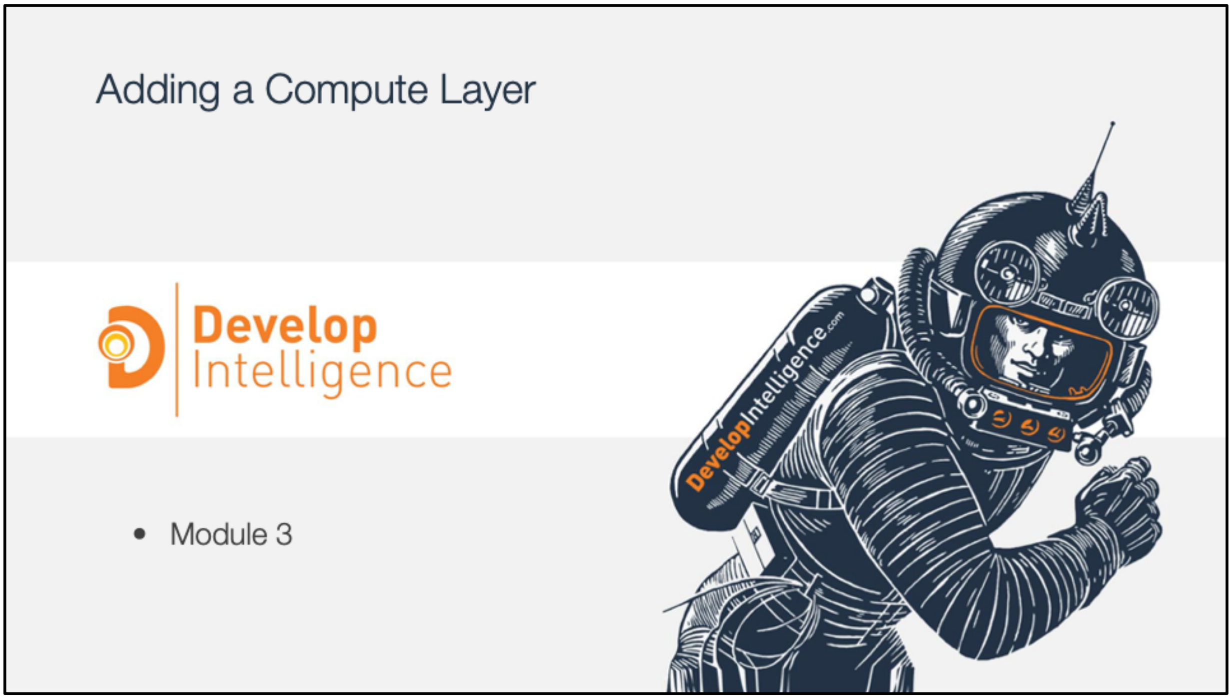
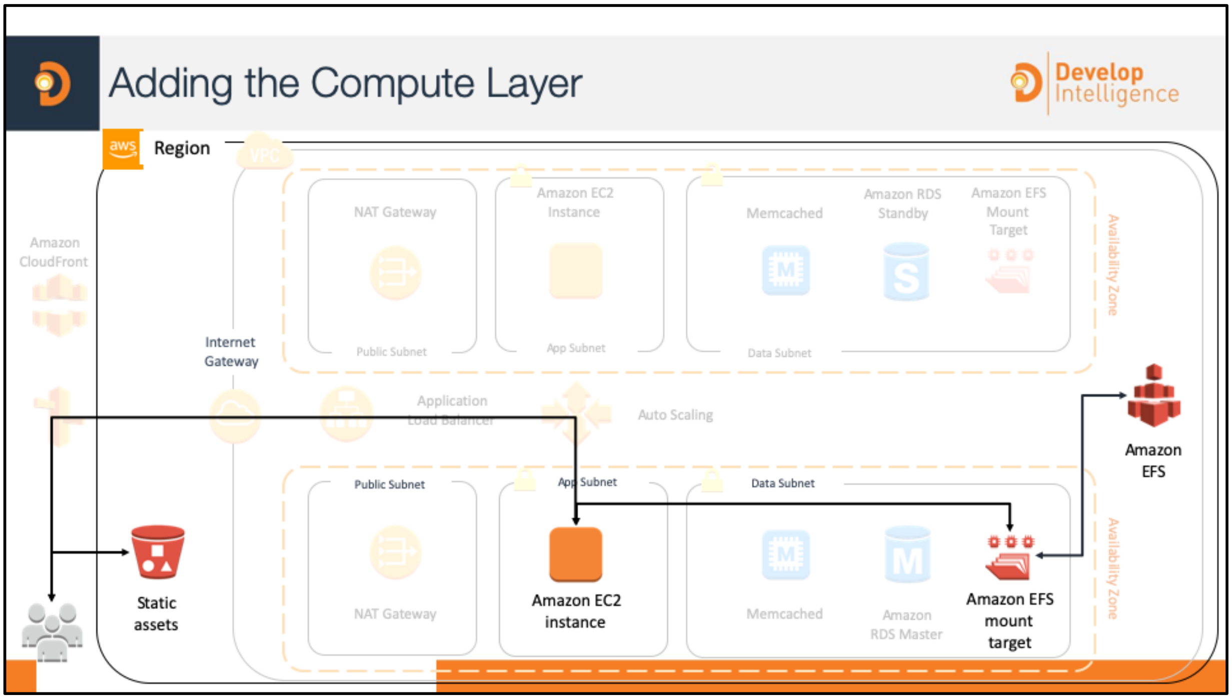


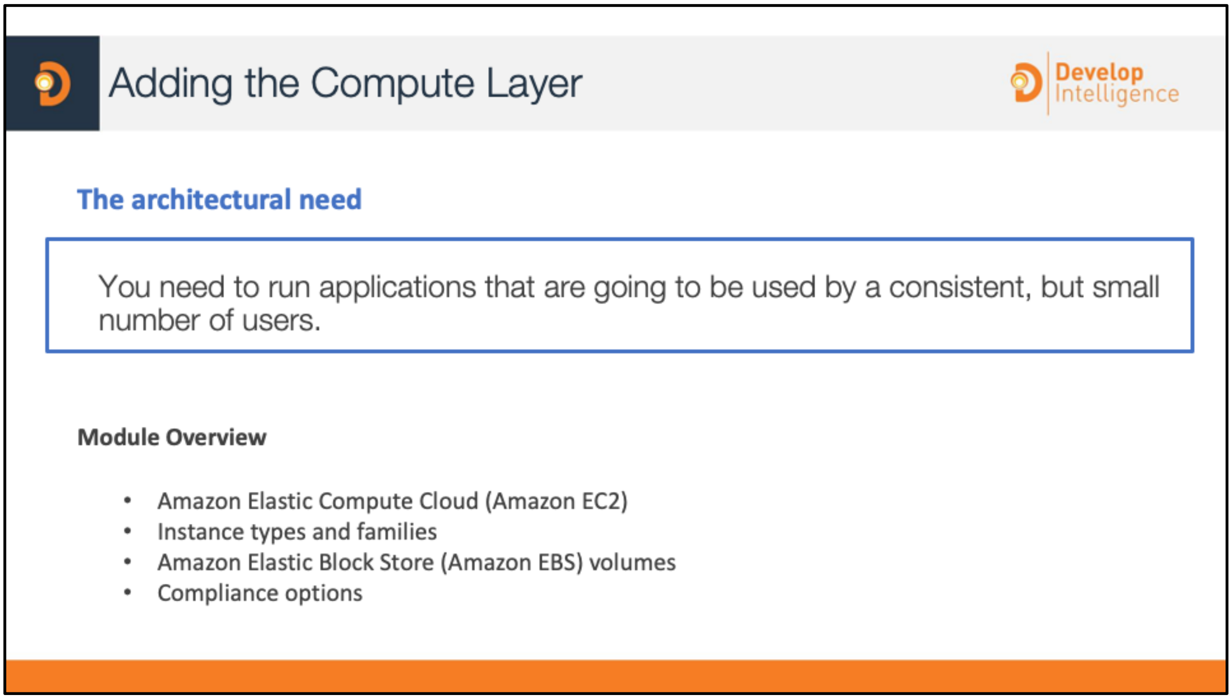
Adding a Compute Layer

- Module 3



Adding the Compute Layer

By the end of class, you will be able to understand all of the components of this architectural diagram. You will also be able to construct your own architectural solutions that are just as large and robust.



