# Introduction to Amazon Cloud & EC2 Overview

Shibu Nair

October 16, 2020

# Agenda

- Introduction to AWS Cloud
- Global Reach
- EC2 Overview
- EC2 Details

# What is AWS?

AWS provides a highly reliable, scalable, low-cost infrastructure platform in the cloud that powers millions of businesses in over 190 countries around the world.

Benefits

- Low Cost
- Elasticity & Agility
- Open & Flexible
- Secure
- Global Reach

# What sets AWS apart?

| | |
|---|---|
| Experience | Building and managing cloud since 2006 |
| Service Breadth & Depth | 200+ services to support any cloud workload |
| Pace of Innovation | History of rapid, customer-driven releases |
| Global Footprint | 24 regions, 76 availability zones, 200+ edge locations |
| Pricing Philosophy | 77 proactive price reductions to date |
| Ecosystem | Thousands of consulting/system integrator & technology partners |

# Experience with Operational Reliability

Our goal is to make our operational performance indistinguishable from perfect. We are driven to remove any all causes of failure.

- We have spent over a decade building the world's most reliable, secure, scalable, and cost-effective infrastructure.
- Service SLAs between 99.9% and 100% availability. Amazon S3 is designed for 99.999999999% durability.
- Availability Zones exist on isolated fault lines, flood plains, and electrical grids to substantially reduce the chance of simultaneous failure.
- The AWS Service Health Dashboard provides 24/7 visibility in the real-time operational status of all services around the globe.



aws

# Pricing Philosophy

High volume / low margin businesses are in our core DNA

| Trade CapEX for variable expense | Our economies of scale provide us with lower costs | Pricing model choice to support variable and stable workloads | Save more money as you grow bigger |
| --- | --- | --- | --- |
| Pay for what you use | 77 price reductions since 2006 | On-demand<br>Reserved Instances<br>Spot | Tiered pricing<br>Volume discounts<br>Custom pricing |

# Customer obsessed

90%

of roadmap originates with customer requests and are designed to meet specific needs

heroku

"Performance, reliability, and responsiveness are fundamental to our customer experience, and T3 instances help us to deliver on that customer promise while also controlling our costs."

—Heroku



aws

**Figure 1. Magic Quadrant for Cloud Infrastructure and Platform Services**


# AWS Recognized as a Cloud Leader for the 10th Consecutive Year

*Gartner, Magic Quadrant for Cloud Infrastructure as a Service, Worldwide, Raj Bala, Bob Gill, Dennis Smith, David Wright, July 2019. ID G00365830. Gartner does not endorse any, product or service depicted in its research publications, and does not advise technology users to select only those vendors with the highest ratings. Gartner research publications consist of the opinions of Gartner's research organization and should not be construed as statements of fact. Gartner disclaims all warranties, expressed or implied, with respect to this research, including any warranties of merchantability or fitness for a particular purpose. The Gartner logo is a trademark and service mark of Gartner, Inc., and/or its affiliates, and is used herein with permission. All rights reserved.*


