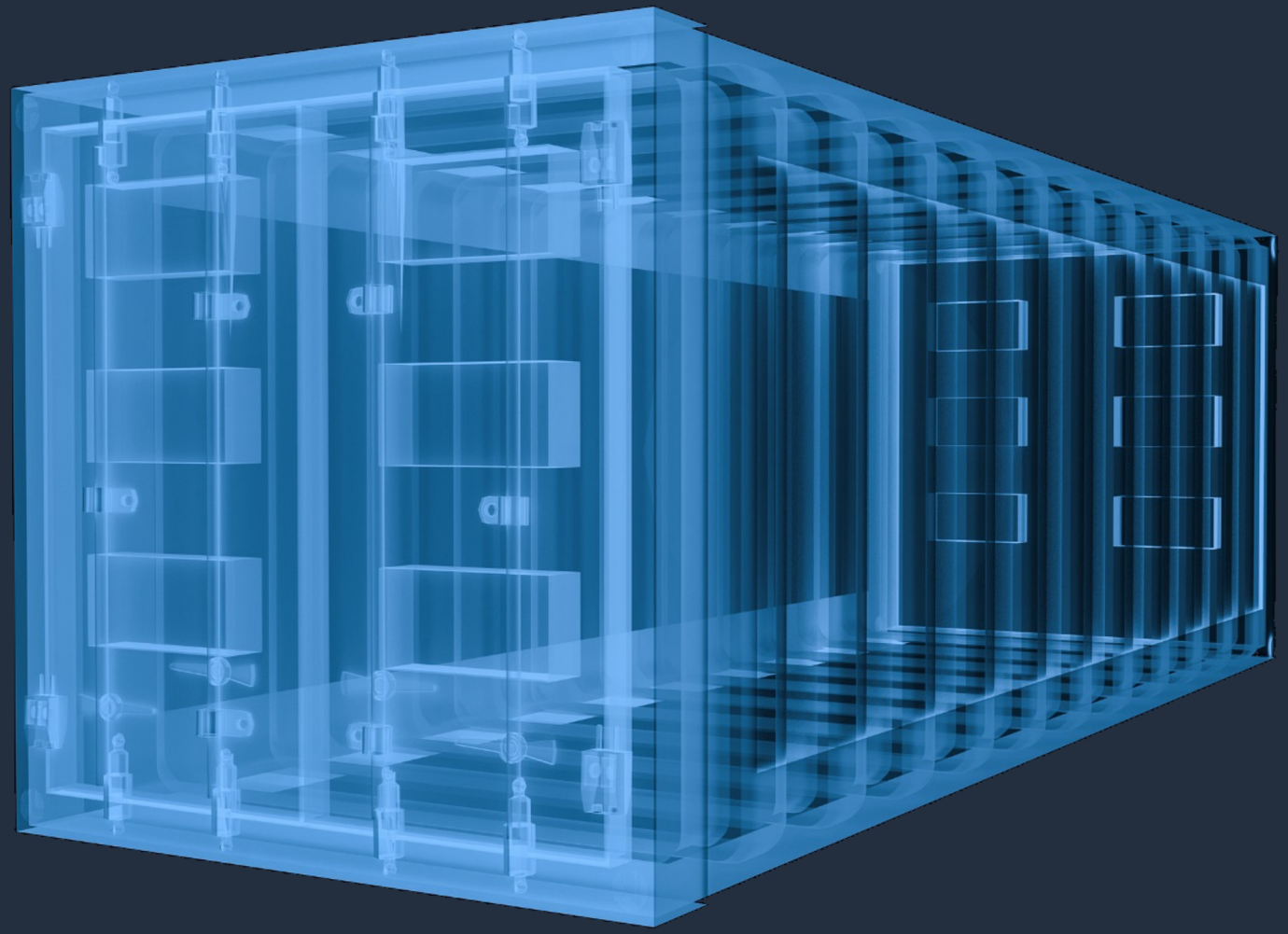
First things first…

- What are containers and why are customers using them?

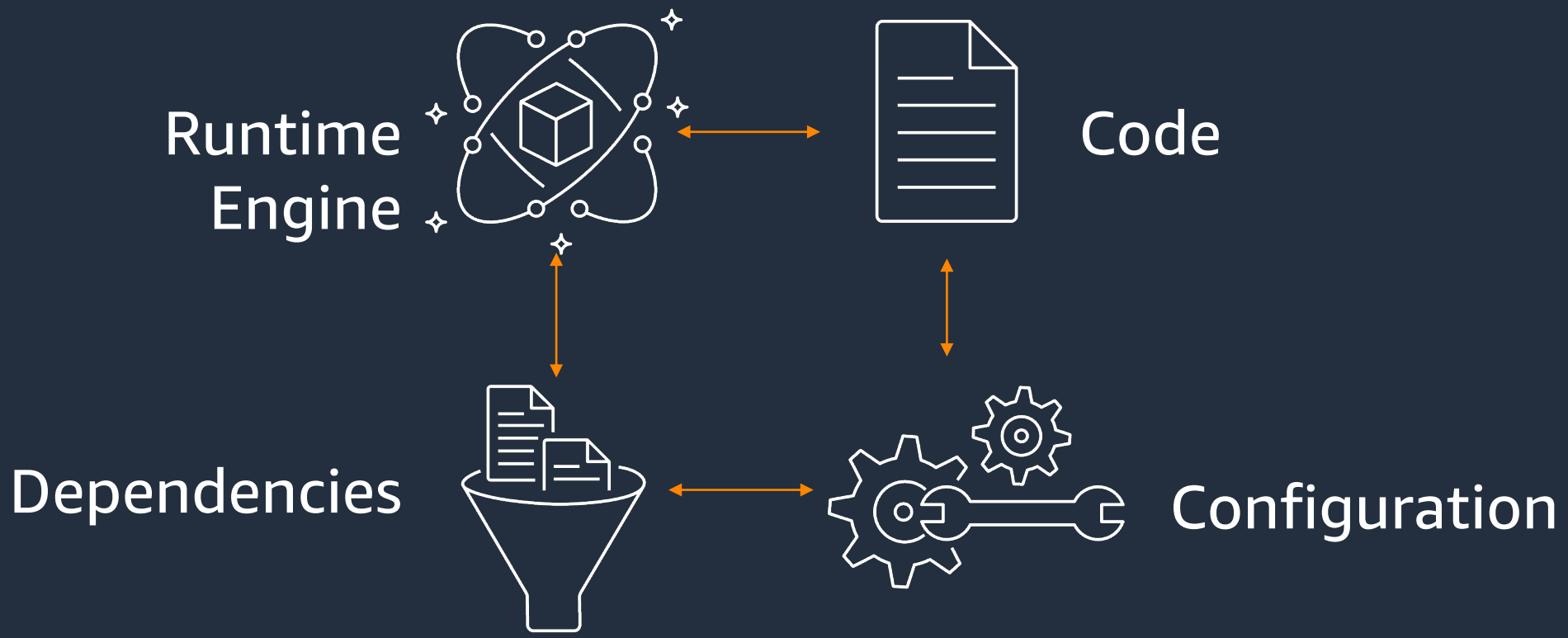


Why are companies adopting containers?