Adding the Compute Layer

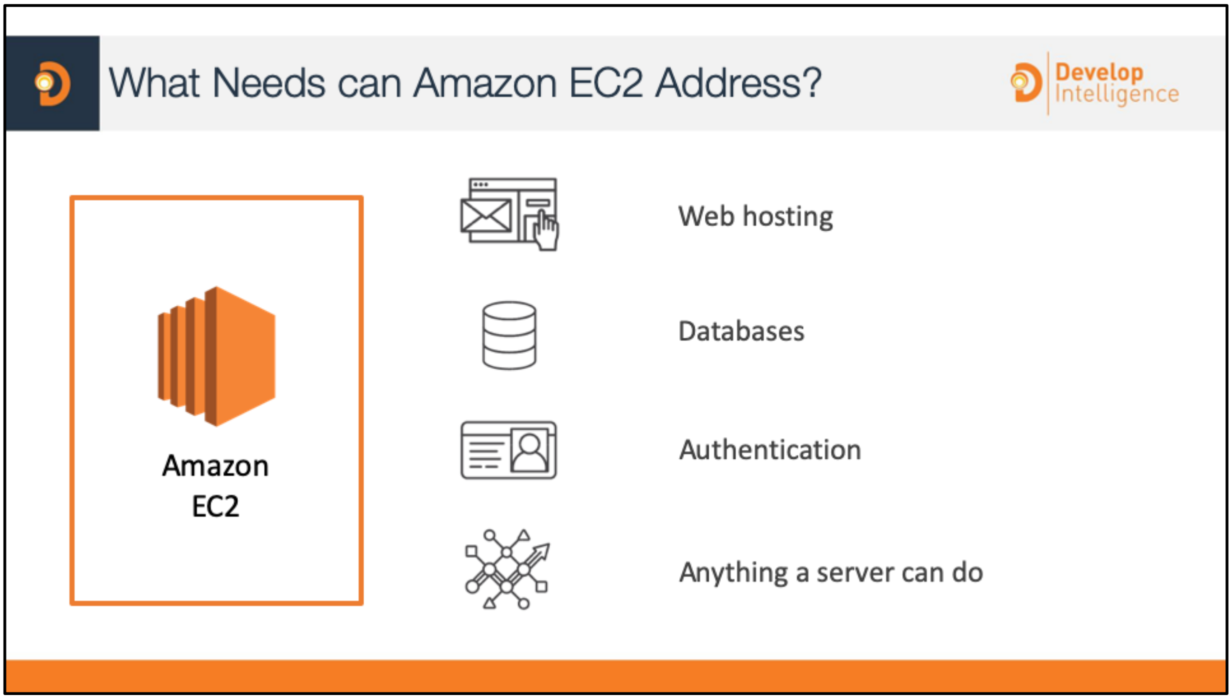
Develop Intelligence

The architectural need

You need to run applications that are going to be used by a consistent, but small number of users.

Module Overview

- Amazon Elastic Compute Cloud (Amazon EC2)
- Instance types and families
- Amazon Elastic Block Store (Amazon EBS) volumes
- Compliance options

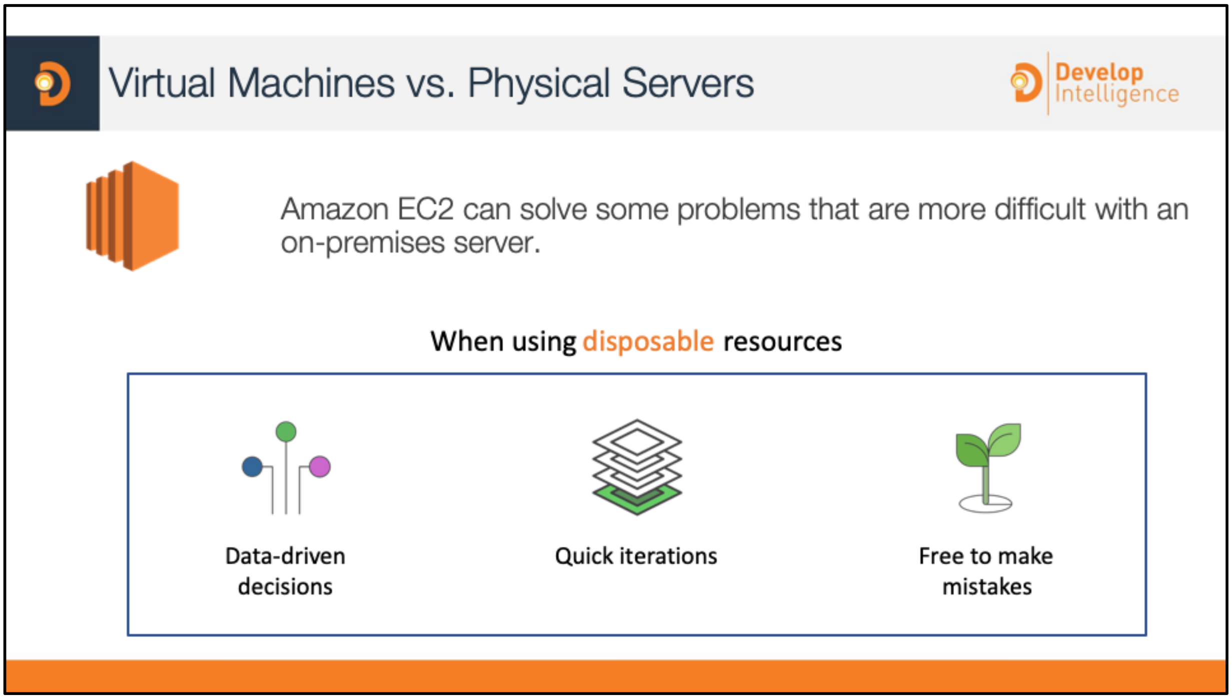


What Needs can Amazon EC2 Address?

Amazon EC2

- Web hosting
- Databases
- Authentication
- Anything a server can do

Amazon EC2 is just like your traditional on-premises server, but it is available in the cloud. It can support workloads such as web hosting, applications, databases, authentication services, and anything else a server can do.



Virtual Machines vs. Physical Servers

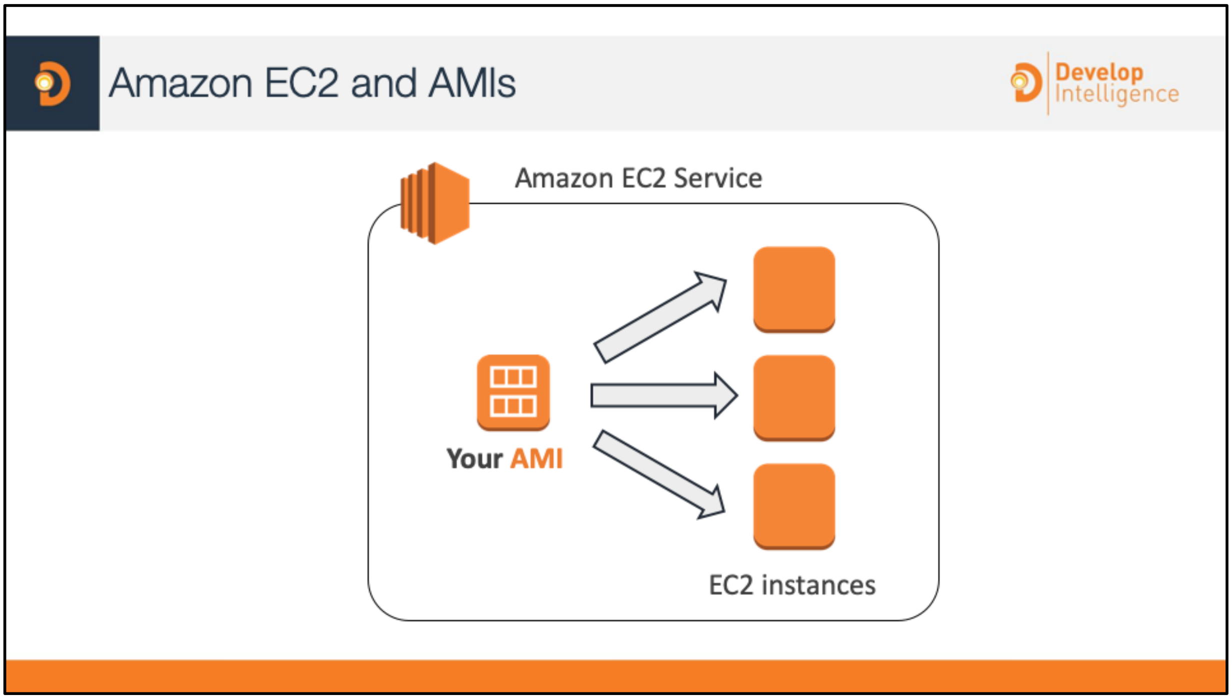
Develop Intelligence

Amazon EC2 can solve some problems that are more difficult with an on-premises server.