# AWS Global Reach

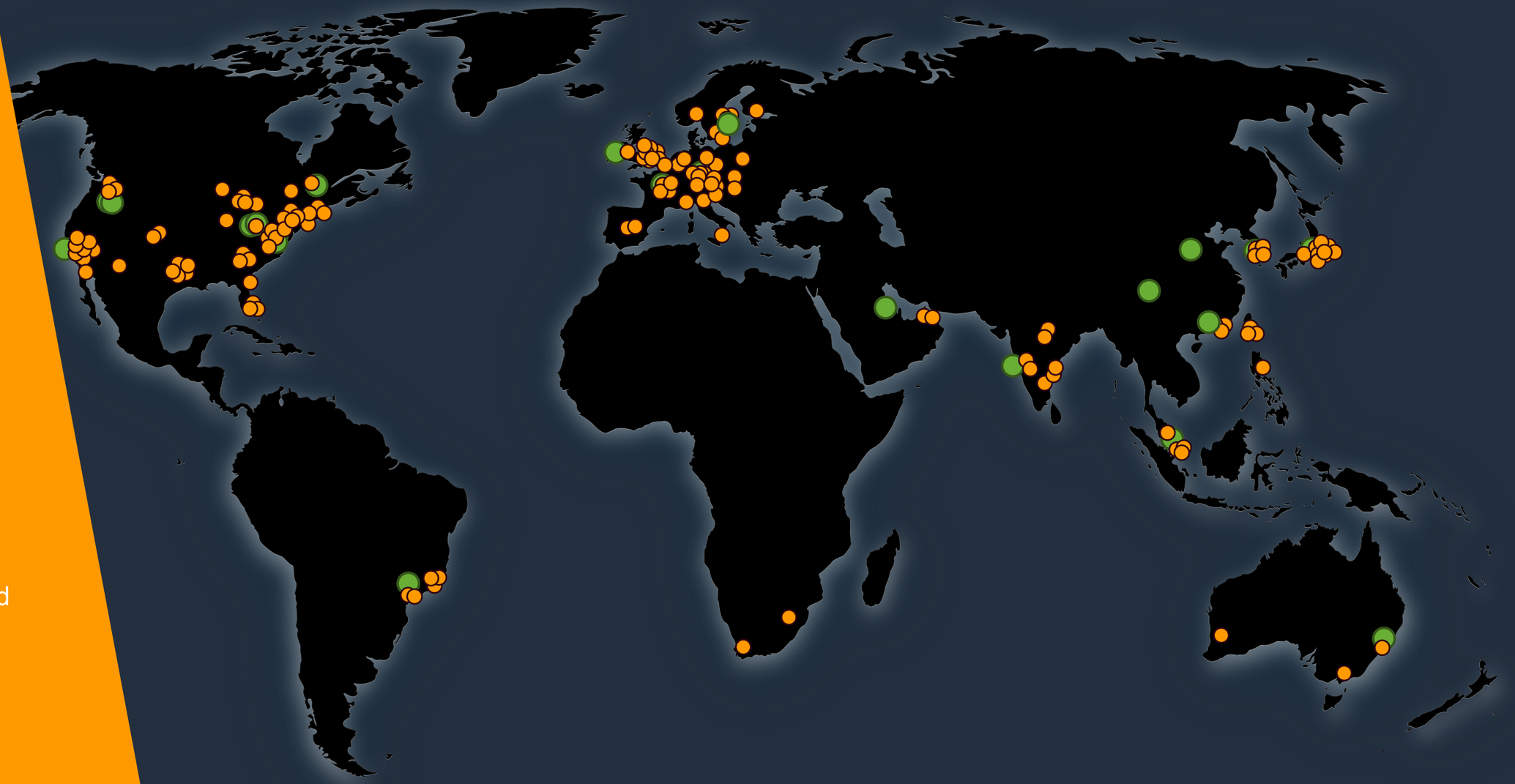
# 24 Regions
# 77 Availability Zones

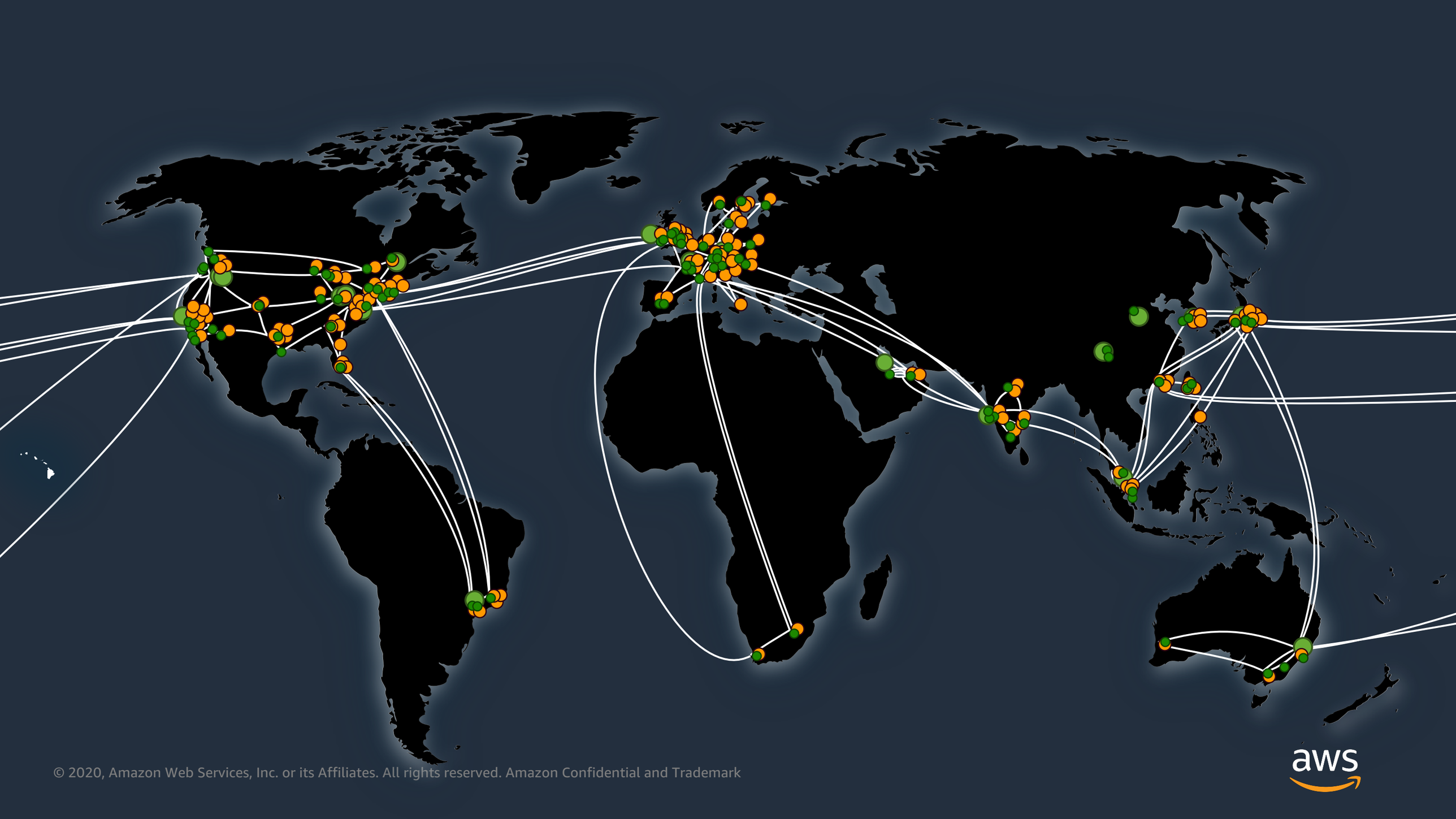
AWS Local Zones
AWS Wavelength
AWS Outposts

# 216

## Amazon CloudFront Points of Presence

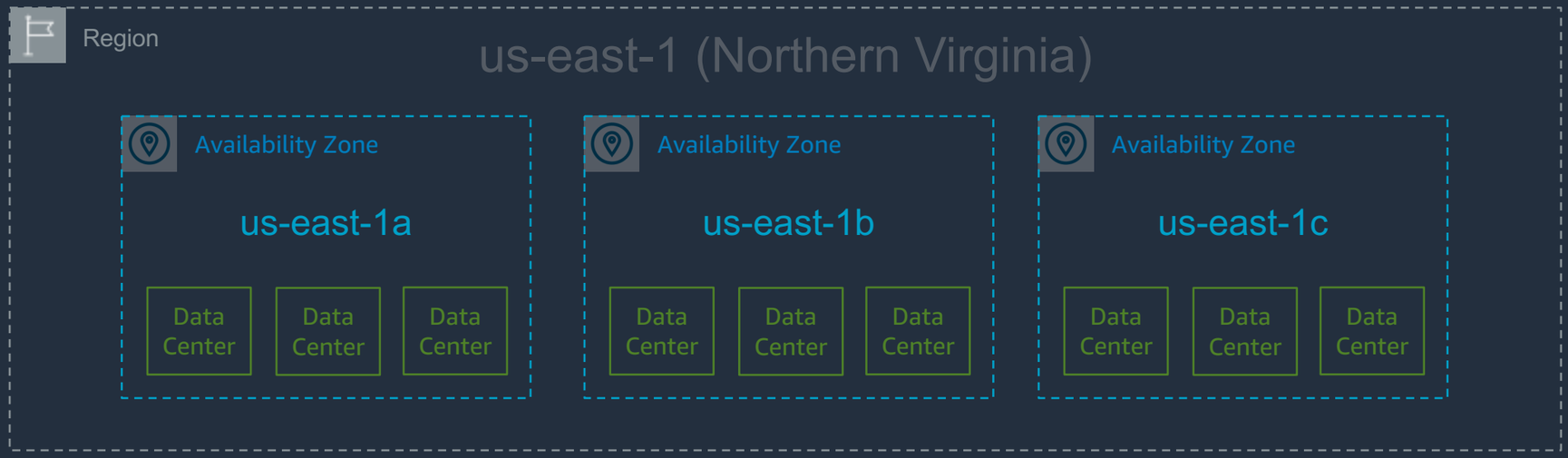
205 Edge Locations and
11 Regional Edge Caches