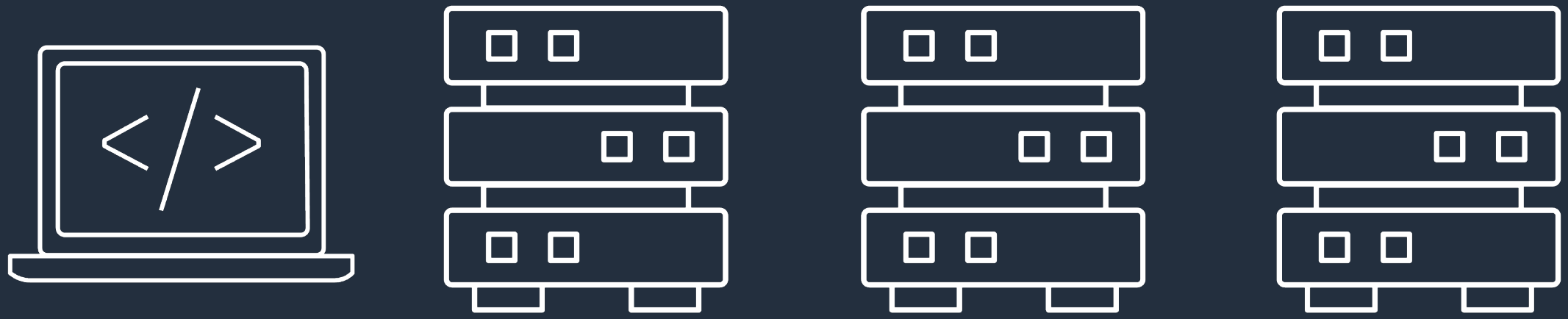
- Accelerate software development
- Build modern applications
- Automate operations at web scale



Application environment components



Different environments



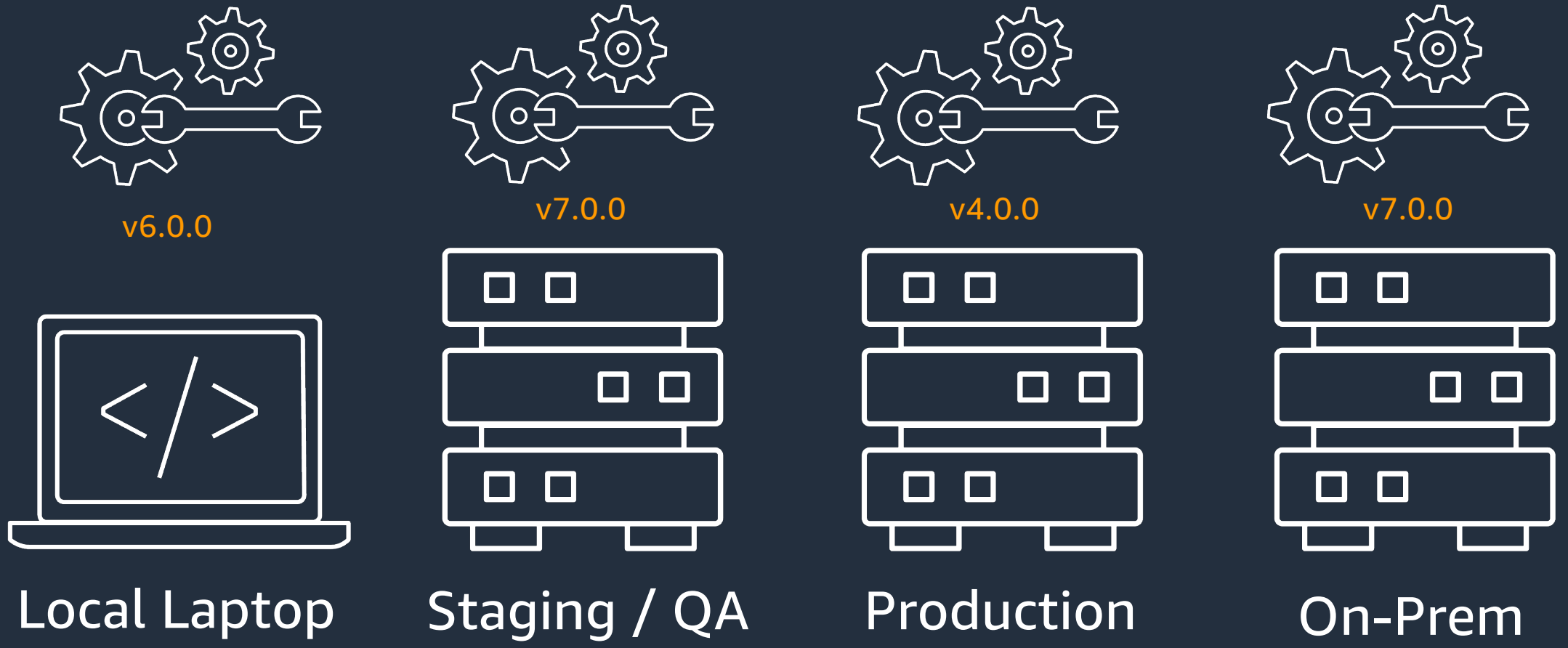
Local Laptop

Staging / QA

Production

On-Prem

It worked on my machine, why not in prod?