When using disposable resources

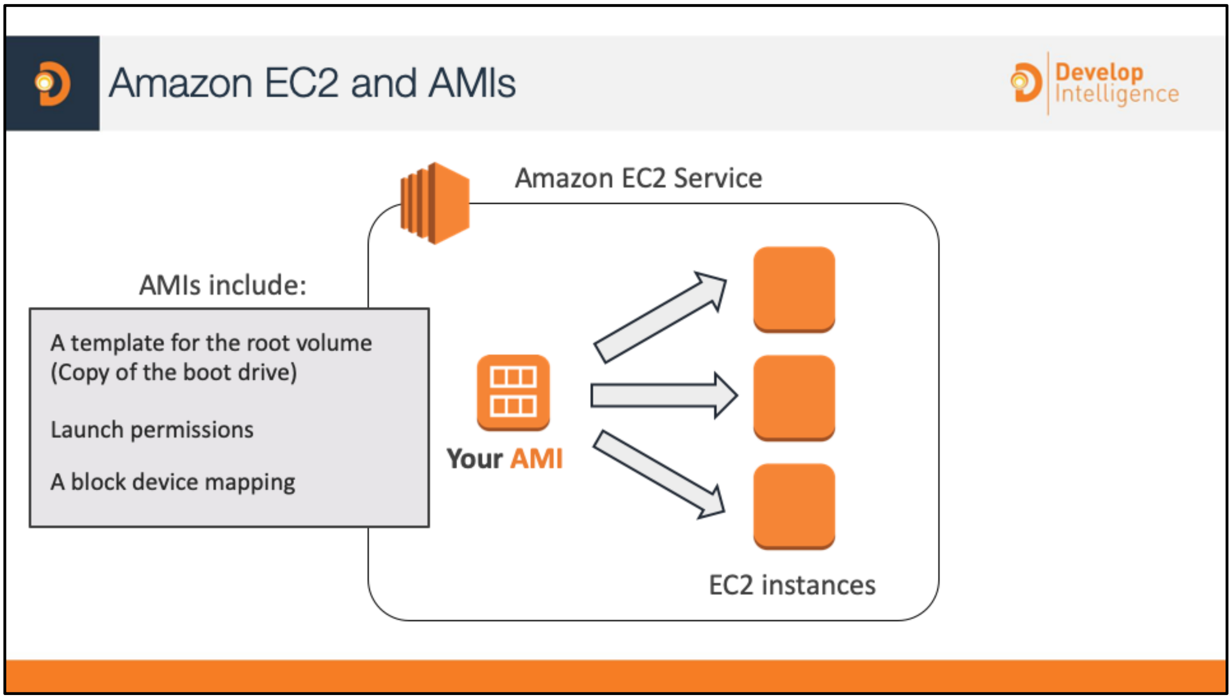
- Data-driven decisions
- Quick iterations
- Free to make mistakes

On AWS, servers, databases, storage, and higher-level application components can be instantiated within seconds. You can treat these as temporary and disposable resources, free from the inflexibility and constraints of a fixed and finite IT infrastructure. This resets the way you approach change management, testing, reliability, and capacity planning.



Amazon EC2 and AMIs

An *Amazon Machine Image (AMI)* provides the information required to launch an *instance*, which is a virtual server in the cloud. You must specify a source AMI when you launch an instance. You can launch multiple instances from a single AMI when you need multiple instances with the same configuration. For example, you can use a single AMI to launch a cluster of instances (identical except for their IP address) to be placed beneath a load balancer. You can also use different AMIs to launch different types of instance. For example, I might have one AMI to implement web server instances in my architecture, and another to implement application server instances.



Amazon EC2 and AMIs

Amazon EC2 Service

AMIs include:

A template for the root volume (Copy of the boot drive)

Launch permissions

A block device mapping

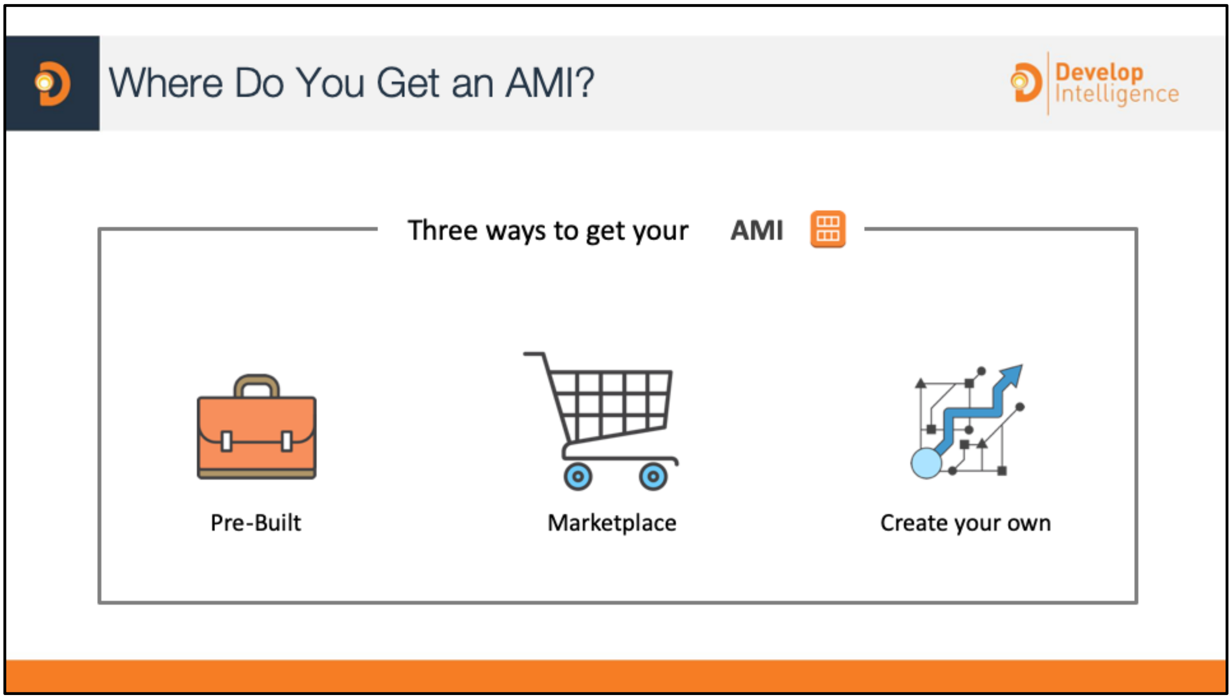
Your AMI

EC2 instances

An AMI includes the following:

- A template for the root volume of an EC2 instance. A root volume typically contains a full operating system (OS) and everything that has been installed into that OS (the applications, libraries, utilities, and so on). The EC2 service copies the template to the root volume of a new EC2 instance and then starts it up.
- Launch permissions that control which AWS accounts can use the AMI to launch instances
- A block device mapping that specifies the volumes to attach to the instance (if any) when it's launched

For more information about AMIs, see https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/AMIs.html



Where Do You Get an AMI?

Develop Intelligence

Three ways to get your AMI

Pre-Built

Marketplace

Create your own

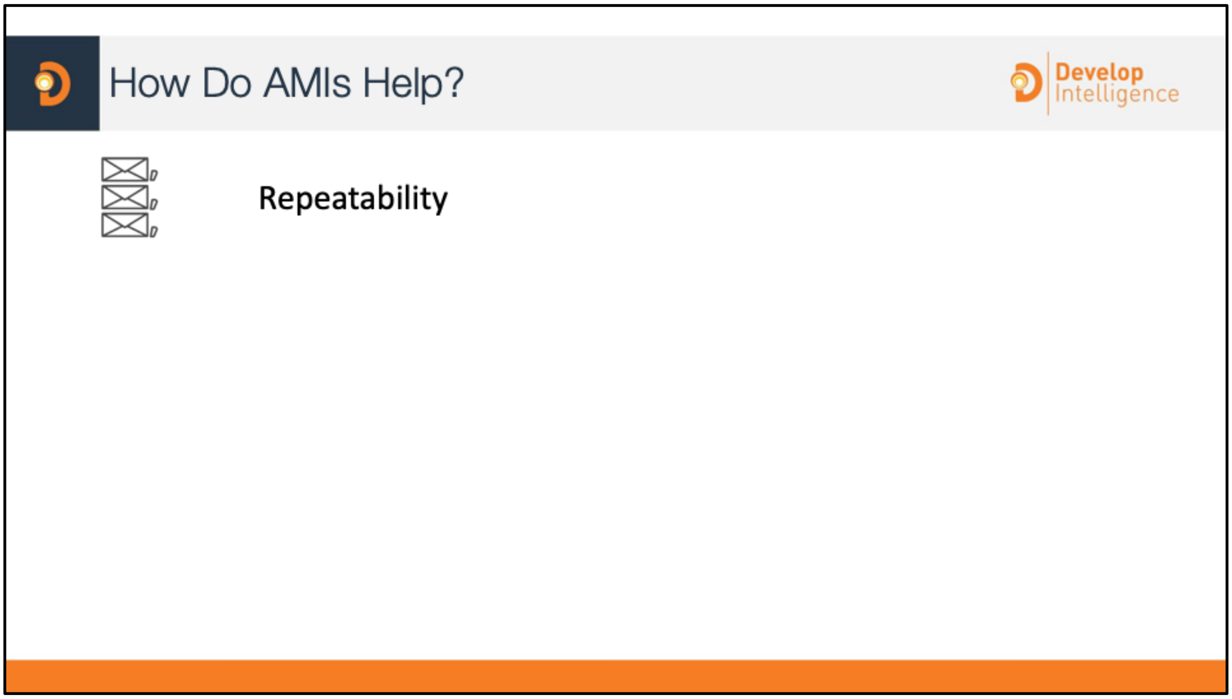
Pre-Built: Amazon offers a number of prebuilt AMIs to launch your instances. These AMIs include Linux and Windows options, with various sub-options to tailor your setup.

Marketplace: The AWS Marketplace offers a digital catalog with thousands of software solutions listed. These AMIs can offer specific use cases to help you get started quickly

Create your own: An AMI is simply an anonymized, block-level copy of the root volume of a "donor machine" or "golden instance" – a virtual machine (VM) that you've configured with the specific OS and application content you want placed on the AMI. When you create an AMI, Amazon EC2 stops the instance, snapshots its root volume, and finally registers the snapshot as an AMI.