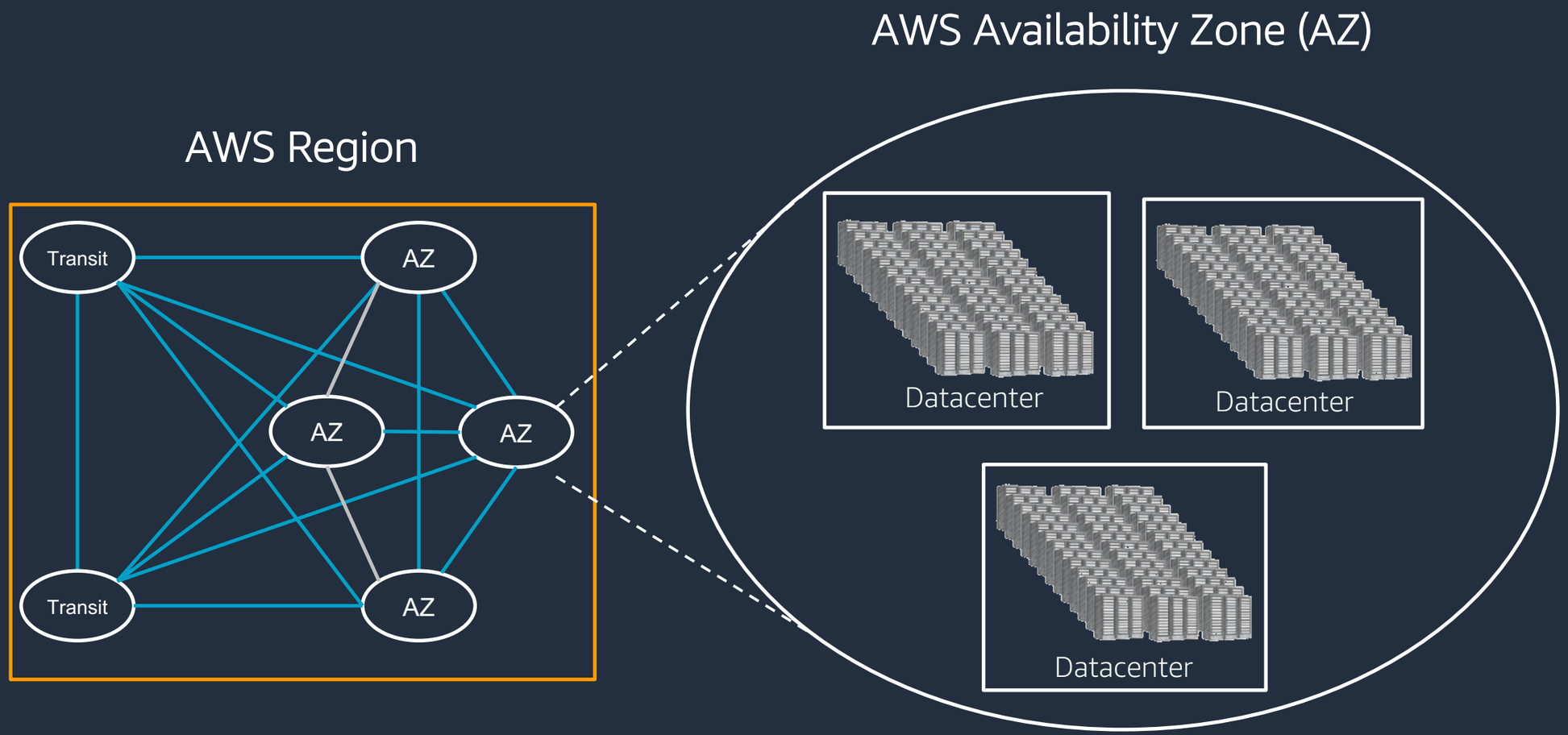


# Availability Zones

- A region is comprised of multiple Availability Zones (typically 3)
- Fully independent partitions on isolated fault lines, flood plains, and power grids
- Each AZ: redundant power and redundant dedicated network
- Each AZ: typically multiple data centers
- Between AZs: high throughput, low latency (<10ms) network
- Between AZs: physical separation < 100km (60mi)