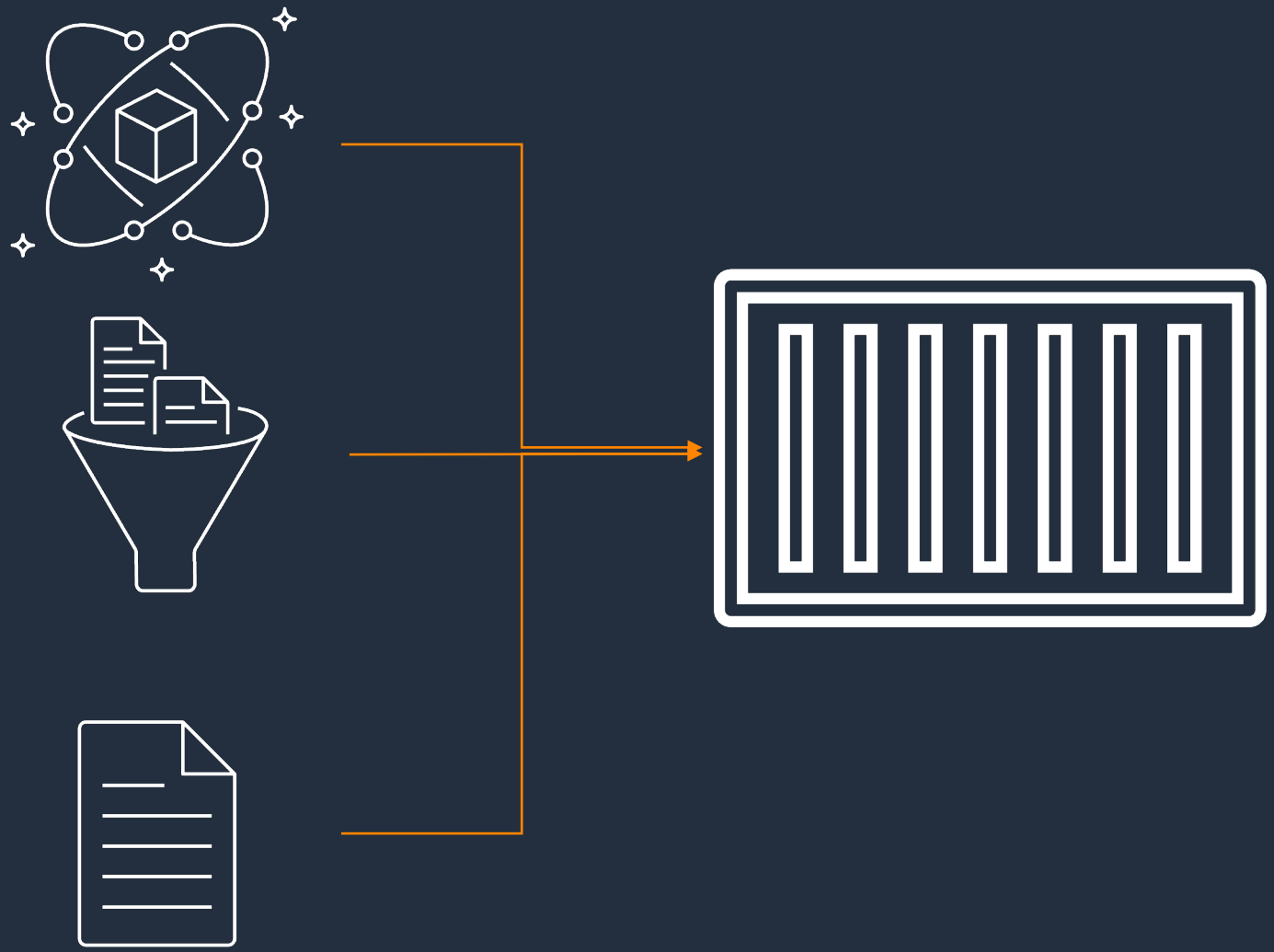


Containers to the rescue

Runtime Engine

Dependencies

Code




APM TERMINALS
APM TERMINALS
APM TERMINALS
58
57
54
KALMAR
58
54

What is Docker?

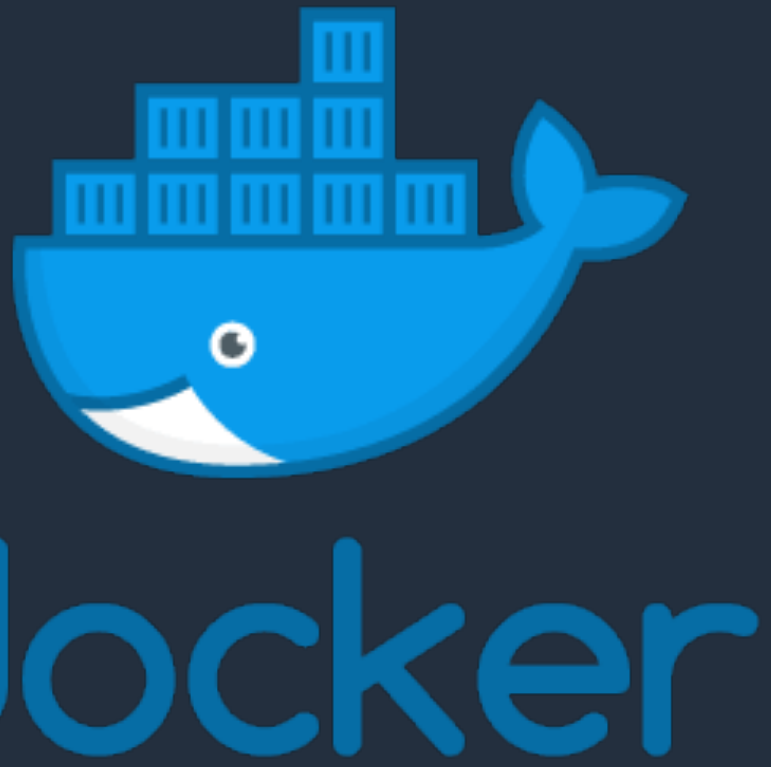
Lightweight container virtualization platform.

Ecosystem of tools to manage and deploy your applications

Licensed under the Apache 2.0 license.

Built by Docker, Inc.

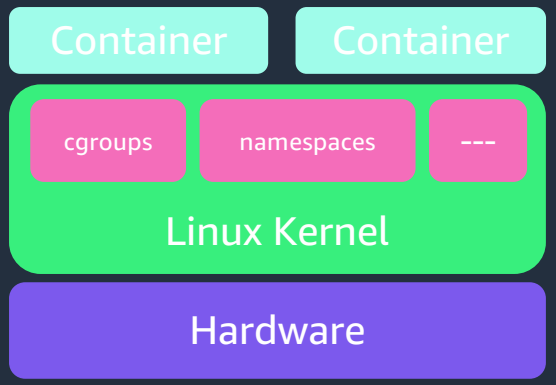
Moby: Open source project



Containers vs VMs

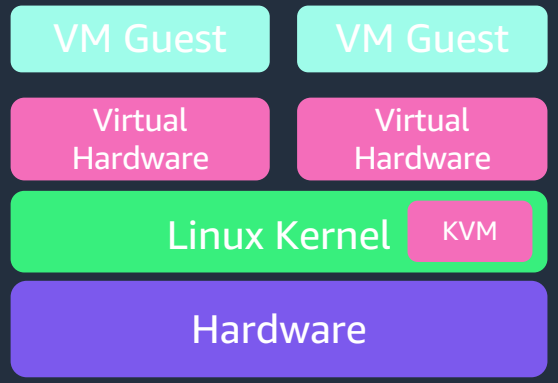
Containers

- Using Linux primitives for isolation
- Share Linux Kernel
- Fast starts, minimal overhead
- Flexible isolation