There are also **community AMIs** created by people all over the globe. These AMIs are not vetted by AWS and are used at your own risk. These AMIs can offer many different solutions to various problems, but please use them with great care. Do not use them in any production/corporate environment.

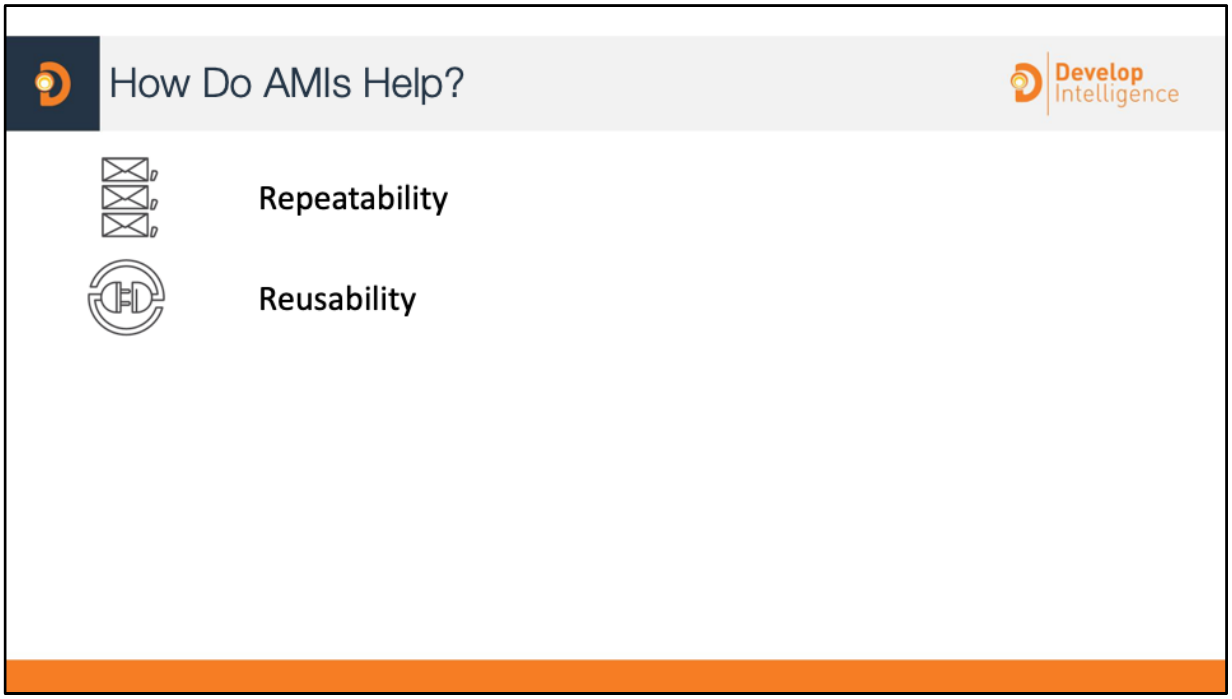
For more information about AMIs, see https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/AMIs.html



How Do AMIs Help?

Repeatability

Using AMIs solves a whole host of problems. First, repeatability: instances launched from the same AMI are exact replicas of one another. That makes building clusters of similar instances or recreating compute environments considerably easier.



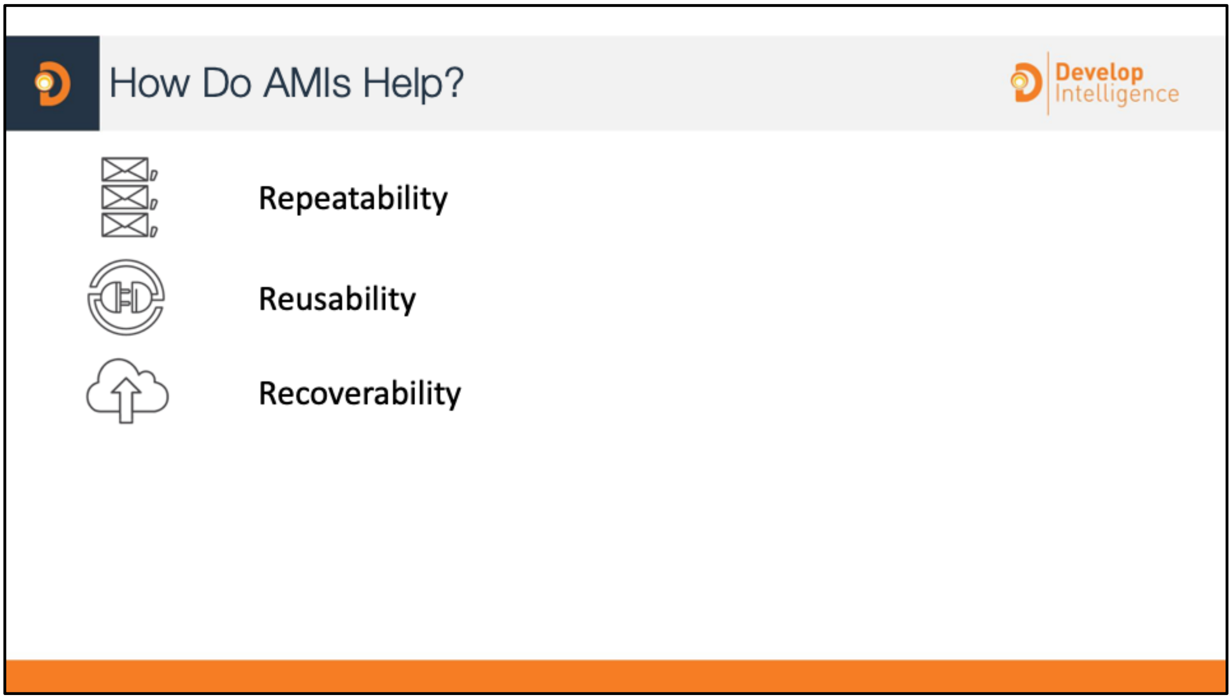
How Do AMIs Help?

Develop Intelligence

Repeatability

Reusability

Reusability: AMIs package the full configuration and content of an EC2 instance such that it can be used over and over again, with efficiency and precision.



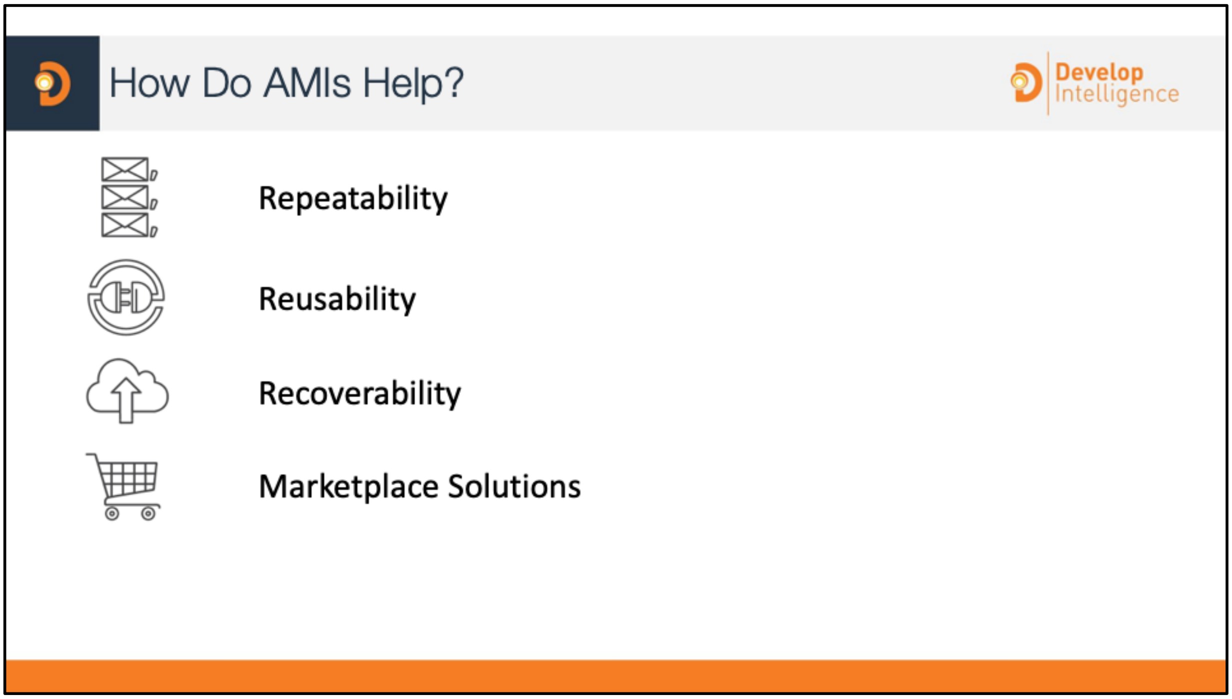
How Do AMIs Help?

Repeatability

Reusability

Recoverability

Recoverability: An AMI is perfect for replacing failed machines with new instances created from the same AMI.