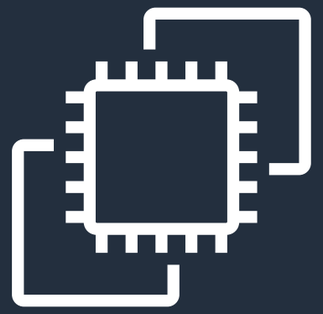


# Availability Zones




# EC2 Overview

# Choices for Compute



## Amazon EC2

Virtual server instances
in the cloud

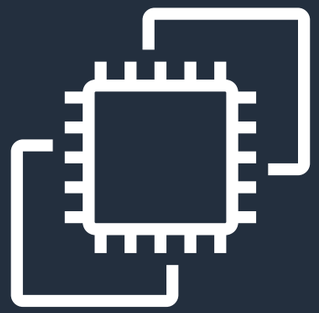
## Amazon ECS, EKS, and Fargate

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 EC2



Amazon EC2

Linux | Windows

Arm and x86 architectures

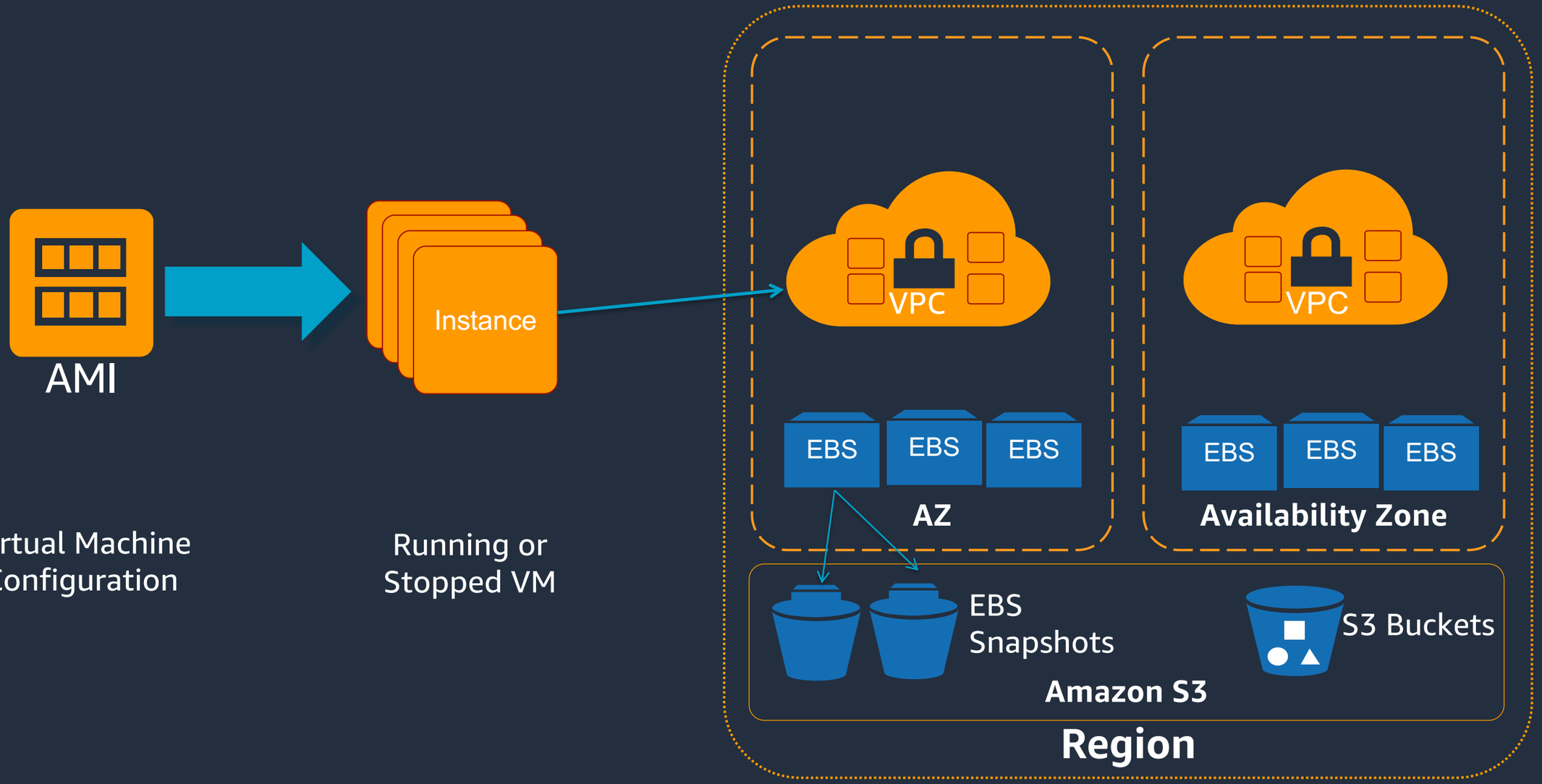
General purpose and workload optimized

Bare metal, disk, networking capabilities

Packaged | Custom | Community AMIs

Multiple purchase options: On-demand, RI, Spot

# EC2 Terminology




Virtual Machine Configuration

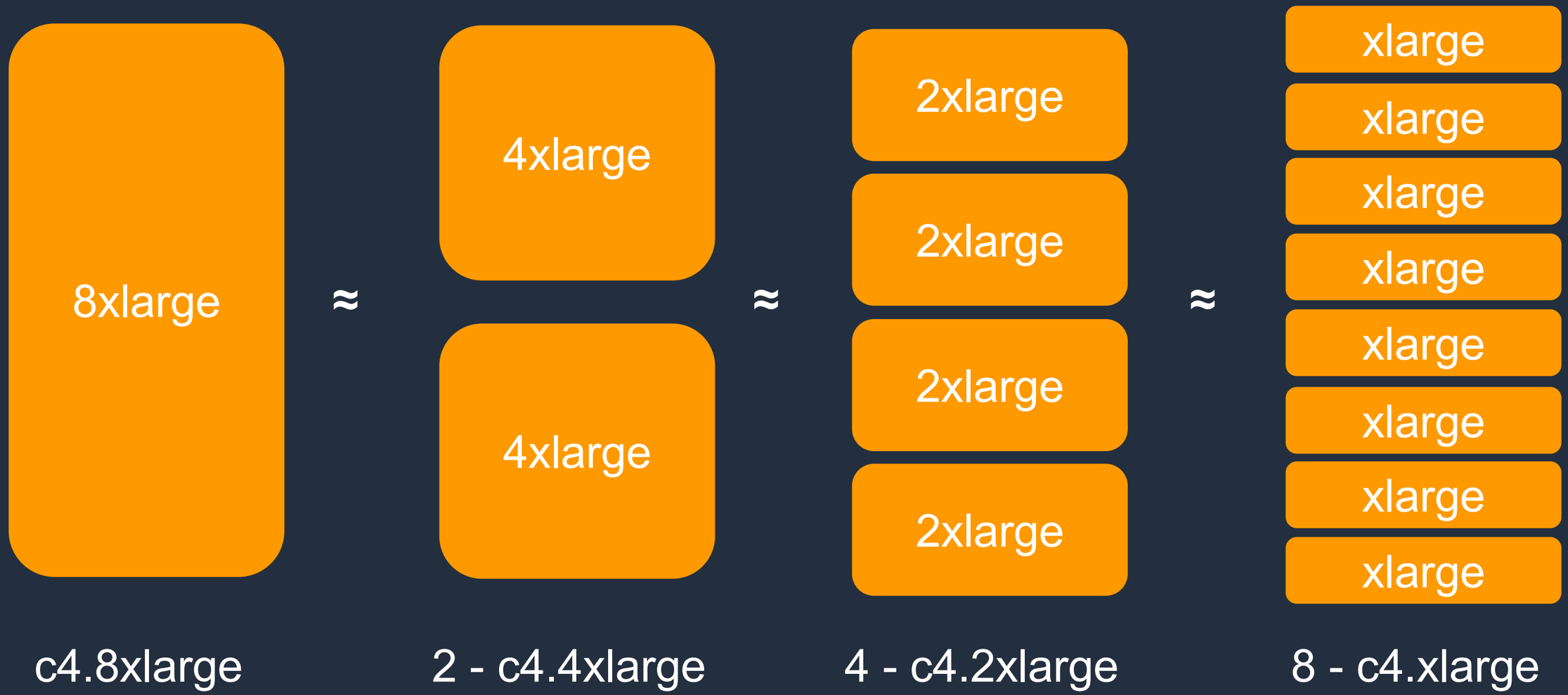
Running or Stopped VM

# What’s a virtual CPU? (vCPU)

- A vCPU is typically a hyper-threaded physical core*
- Divide vCPU count by 2 to get core count
- Cores by Amazon EC2 & RDS DB Instance type: https://aws.amazon.com/ec2/virtualcores/

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

# Instance sizing



# 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. https://aws.amazon.com/ec2/dedicated-hosts/pricing/
  - https://aws.amazon.com/ec2/instance-types/

*Again, the “T” family is special

# Choose your processor and architecture

Intel® Xeon® Scalable (Skylake) processor

NVIDIA V100 Tensor Core GPUs

AMD EPYC processor

Amazon ARM based Cloud Processor

FPGAs for custom hardware acceleration

**Right compute for the right application and workload**

# EC2 Naming Explained

Instance generation

c5n.xlarge

Instance family

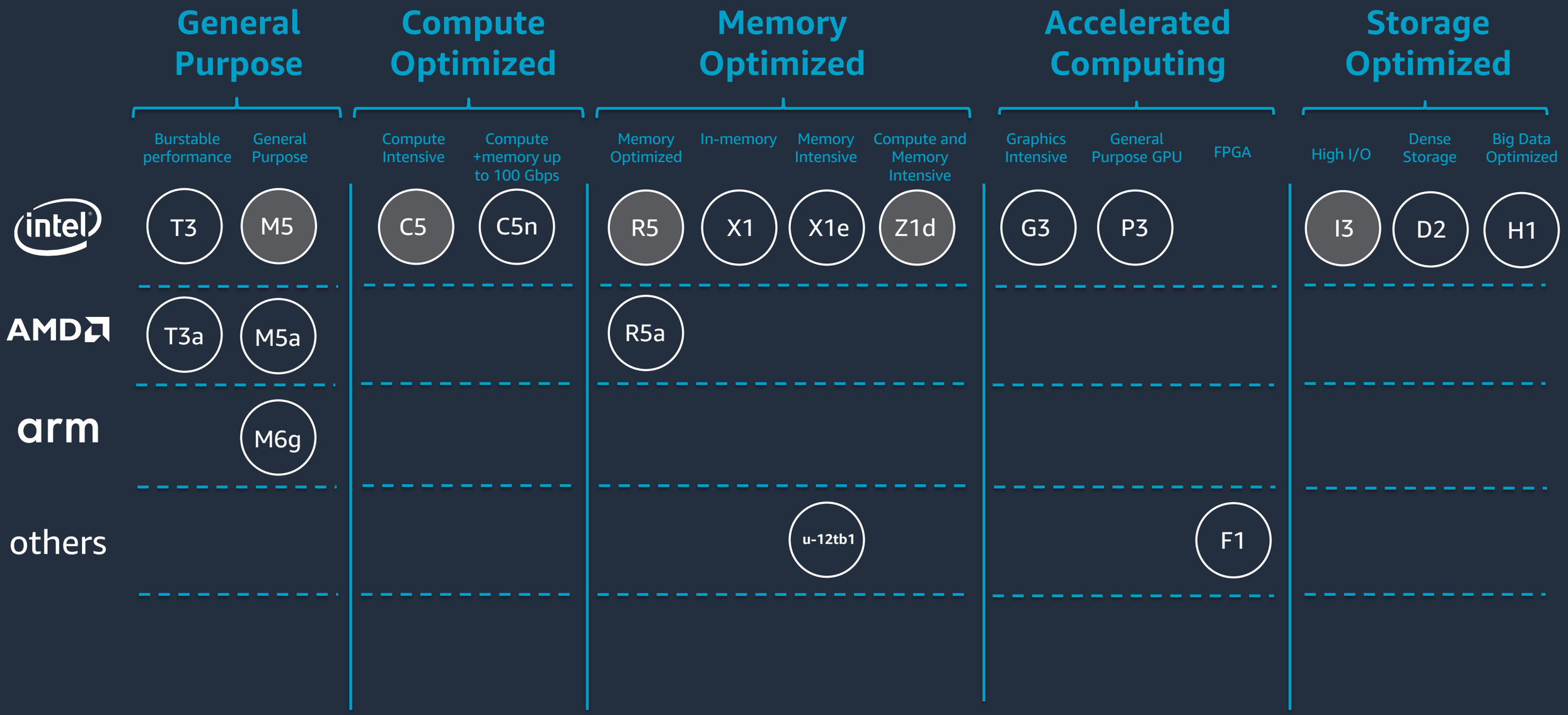
Attribute

Instance size



aws

# Instance Types





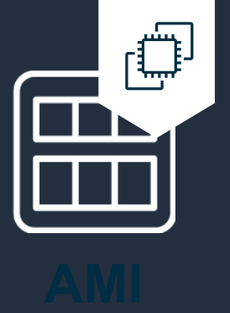

aws

# EC2 Operating Systems Supported

- Windows
- Amazon Linux
- Debian
- Suse
- CentOS
- Red Hat Enterprise Linux
- Ubuntu