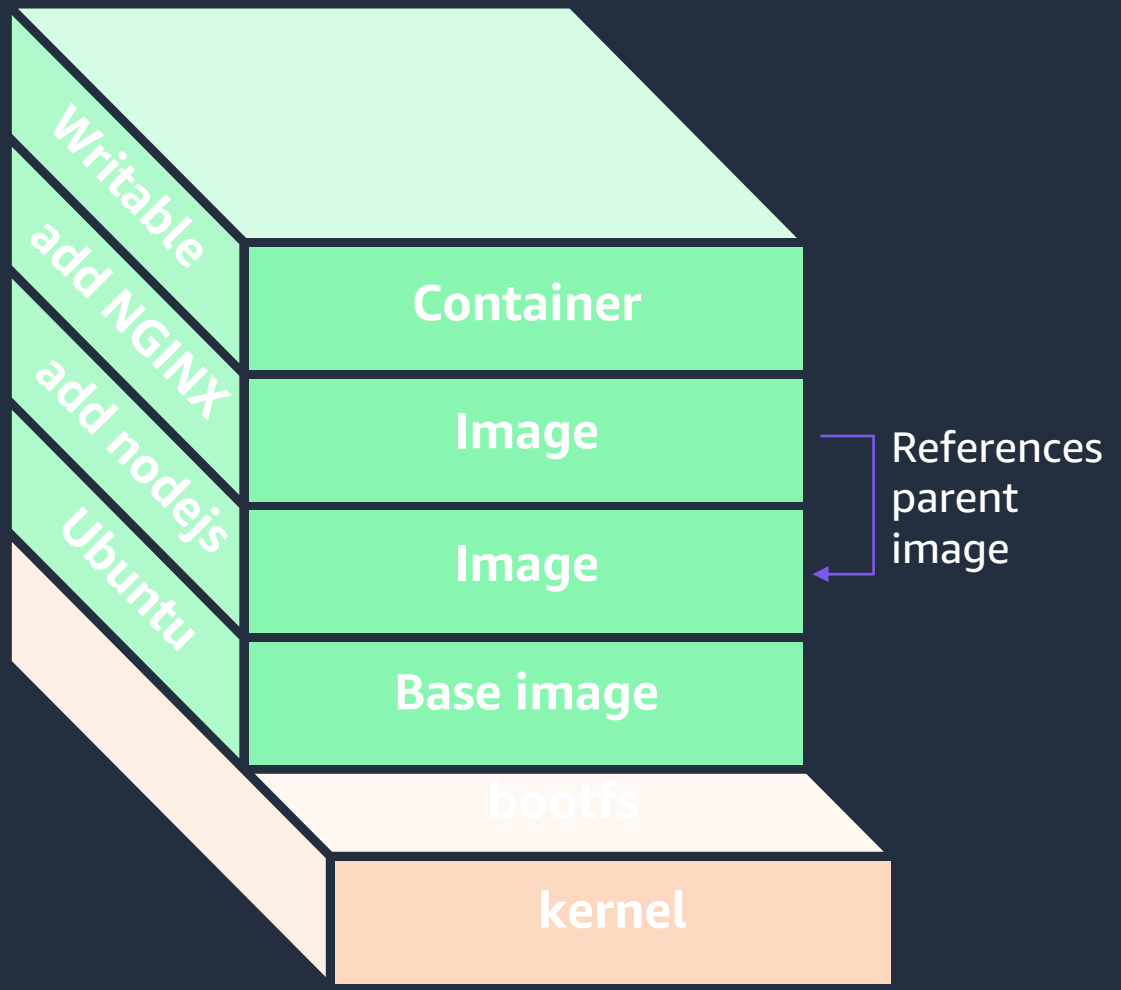
Virtual Machines

- Virtualisation or emulate hardware components
- Completely separate kernels (maybe not Linux)
- Slower starts, must boot kernel and set-up hardware.



Container images

- Read only image that is used as a template to launch a container.
- Start from base images that have your dependencies, add your custom code.
- Dockerfile for easy, reproducible builds.



Four environments, same container

Local Laptop

Staging / QA

Production

On-Prem

Container benefits

Runs reliably everywhere

Run different apps simultaneously

Better resource utilization

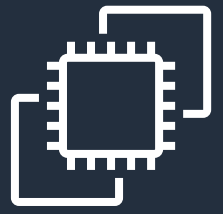
AWS Compute Offerings

Service	 Amazon EC2	 Amazon ECS	 AWS Fargate	 AWS Lambda
How do I choose?	I want to configure servers, storage, networking, and my OS	I want to run servers, configure applications, and control scaling	I want to run my containers	Run my code when it's needed

Amazon EC2 Overview

Choices for Compute

World-class performance, security, and innovation



AMAZON EC2

Virtual server instances in the cloud

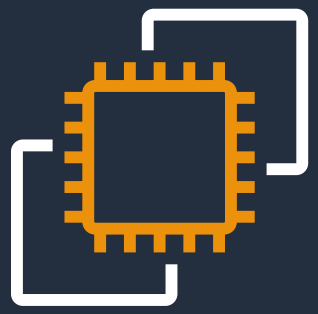
AMAZON ECS, EKS, and FARGATE*

Container management service for running Docker on a managed cluster of EC2

AWS LAMBDA

Serverless compute for stateless code execution in response to triggers

Amazon Elastic Compute Cloud (Amazon EC2)