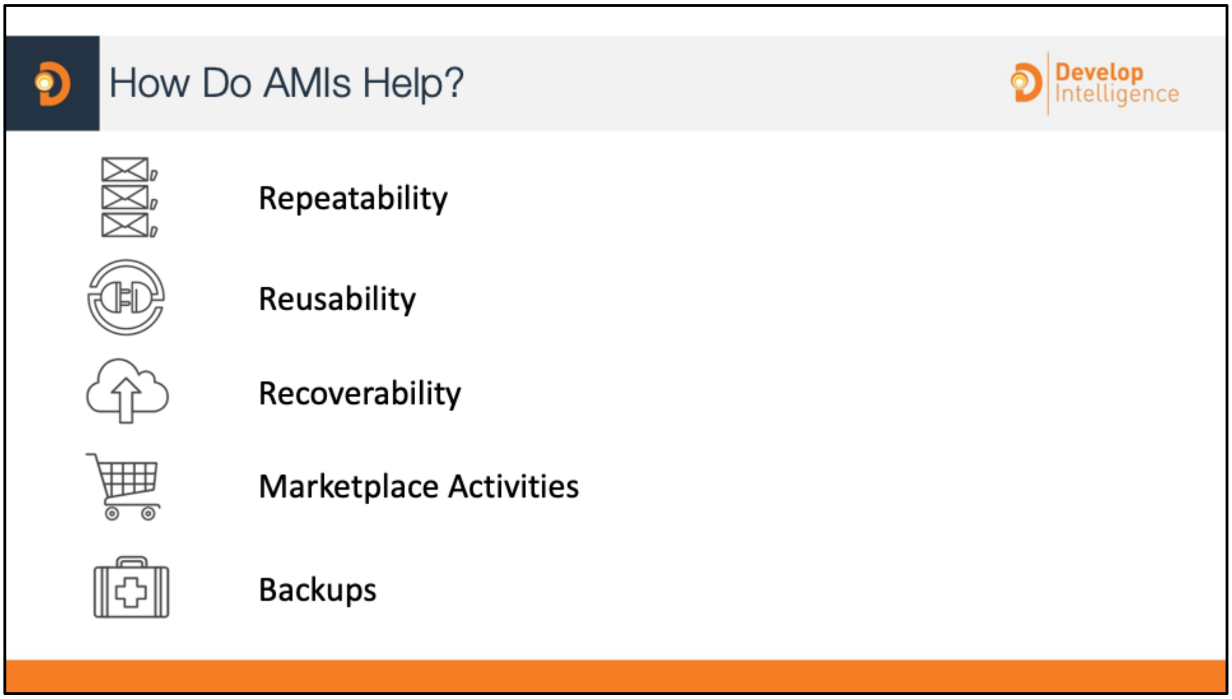


How Do AMIs Help?

Develop Intelligence

- Repeatability
- Reusability
- Recoverability
- Marketplace Solutions

Marketplace: If you are looking for a software solution from a specific vendor, there is probably an AMI on the marketplace you can launch to implement that solution on an EC2 instance. Additionally, authorized software vendors can create AMIs and sell them there as well.

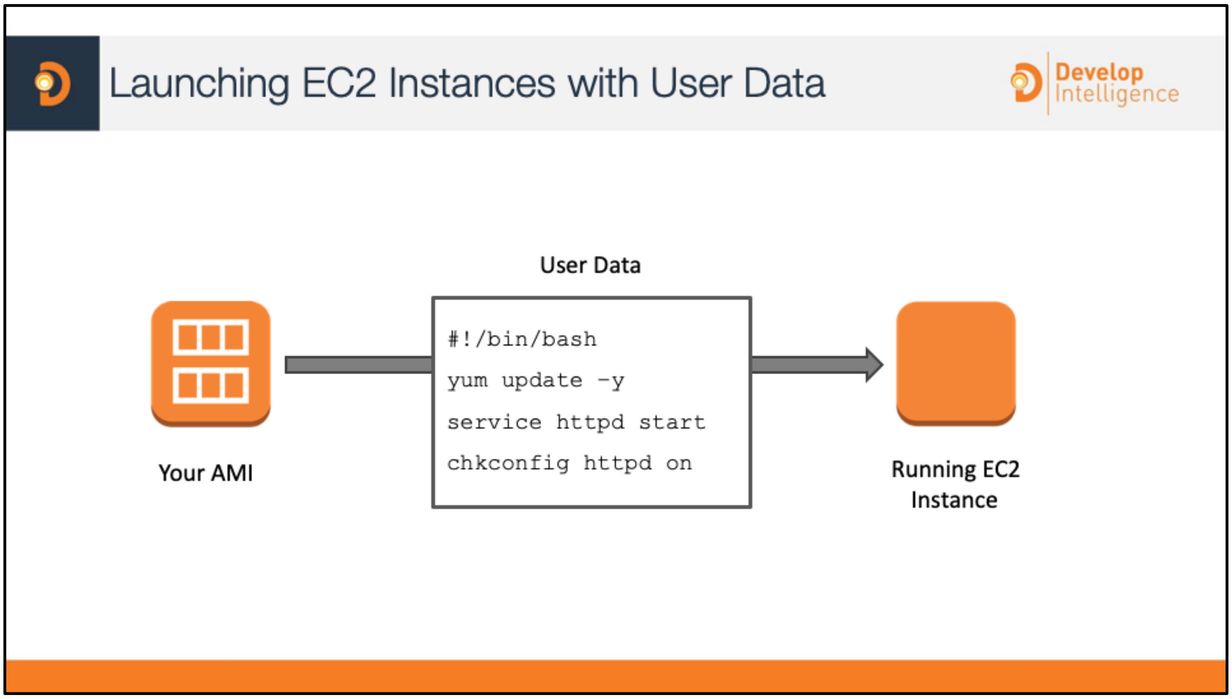


How Do AMIs Help?

Develop Intelligence

- Repeatability
- Reusability
- Recoverability
- Marketplace Activities
- Backups

Backup: AMIs provide a great way to back up a complete EC2 instance configuration, which you can use to launch a replacement instance in the event of a failure.

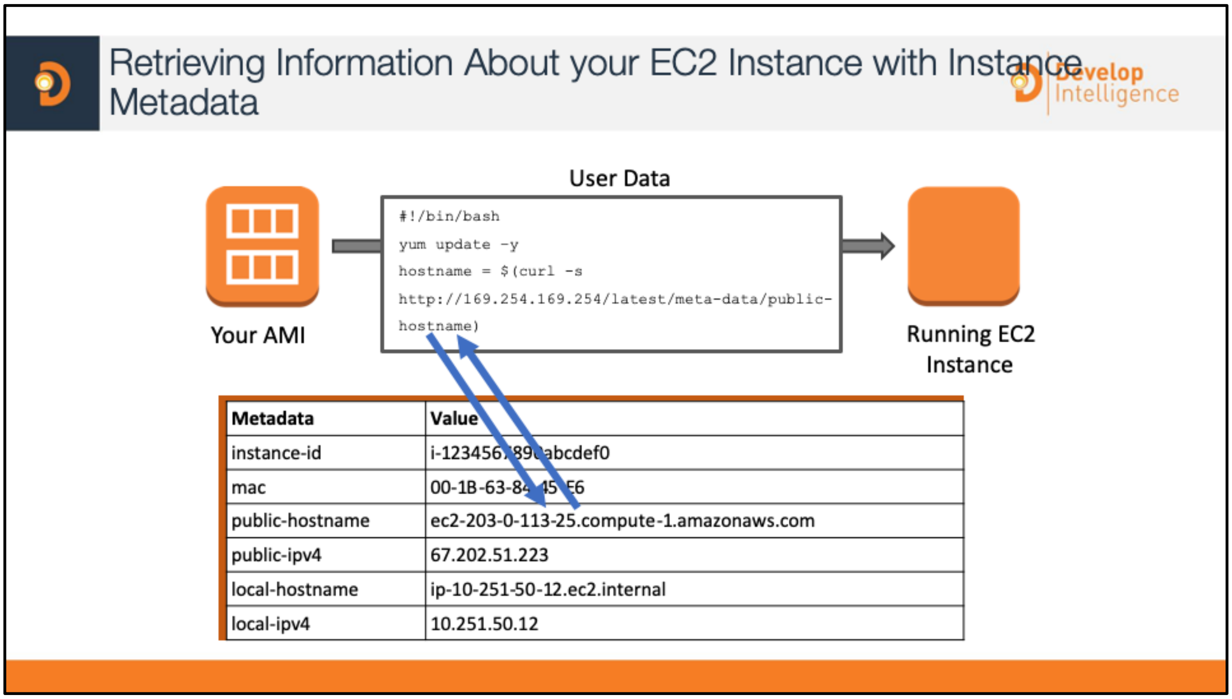


Launching EC2 Instances with User Data

User Data

```
#!/bin/bash
yum update -y
service httpd start
chkconfig httpd on
```

When creating your EC2 instances, you have the option of passing *user data* to the instance. User data can automate the completion of the instance launch. For example, it might patch and update the instance AMI, fetch and install software license keys, or install additional software. User data is implemented as a shell script or cloud-init directive that executes with root or Administrator privilege after the instances starts but before it becomes accessible on the network.