for more OSes see: https://aws.amazon.com/marketplace/b/2649367011

# What is an Amazon Machine Image (AMI)?




Provides the information required to launch an instance

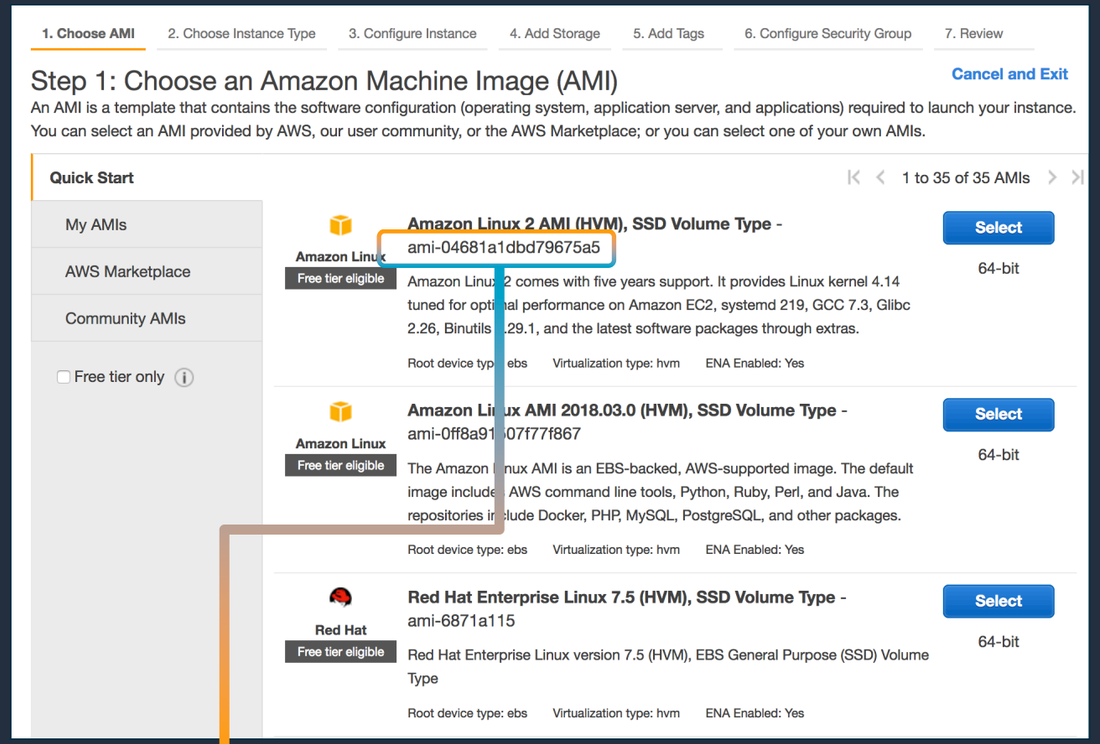
Launch multiple instances from a single AMI

An AMI includes the following

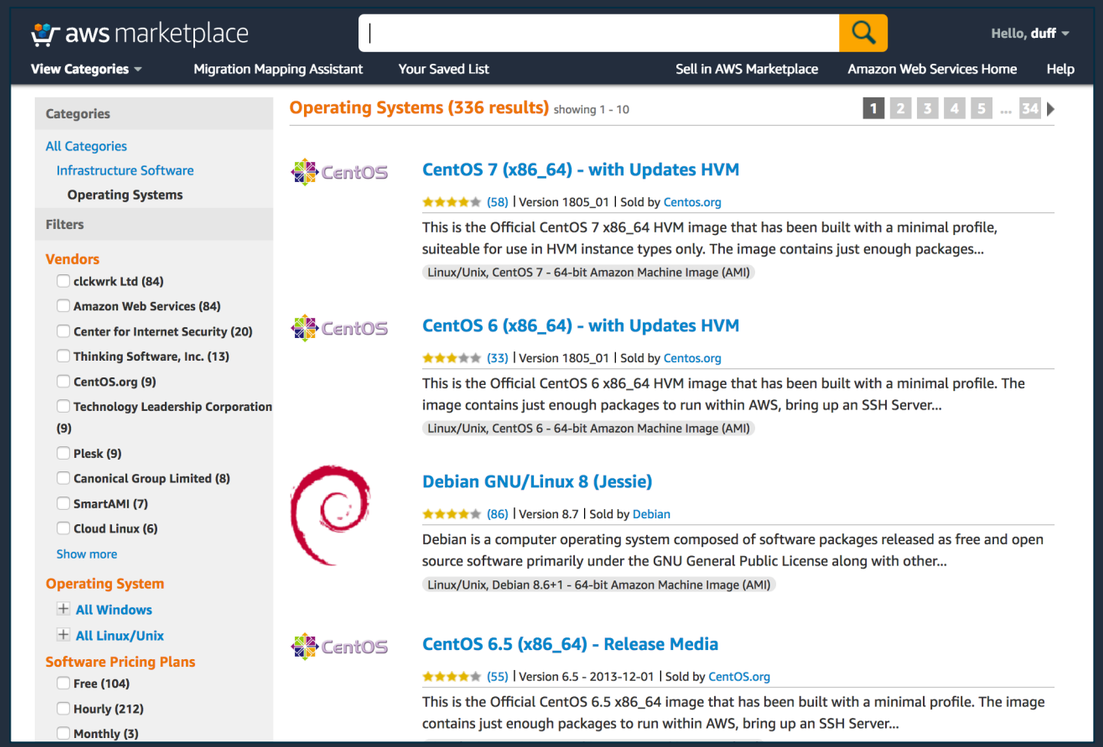
- A template for the root volume (for example, operating system, applications)
- Launch permissions that control which AWS accounts can use the AMI
- Block device mapping that specifies volumes to attach to the instance