Linux | Windows | Mac

Arm and x86 architectures

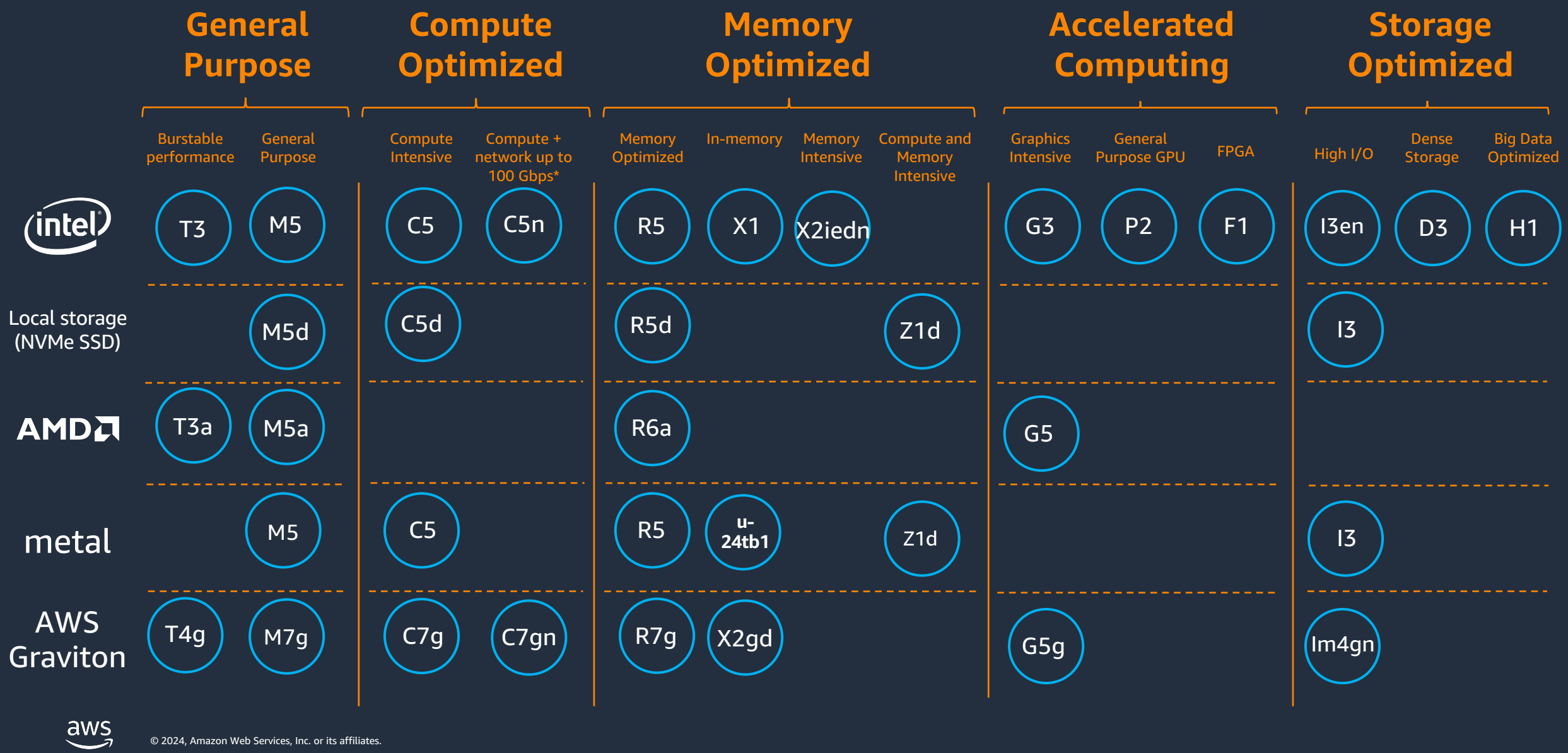
General purpose and workload optimized

Bare metal, disk, networking capabilities

Packaged | Custom | Community AMIs

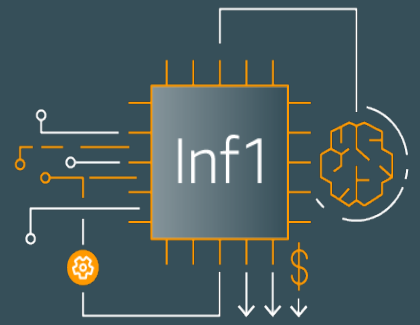
Multiple purchase options: On-Demand, Spot instances, Reserved Instances, Savings Plans, Dedicated Hosts

Instance Types



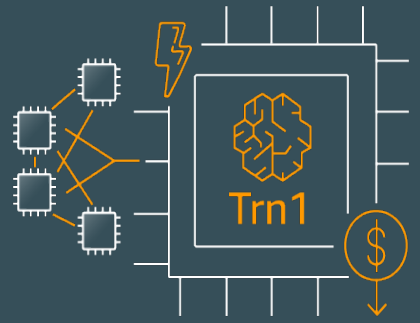
AWS chips optimized for deep learning

AWS Inferentia



Lowest cost inference in the cloud for running deep learning models—up to 70% lower cost than GPU instances

AWS Trainium



The most cost-efficient high performance DL training instance

Instance Naming

Instance generation

Instance Sizing

c7gn.8xlarge

2 – c7gn.4xlarge

4 – c7gn.2xlarge

8 – c7gn.xlarge

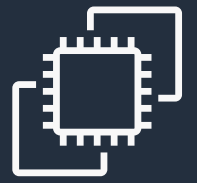
Choose your processor and architecture

Intel® Xeon® Scalable (Skylake) processor

NVIDIA V100 Tensor Core GPUs

AMD EPYC processor

AWS Graviton Processor (arm)



FPGAs for custom hardware acceleration

Right compute for the right application and workload

AWS Graviton Processor

Enabling the best price/performance for your cloud workloads

Graviton2 Processor

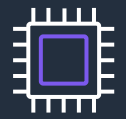
7x performance, 4x compute cores, and 5x faster memory

Built with 64-bit Arm Neoverse cores with AWS-designed silicon using 7 nm manufacturing technology

Up to 64 vCPUs, 25 Gbps enhanced networking, 19 Gbps EBS bandwidth

Graviton3/3E Processor

25% higher performance, 2x higher floating-point performance, 2x faster cryptographic performance



DDR5 memory provides 50% more memory bandwidth compared to DDR4

Support for bfloat16 and delivers up to 3x better performance for ML workloads

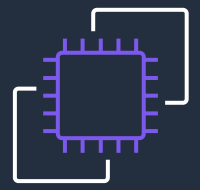
What's a virtual CPU? (vCPU)

- A vCPU is typically a hyper-threaded physical core*
- Divide vCPU count by 2 to get core count
- On Linux, "A" threads enumerated before "B" threads
- On Windows, threads are interleaved

- Cores by Amazon EC2 & RDS DB Instance type: https://aws.amazon.com/ec2/physicalcores/

* *CPU Optimizing options allow disabling hyperthreading and reduce number of cores*

Choice of accelerators for specialized workloads



Elastic Graphics

Easily add graphics acceleration to your EC2 instance

Configure right amount of graphics acceleration for your workload

Accelerate application for fraction of cost of standalone graphics instances

Elastic Inference

Reduce deep learning inference costs by up to 75%

Easily attach fractional sizes of a full GPU instance to EC2 or SageMaker instances

Scale inference acceleration up or down as needed with EC2 Auto Scaling

Broadest and deepest platform choice

Categories	Capabilities	Options
General purpose	Choice of processor (AWS Graviton, Intel, AMD)	Elastic Block Store (EBS)
Burstable	Fast processors (up to 4.5 GHz)	Elastic Fabric Adapter
Compute intensive	High memory footprint (up to 24 TiB)	Elastic Inference
Memory intensive	Instance storage (HDD and SSD)	Elastic Graphics
Storage (High I/O)	Accelerated computing (GPUs ,FPGA & ASIC)	Linux, Unix, Windows, macOS
Dense storage	Networking (up to 800 Gbps)	
GPU compute	Bare Metal	
Graphics intensive	Size (Nano to 48xlarge)	

650+

instance types

for virtually every workload and business need

Memory and Storage

What's a GiB?

- Memory is presented as GibiBytes (GiB) and not Gigabytes (GB)
- 256 GiB = 275 GB

What about storage?

- Storage is independent of compute
- You allocate drives known as Amazon Elastic Block Store (EBS) volumes
- Amazon EBS volumes support up to 64 TiB per volume
- Some instance types provide physically attached (ephemeral) storage

EC2 Operating Systems

- Windows Server 2012/2012 R2/2016/2019/2022
- Amazon Linux (NEW: Amazon Linux 2023)
- Debian
- SUSE
- CentOS
- Red Hat Enterprise Linux (RHEL)
- Ubuntu
- Mac, including M1 Mac instances

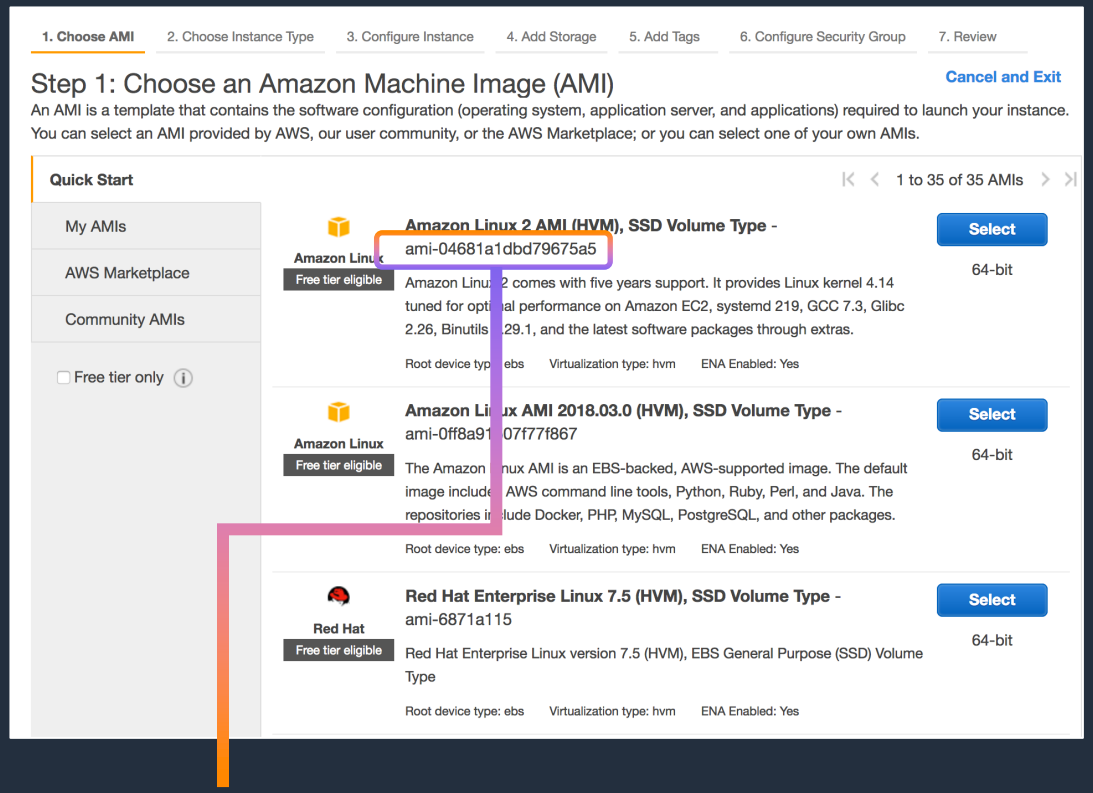
Visit the AWS Marketplace for more Operating Systems