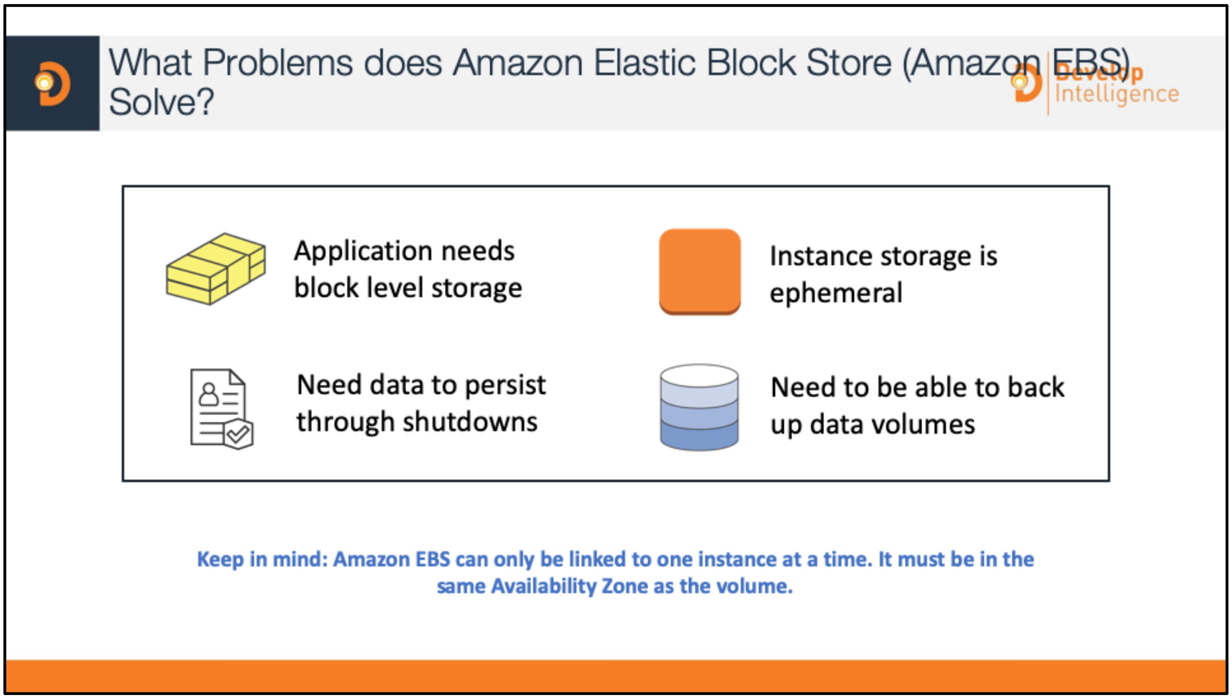
Retrieving Information About your EC2 Instance with Instance Metadata

```
#!/bin/bash
yum update -y
hostname = $(curl -s
http://169.254.169.254/latest/meta-data/public-
hostname)
```

Metadata	Value
instance-id	i-1234567890abcdef0
mac	00-1B-63-84-45-E6
public-hostname	ec2-203-0-113-25.compute-1.amazonaws.com
public-ipv4	67.202.51.223
local-hostname	ip-10-251-50-12.ec2.internal
local-ipv4	10.251.50.12

In order for User Data to complete the launch of a new EC2 instance, it may need to look up information about the instance itself. For example, it might need to learn and share the public IP address, hostname, or mac address of the new instance to complete the launch. The Instance Metadata Service can provide that information.

https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/ec2-instance-metadata.html



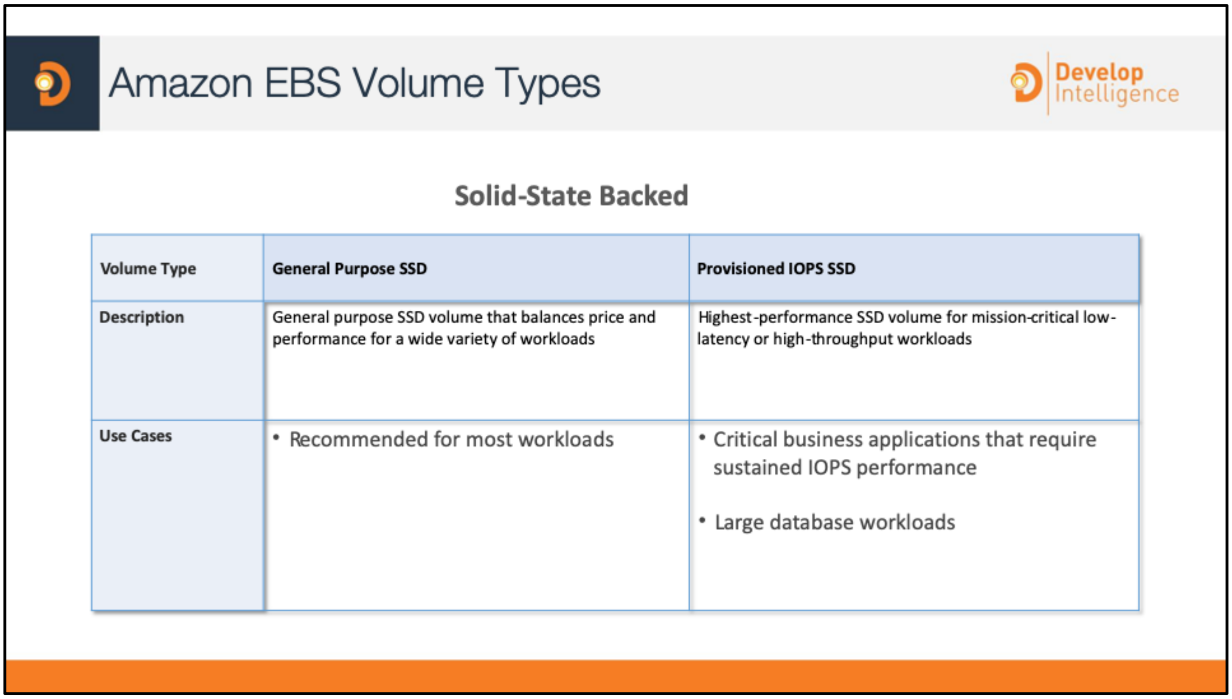
What Problems does Amazon Elastic Block Store (Amazon EBS) Solve?

- Application needs block level storage
- Instance storage is ephemeral
- Need data to persist through shutdowns
- Need to be able to back up data volumes

Keep in mind: Amazon EBS can only be linked to one instance at a time. It must be in the same Availability Zone as the volume.

Amazon EBS volumes provide durable, detachable, block-level storage (like an external hard drive) for your Amazon EC2 instances. Because they are directly attached to the instances, they can provide extremely low latency between where the data is stored and where it might be used on the instance. For this reason, they can be used to run a database with an Amazon EC2 instance. Amazon EBS volumes can also be used to back up your instances into AMIs, which are stored in Amazon S3 and can be reused to create new Amazon EC2 instances later.

An *instance store* provides temporary block-level storage for your instance. This storage is located on disks that are physically attached to the host computer. Instance store is ideal for temporary storage of information that changes frequently, such as buffers, caches, scratch data, and other temporary content, or for data that is replicated across a fleet of instances, such as a load-balanced pool of web servers.



Amazon EBS Volume Types

Develop Intelligence

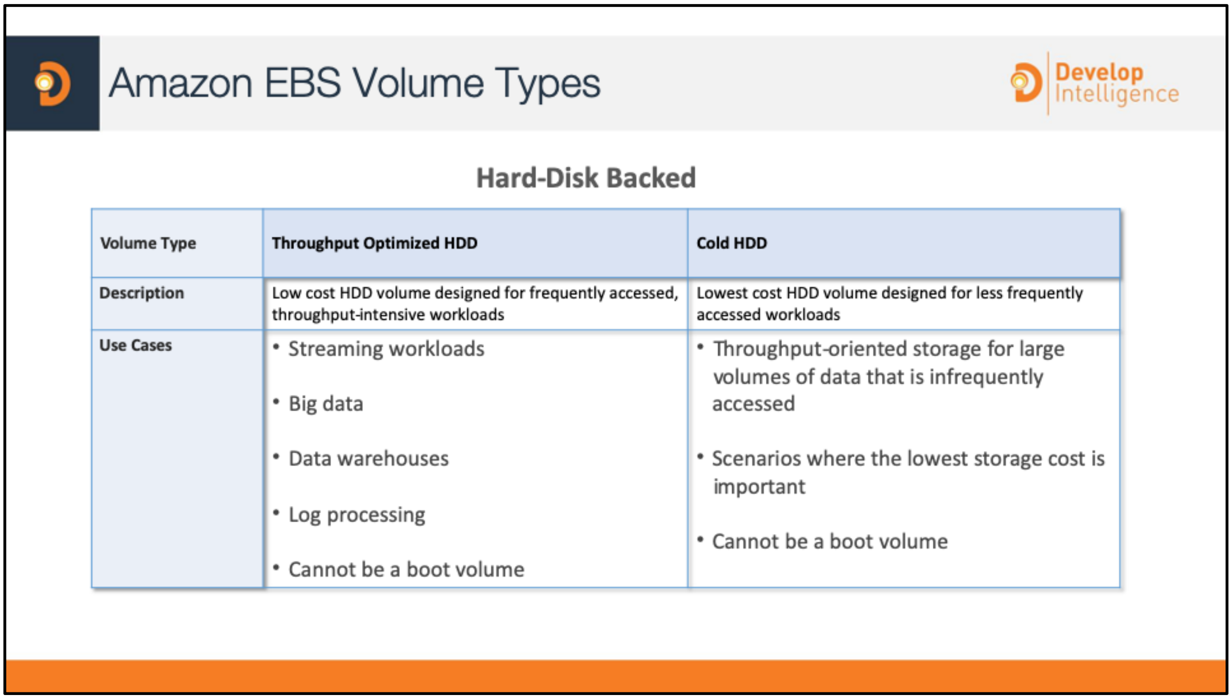
Solid-State Backed

Volume Type	General Purpose SSD	Provisioned IOPS SSD
Description	General purpose SSD volume that balances price and performance for a wide variety of workloads	Highest-performance SSD volume for mission-critical low-latency or high-throughput workloads
Use Cases	• Recommended for most workloads	• Critical business applications that require sustained IOPS performance • Large database workloads

SSD-backed volumes are optimized for transactional workloads involving frequent read/write operations with small I/O size, where the dominant performance attribute is IOPS.