# Choosing an AMI

## AWS Console



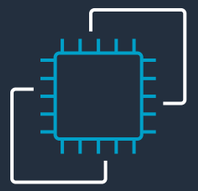

## AWS Marketplace




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
```

# 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



aws

# Purchasing Options

## On-Demand

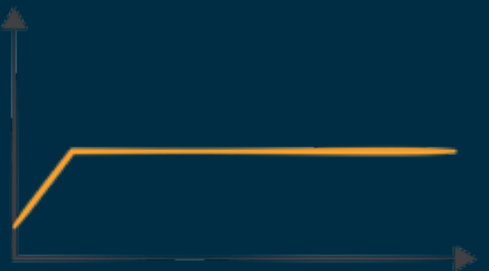
Pay for compute capacity by **the second** with no long-term commitments

For Spiky workloads or to define needs

## Reserved Instances

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

For committed utilization



## Spot Instances

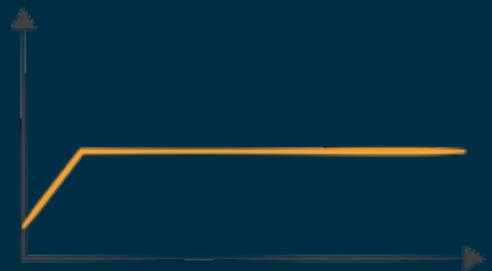
Spare EC2 capacity at **savings of up to 90%** off On-Demand prices

For time-insensitive or transient workloads Need to be Fault-tolerant, stateless

## Savings Plans

Commit to a $/h spend and **share discount** across compute options and regions

For committed utilization

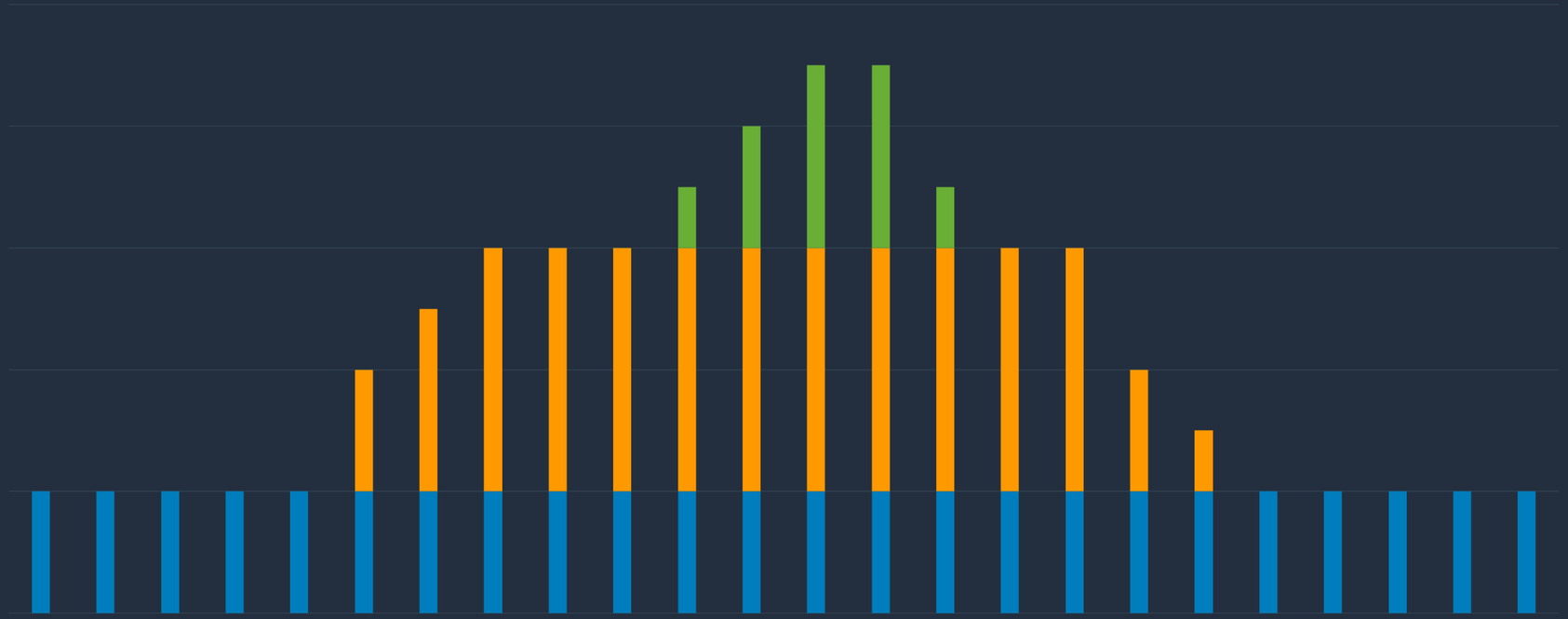


To optimize EC2, combine all three purchase options!



aws

# Simplify capacity and cost optimization



Now with hibernate

Scale using
**Spot,**
**On-Demand,**
or both

Use **Reserved Instances**
for known/steady-state
workloads

## AWS services make this easy and efficient

- Amazon EC2 Auto Scaling
- EC2 Fleet
- Amazon Elastic Container Service
- Amazon Elastic Container Service for Kubernetes
- 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



aws

# EC2 Options

## Categories

General purpose

Burstable

Compute intensive

Memory intensive

Storage (High I/O)

Dense storage

GPU compute

Graphics intensive

+

## Capabilities

NEW! Choice of processor
(AWS, Intel, AMD)

Fast processors
(up to 4.0 GHz)

High memory footprint
(up to 12 TiB)

Instance storage
(HDD and NVMe)

Accelerated computing