What is an Amazon Machine Image (AMI)?

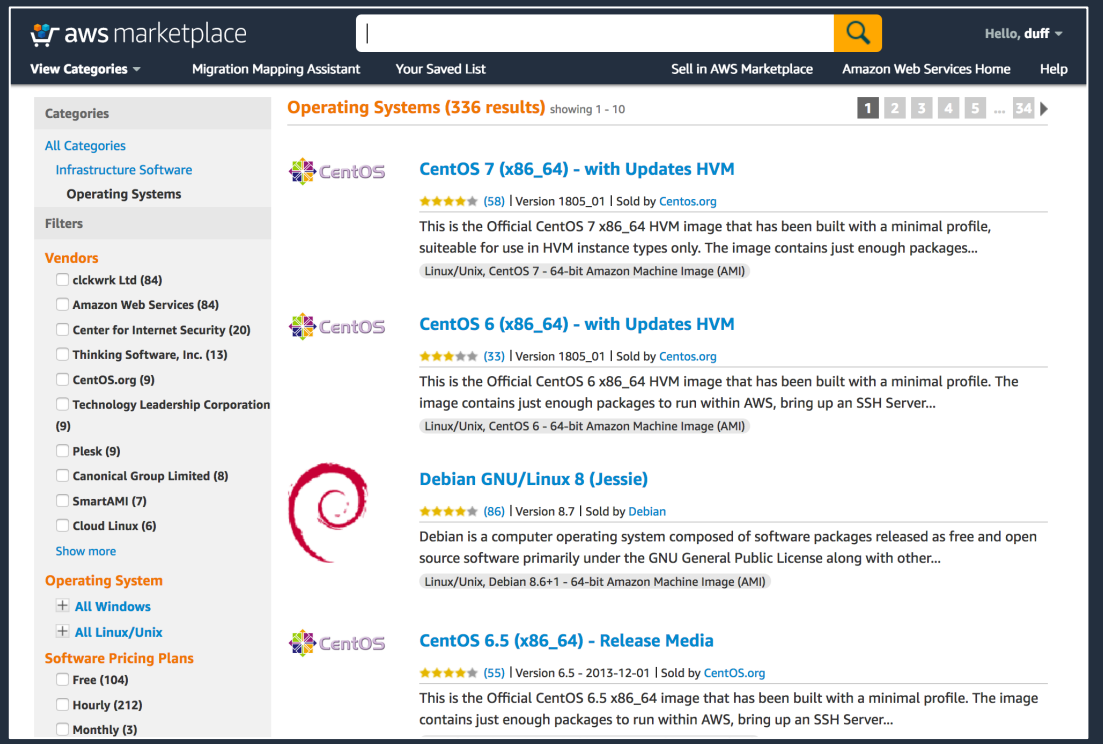
- Provides the information required to launch an instance
- Launch multiple instances from a single AMI with the same configuration
- An AMI includes the following:
  - One or more Amazon Elastic Block Store (Amazon EBS) snapshots, or a template for the root volume (operating system, applications)
  - Launch permissions that control which AWS accounts can use the AMI
  - Block device mapping that specifies volumes to attach to the instance

Choosing an AMI

AWS Console



AWS Marketplace



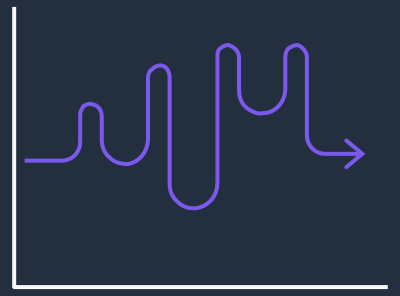
Use the AMI ID to launch through the API or AWS Command Line Interface (AWS CLI)

```
aws ec2 run-instances --image-id ami-04681a1dbd79675a5 --instance-type c4.8xlarge --count 10 --key-name MyKey
```

Amazon EC2 purchase options

On-Demand

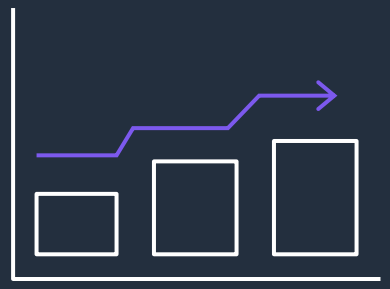
Pay for compute capacity by **the second** with no long-term commitments



Spiky workloads, to define needs

Reserved Instances

Make a 1 or 3 year commitment and receive a **significant discount** off On-Demand prices