HDD-backed volumes are optimized for large streaming workloads where throughput (measured in MiB/s) is a better performance measure than IOPS.

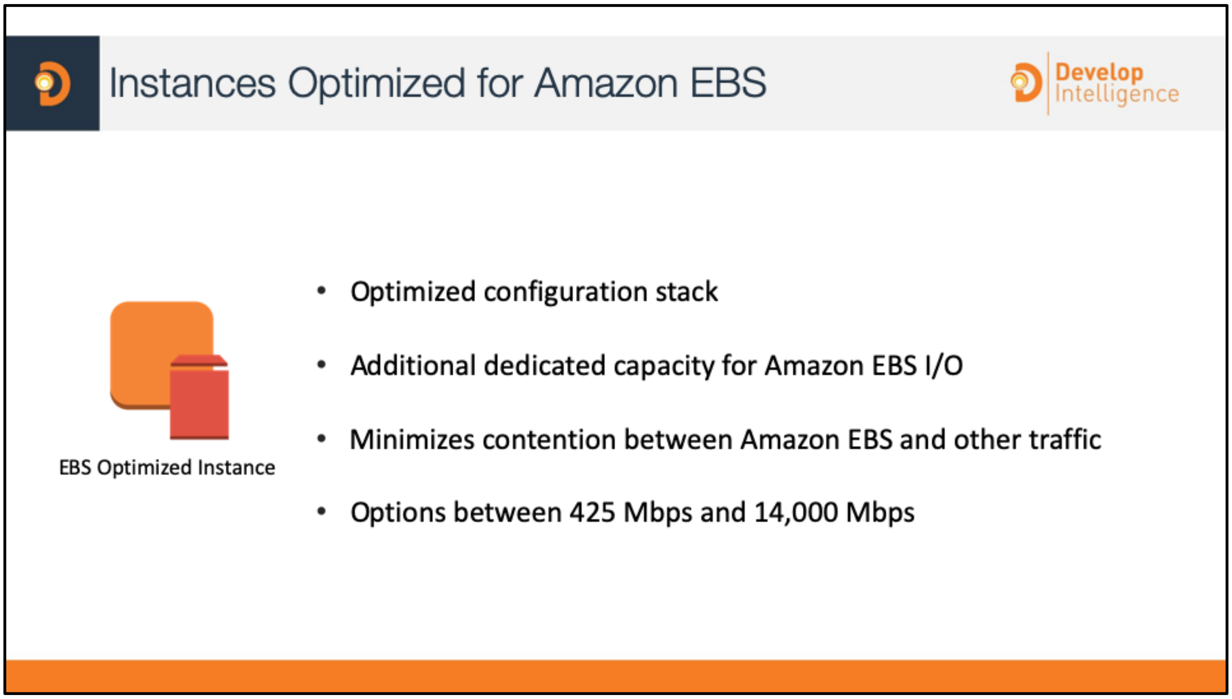
For more information about Amazon EBS volume types, see https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/EBSVolumeTypes.html



Amazon EBS Volume Types

Hard-Disk Backed

Volume Type	Throughput Optimized HDD	Cold HDD
Description	Low cost HDD volume designed for frequently accessed, throughput-intensive workloads	Lowest cost HDD volume designed for less frequently accessed workloads
Use Cases	• Streaming workloads • Big data • Data warehouses • Log processing • Cannot be a boot volume	• Throughput-oriented storage for large volumes of data that is infrequently accessed • Scenarios where the lowest storage cost is important • Cannot be a boot volume



Instances Optimized for Amazon EBS

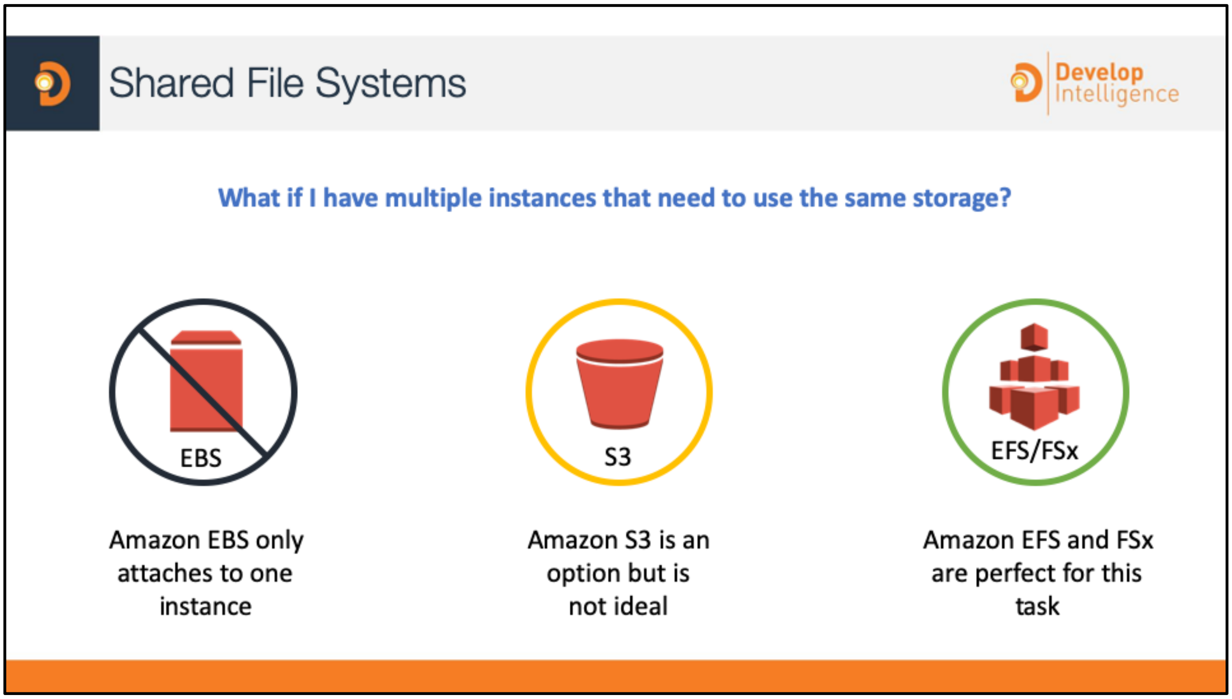
EBS Optimized Instance

- Optimized configuration stack
- Additional dedicated capacity for Amazon EBS I/O
- Minimizes contention between Amazon EBS and other traffic
- Options between 425 Mbps and 14,000 Mbps

An instance optimized for Amazon EBS uses an optimized configuration stack and provides additional dedicated capacity for Amazon EBS I/O. This optimization provides the best performance for your EBS volumes by minimizing contention between Amazon EBS I/O and other traffic from your instance.
EBS-optimized instances deliver dedicated bandwidth to Amazon EBS, with options between 425 Mbps and 14,000 Mbps, depending on the instance type you use.

For more information, see
https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/EBSOptimized.html

If your EC2 instance is EBS-backed, you can put it into EC2 Hibernation. This keeps the in-memory storage, private IP, and elastic IP to remain the same, and allows you to pick up when the instance left off. It currently can only be enabled on Linux1 EC2 instances, with Linux2 support coming soon. When the instance is in hibernation, you pay only for the EBS volume attached, and the Elastic IP in your account.
https://aws.amazon.com/blogs/aws/new-hibernate-your-ec2-instances/



Shared File Systems

Develop Intelligence

What if I have multiple instances that need to use the same storage?