(GPUs and FPGA)

NEW! Networking
(up to 100 Gbps)

Bare Metal

Size
(Nano to 32xlarge)

+

## Options

Elastic Block Store

Elastic Graphics

NEW! Elastic Inference

=

**200+**

**instance types**

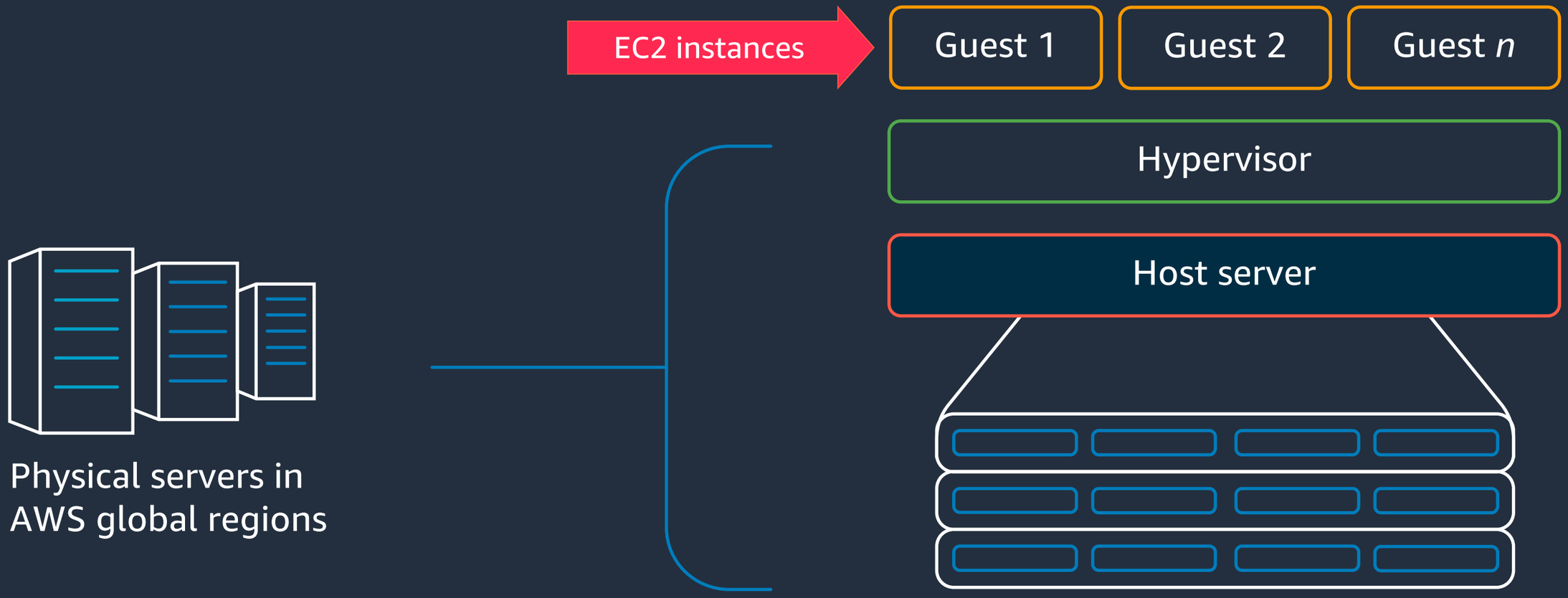
for virtually every workload and business need



aws


# EC2 Design

# EC2 Host Virtualization



# 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**

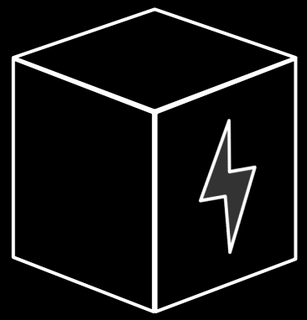
- AWS Nitro System (launched on Nov 2017)
- Less server resources used, more resources for the customer
- AWS optimized

Bare metal: **Direct access to processor and memory resources**

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

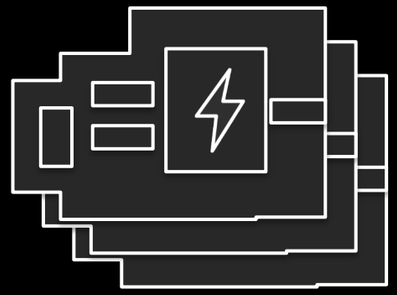
# AWS Nitro System

## *Nitro Hypervisor*



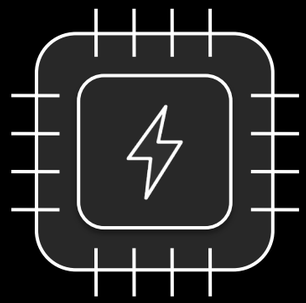
- Lightweight hypervisor
- Memory and CPU allocation
- Bare Metal-like performance

## *Nitro Card*



- VPC Networking
- Amazon EBS
- Local Instance
- System Controller

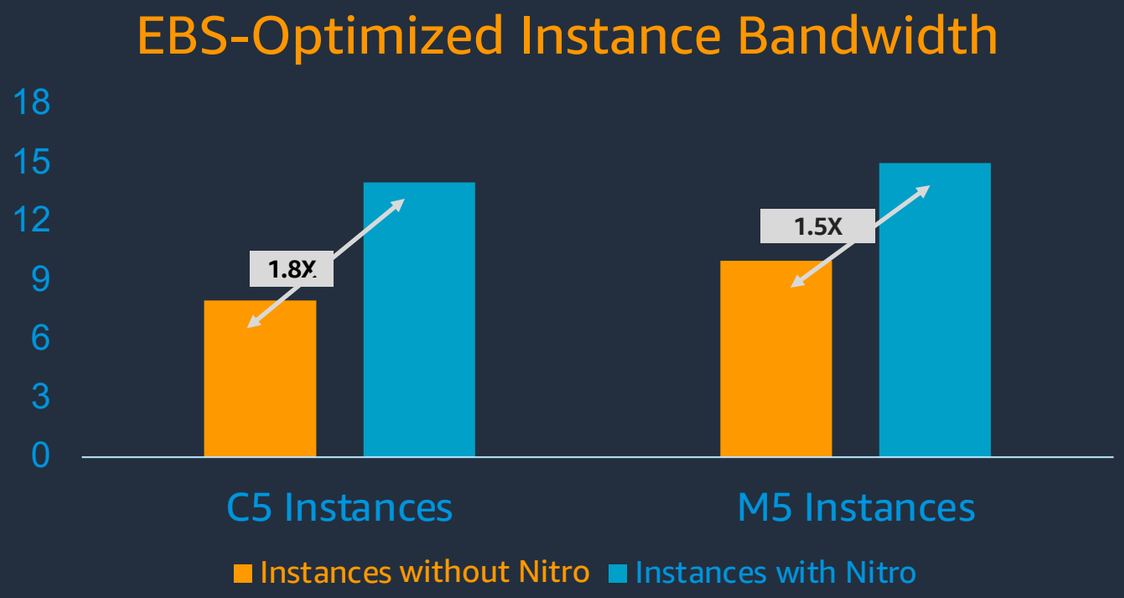
## *Nitro Security Chip*

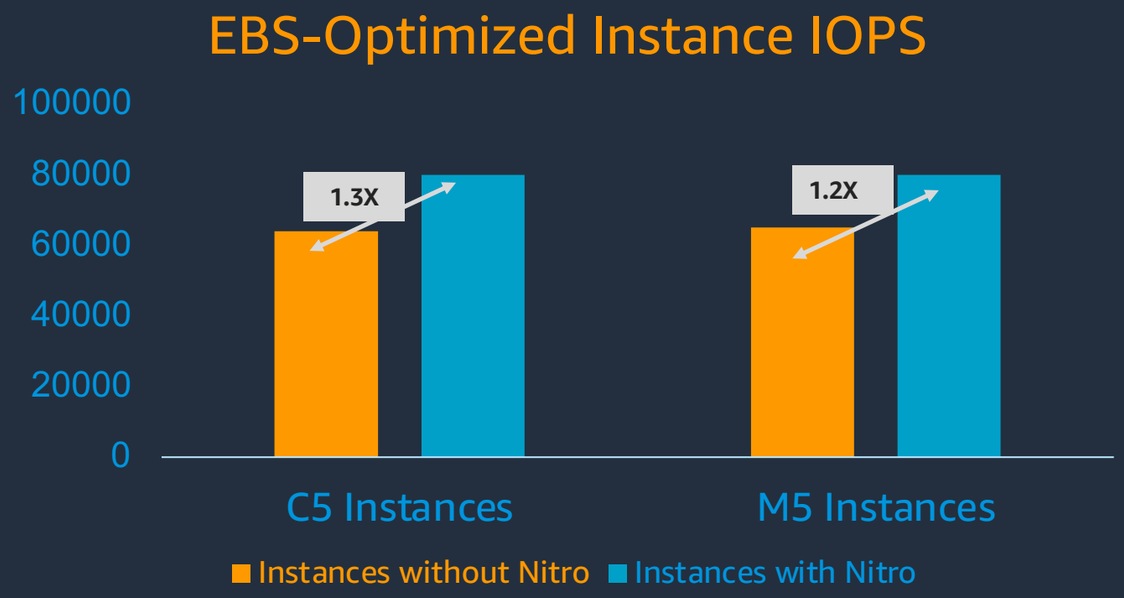


- Integrated into motherboard
- Protects hardware resources
- Hardware Root of Trust

**Modular Building Blocks for rapid design and delivery of EC2 instances**

# AWS Nitro System



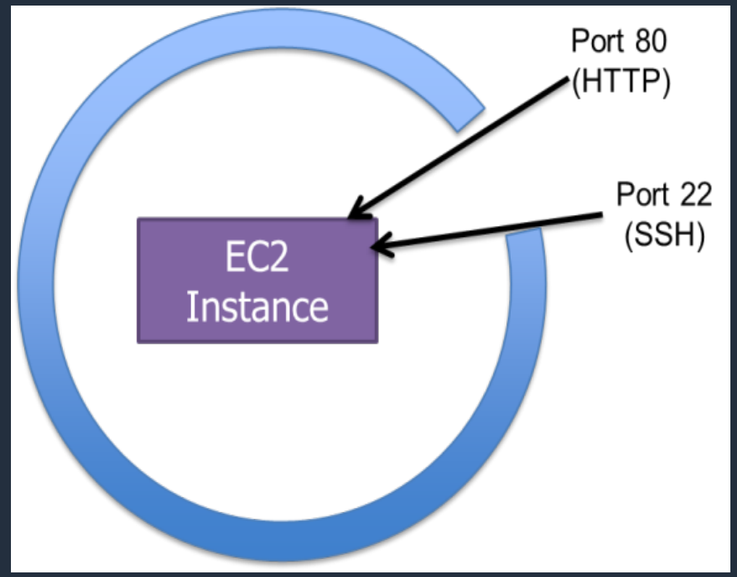




Nitro instances provide **bandwidth, performance, and price improvements** over previous instance generations

# EC2 Security Groups

## Security Group Rules

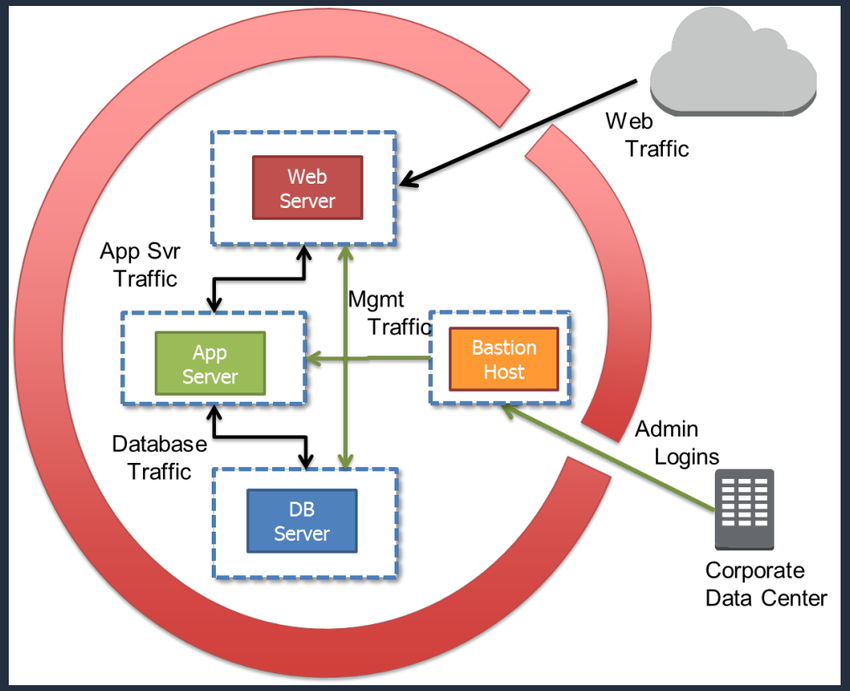
- Name
- Description
- Protocol
- Port range
- IP address, IP range, Security Group name



# Tiered EC2 Security Groups

## Hierarchical Security Group Rules

- Dynamically created rules
- Based on Security Group membership