Committed and steady-state usage

Savings Plans

Same great discounts as Amazon EC2 RIs with **more flexibility**

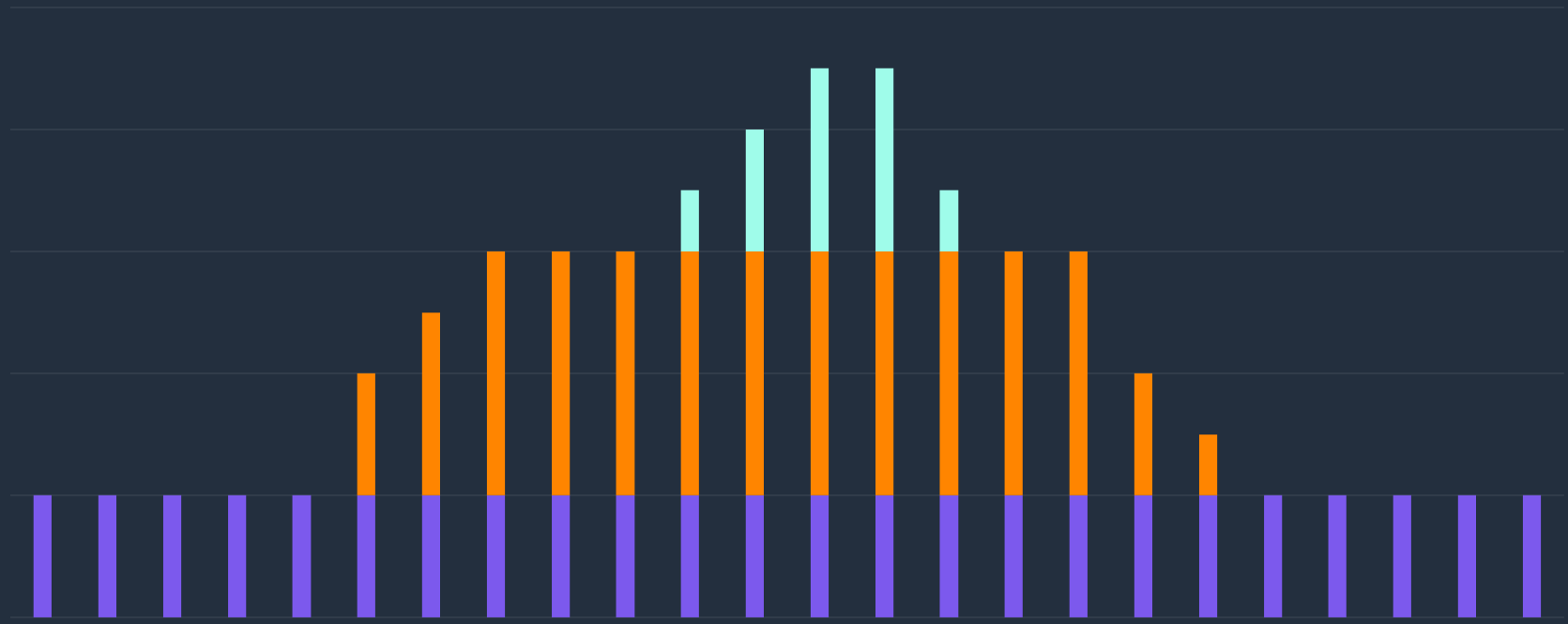
Committed flexible access to compute

Spot Instances

Spare Amazon EC2 capacity at **savings of up to 90%** off On-Demand prices

Fault-tolerant, flexible, stateless workloads

Simplifying capacity and cost optimization



SCALE USING **SPOT** FOR FLEXIBLE, FAULT-TOLERANT WORKLOADS

SCALE USING **ON-DEMAND** FOR NEW OR STATEFUL SPIKY WORKLOADS

USE **RIs** AND **SAVINGS PLANS** FOR KNOWN/ STEADY-STATE WORKLOADS

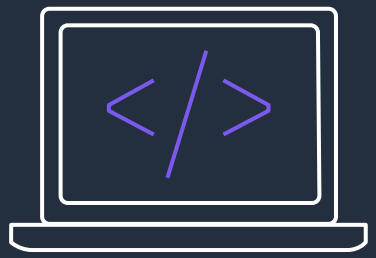
AWS services make this easy and efficient

- Amazon EC2 Auto Scaling
- EC2 Fleet
- Amazon Elastic Container Service (Amazon ECS)
- Amazon Elastic Kubernetes Service (Amazon EKS)
- AWS Thinkbox
- Amazon EMR
- AWS CloudFormation
- AWS Batch

Hibernate Amazon EC2 Instances

Maintain a fleet of pre-warmed instances to quickly get to a productive state

Available with Amazon EBS-backed instances



Use familiar Stop and Start APIs

Memory data saved in EBS root volume

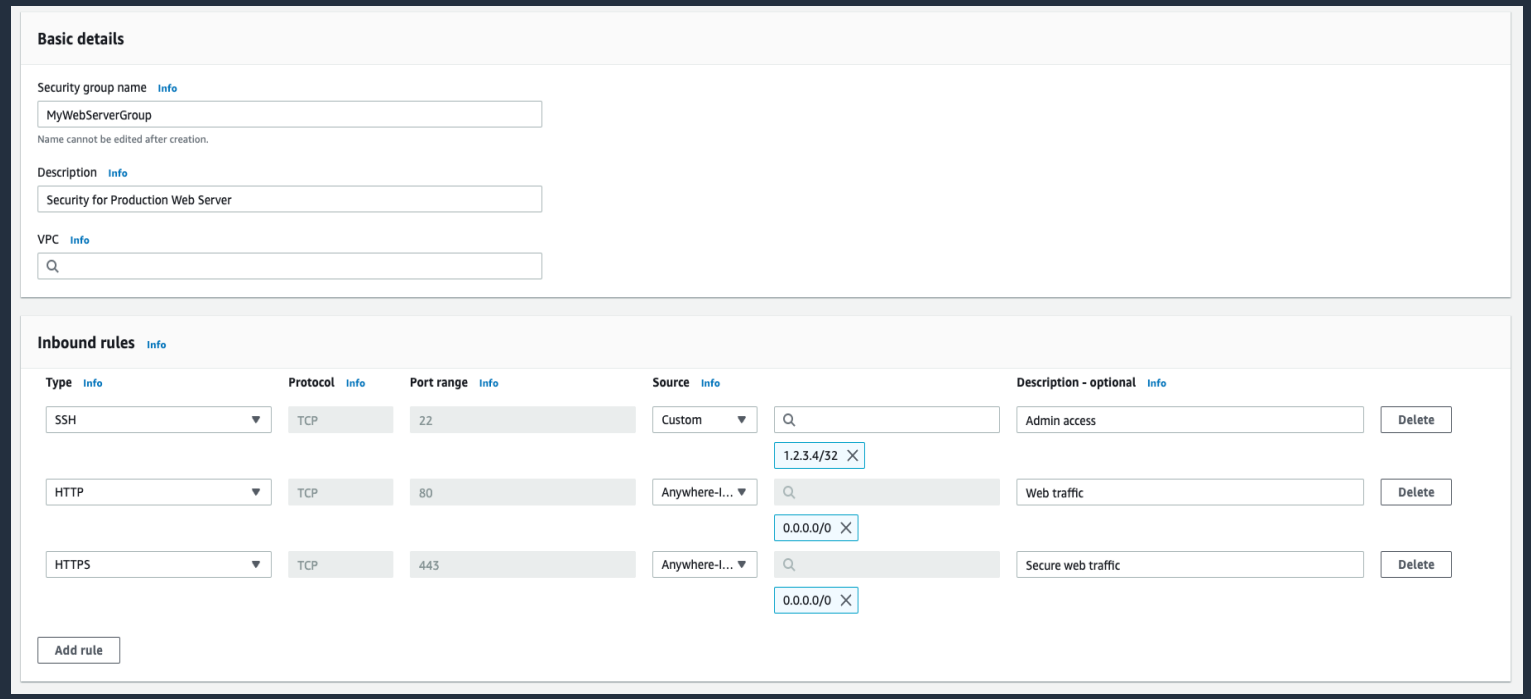
RAM contents are encrypted on EBS

Its just like closing and opening your laptop!