- Create tiered network architectures

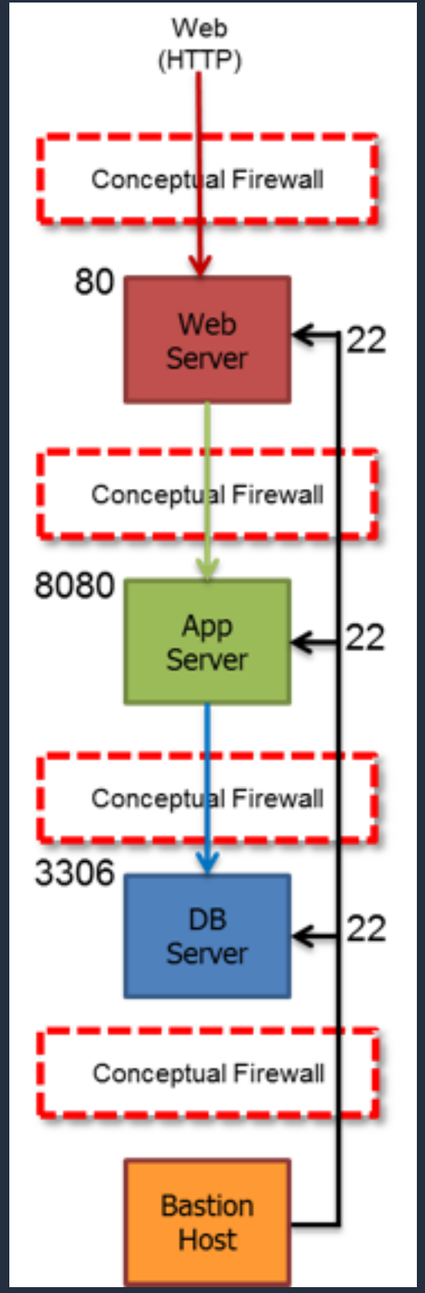



"Web" Security Group:
TCP 80 0.0.0.0/0
TCP 22 "Mgmt"

"App" Security Group:
TCP 8080 "Web"
TCP 22 "Mgmt"

"DB" Security Group:
TCP 3306 "App"
TCP 22 "Mgmt"

"Mgmt" Security Group:
TCP 22 163.128.25.32/32



# 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)

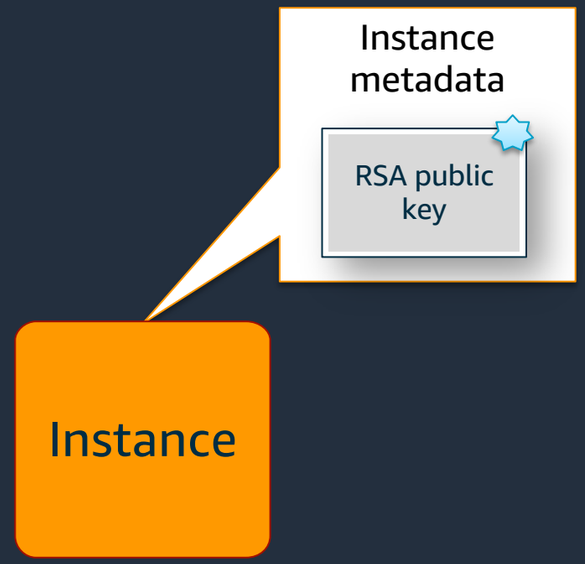
"Public Half" inserted by Amazon into each EC2 instance that you launch

"Private Half" downloaded to your desktop

# EC2 Instance access and Key Pairs

Linux launch (first boot)

- **Public key** made available through metadata
- Public key inserted into ~/.ssh/authorized_keys
- User connects with SSH using their **private key**



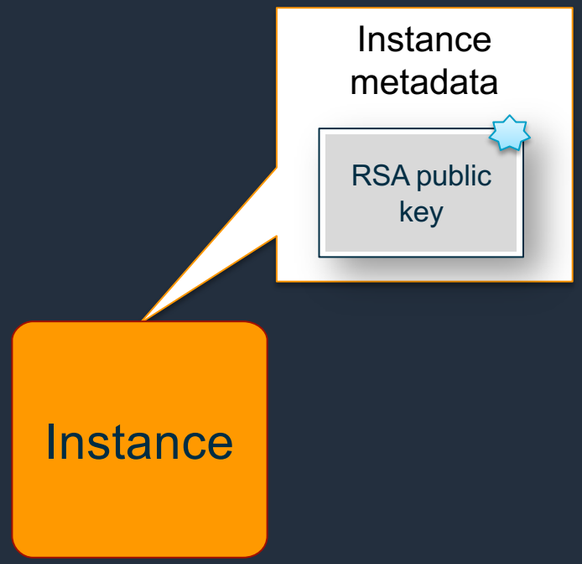
# EC2 Instance access and Key Pairs

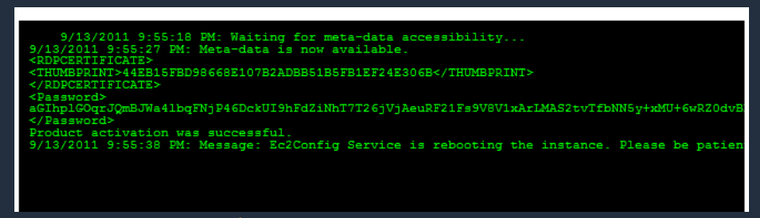
## Linux launch (first boot)

- **Public key** made available through metadata
- Public key inserted into ~/.ssh/authorized_keys
- User connects with SSH using their **private key**

## Windows launch (first boot sequence)

- **Public key** made available through metadata
- Sysprep
- Random Administrator password
- Password encrypted with public key
- User decrypts password with their **private key**




**System log**
<Password>
aGIhplGOqrJQmBJW
…
K9gTD31Q==
</Password>

# Instance Metadata

http://169.254.169.254/latest/meta-data/ contains a wealth of info

- ami-id
- ami-launch-index
- ami-manifest-path
- block-device-mapping/
- hostname
- instance-action
- **instance-id**
- instance-type
- kernel-id
- local-hostname
- local-ipv4
- mac
- network/
- **placement/availability-zone**
- profile
- public-hostname
- public-ipv4
- public-keys/



aws

# Any Questions?