Applications can pick up right where it left off

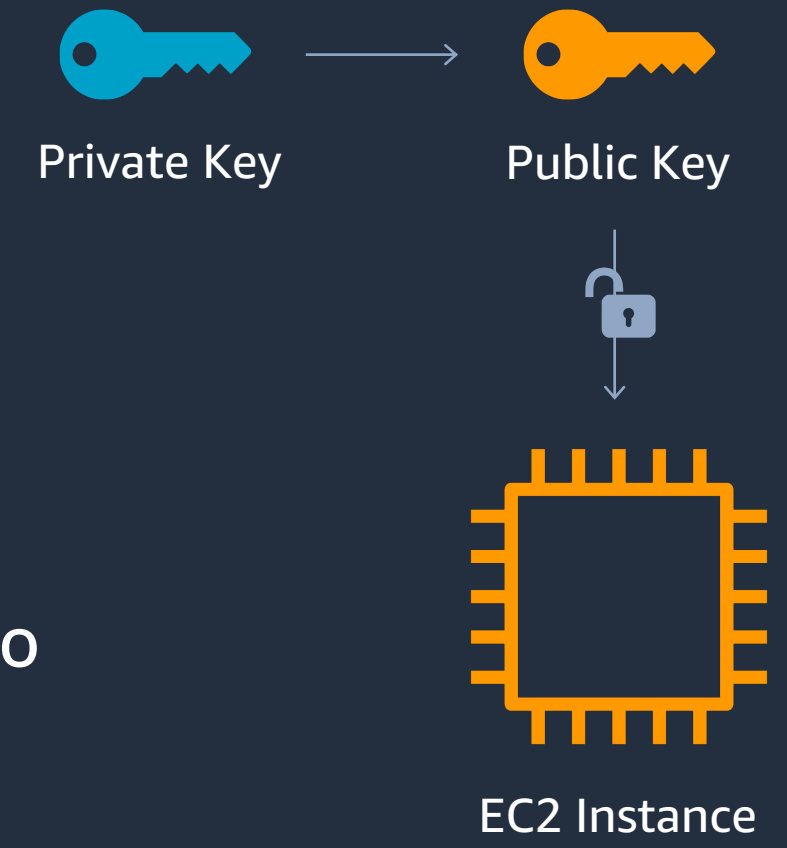
EC2 Security Groups

- Virtual firewall
- Security Group Rules
  - Security Group name
  - Description
  - Protocol
  - Port range
  - IP address, IP range



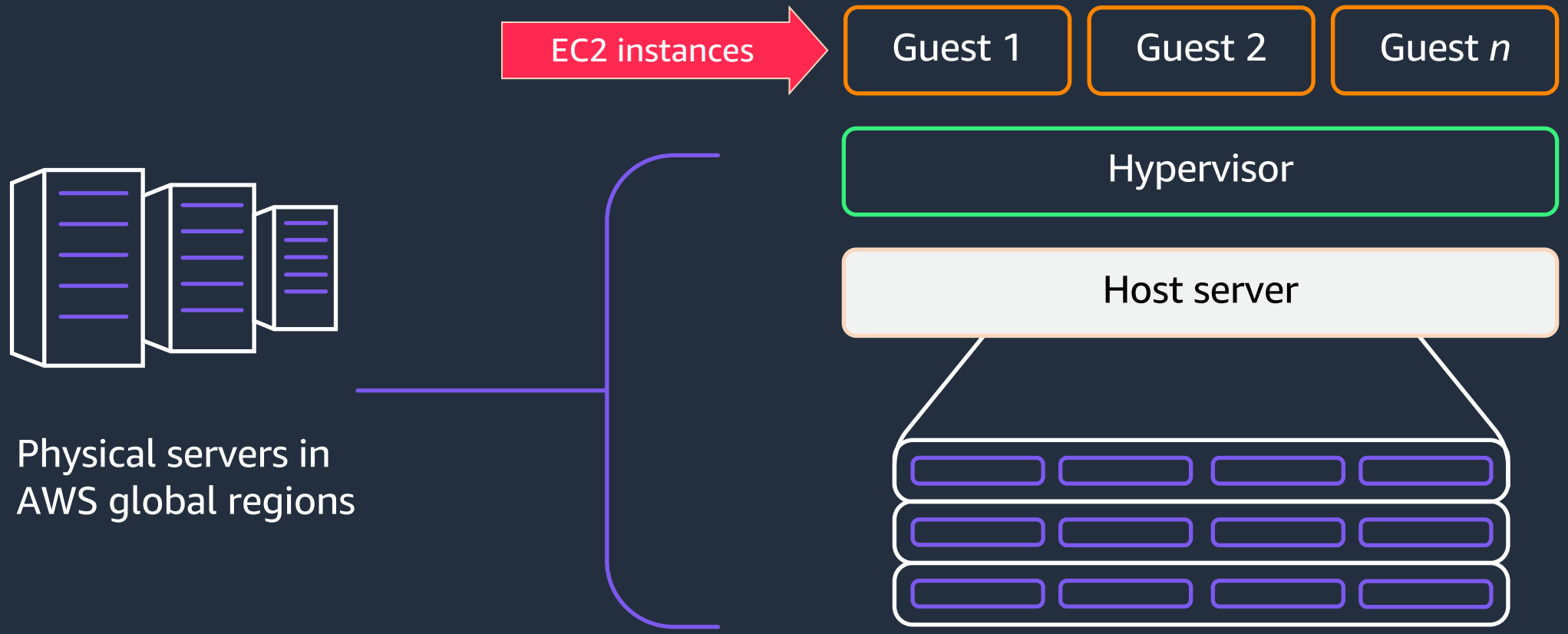
EC2-Specific Credentials

- EC2 key pairs
  - Linux – SSH key pair for first-time host login
  - Windows – Retrieve Administrator password
- Standard SSH RSA key pair
  - Public/Private Keys
  - Private keys are not stored by AWS
- AWS approach for providing **initial** access to a generic OS
  - Secure
  - Personalized
  - Non-generic (NIST, PCI DSS)



Amazon EC2 Design

EC2 Host Virtualization



Resource allocation

- All resources assigned to you are dedicated to your instance with no over commitment*
  - All vCPUs are dedicated to you
  - Memory allocated is assigned only to your instance
  - Network resources are partitioned to avoid “noisy neighbors”
- Curious about the number of instances per host?
  - See “Dedicated Hosts Configuration Table” for a guide.

*the “T” family is special

Which hypervisor do we use?

Original host architecture: Xen-based

- Hypervisor consumed resources from the underlying host
- Limited optimization

AWS Nitro Hypervisor: Custom KVM based hypervisor

- AWS Nitro System (launched on Nov 2017)
- Less server resources used, more resources for the customer
- AWS optimized

Bare metal: Direct access to processor and memory resources

- Built on the AWS Nitro system
- Enables custom hypervisors and micro-VM runtimes

AWS Nitro System

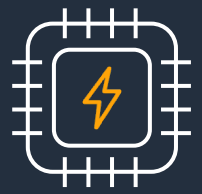
Nitro Card

Local NVMe storage

Elastic Block Storage

Networking, monitoring, and security

Nitro Security Chip



Integrated into motherboard

Protects hardware resources

Nitro Hypervisor

Lightweight hypervisor

Memory and CPU allocation

Bare metal-like performance

Modular building blocks for rapid design and delivery of **Amazon EC2** instances

Thank you!