EBS

Amazon EBS only attaches to one instance

S3

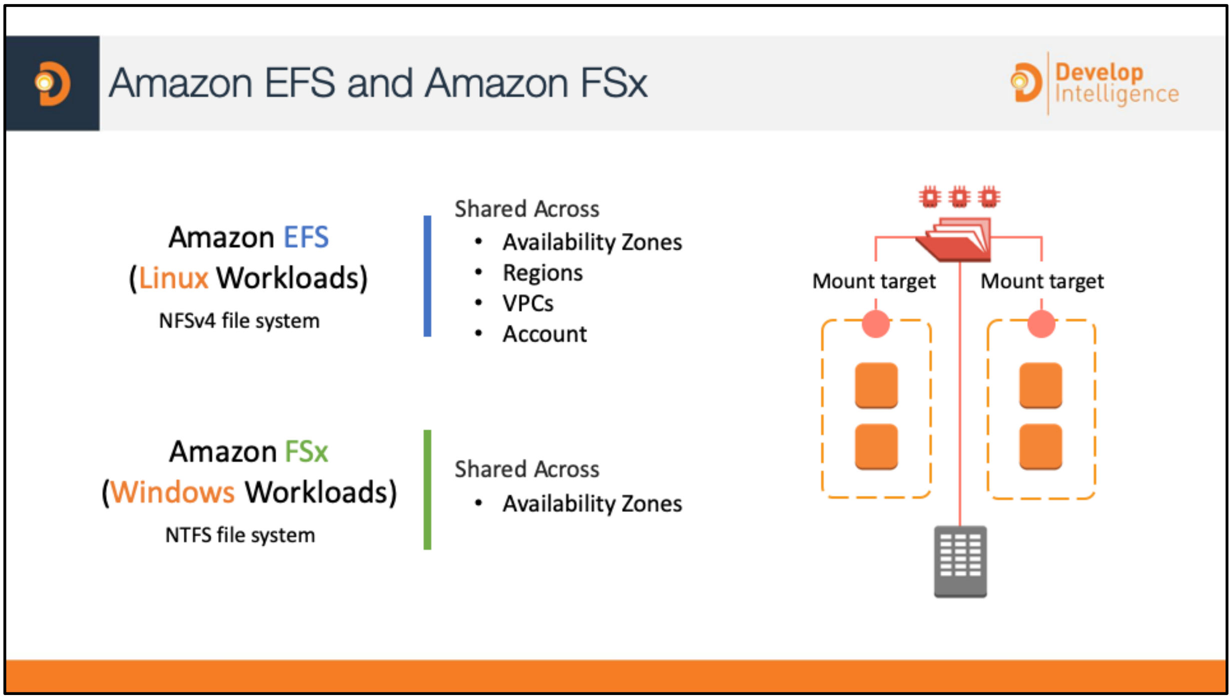
Amazon S3 is an option but is not ideal

EFS/FSx

Amazon EFS and FSx are perfect for this task

Problem: How do we handle an application running on multiple instances that needs to use the same file system? Amazon S3 is one option, but what if you need the performance and read-write consistency of a network file system? Amazon Elastic File System (Amazon EFS) may be your best option.

S3 is an object store system, not a block store, so changes overwrite entire files, not blocks of characters within files. For high throughput changes to files of varying sizes, a file system will be superior to an object store system for handling those changes.



Amazon EFS and Amazon FSx

Develop Intelligence

Amazon EFS
(Linux Workloads)
NFSv4 file system

Shared Across
- Availability Zones
- Regions
- VPCs
- Account

Amazon FSx
(Windows Workloads)
NTFS file system

Shared Across
- Availability Zones

Amazon EFS is a regional file storage service residing, within and across multiple Availability Zones for high availability and durability. You'll be able to access your file system across Availability Zones, AWS Regions, and VPCs while sharing files between thousands of EC2 instances and on-premises servers via Direct Connect or AWS VPN. With Amazon EFS, you can create a file system, mount the file system on an Amazon EC2 instance, and then read and write data to and from your file system. You can mount an Amazon EFS file system in your VPC, through the Network File System versions 4.0 and 4.1 (NFSv4) protocol.

For more information about this, see https://aws.amazon.com/about-aws/whats-new/2018/11/amazon-efs-now-supports-access-across-accounts-and-vpcs/

For more information about VPC peering, see https://docs.aws.amazon.com/efs/latest/ug/manage-fs-access-vpc-peering.html

For a list of Amazon EC2 Linux Amazon Machine Images (AMIs) that support this protocol, see NFS Support. We recommend using a current generation Linux NFSv4.1 client, such as those found in Amazon Linux and Ubuntu AMIs. For some AMIs, you'll need to install an NFS client to mount your file system on your Amazon EC2 instance. For instructions, see Installing the NFS Client.

You can access your Amazon EFS file system concurrently from Amazon EC2 instances in your Amazon VPC, so applications that scale beyond a single connection can access a file system. Amazon EC2 instances running in multiple Availability Zones within the same region can access the file system, so that many users can access and share a common data source.

Note the following restrictions:

- You can mount an Amazon EFS file system on instances in only one VPC at a time.
- Both the file system and VPC must be in the same AWS Region.

How is File Storage Different?

Although object storage solutions enable storage of files as objects, accessing with existing applications requires new code and the use of APIs and direct knowledge of naming semantics. File storage solutions that support existing file system semantics and permissions models have a distinct advantage in that they do not require new code to be written to integrate with applications that are easily configured to work with shared file storage.

Block storage can be used as the underlying storage component of a self-managed file storage solution. However, the one-to-one relationship required between the host and volume makes it difficult to have the scalability, availability, and affordability of a fully managed file storage solution and would require additional budget and management resources to support. Using a fully managed cloud file storage solution removes complexities, reduces costs, and simplifies management.

Amazon FSx for Windows File Server provides a shared file storage system for your Windows Amazon EC2 instances with high levels of throughput and sub-millisecond latency. Amazon FSx supports the following:

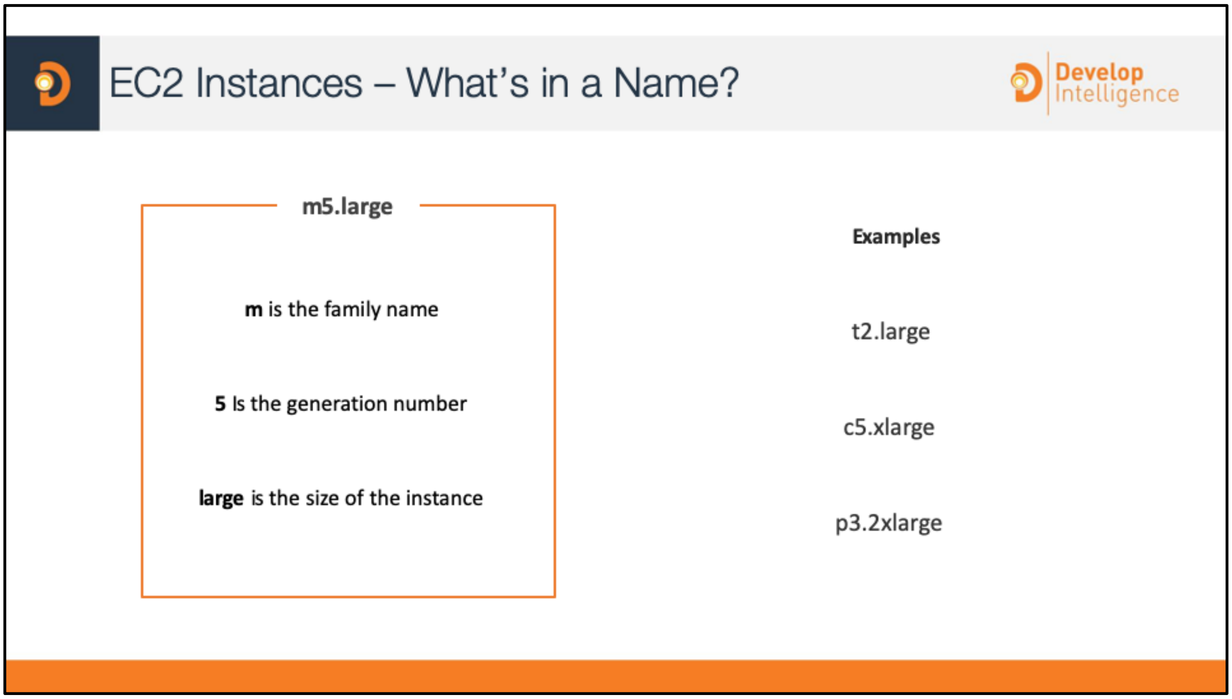
- SMB Protocol
- Windows NTFS
- Active Directory (AD) Integration
- Distributed File System (DFS)

Amazon FSx for Windows File Server is ideal for supporting Windows workloads that require shared storage such as CRM, ERP, .NET applications, and user home directories. Thousands of compute instances can access a single Amazon FSx file system at the same time.

Amazon FSx for Lustre provides a similar, fully managed file system that is optimized for high performance computing (HPC), machine learning, and media processing workflows. A single Amazon FSx for Luster file system can processive massive data sets with hundreds of gigabytes (GB) per second of throughput at sub-millisecond latencies.

Amazon FSx for Luster can be integrated with Amazon S3, so that you can join long-term data sets with a high performance file system. Data can be automatically copied to and from Amazon S3 from your Amazon FSx for Luster file system.

Both Amazon FSx offerings support connection with on-premises workloads using Amazon Direct Connect or a VPN connection. With both offerings, you pay only for the resources you use.



EC2 Instances – What's in a Name?

Develop Intelligence

m5.large

m is the family name

5 Is the generation number

large is the size of the instance

Examples

t2.large

c5.xlarge

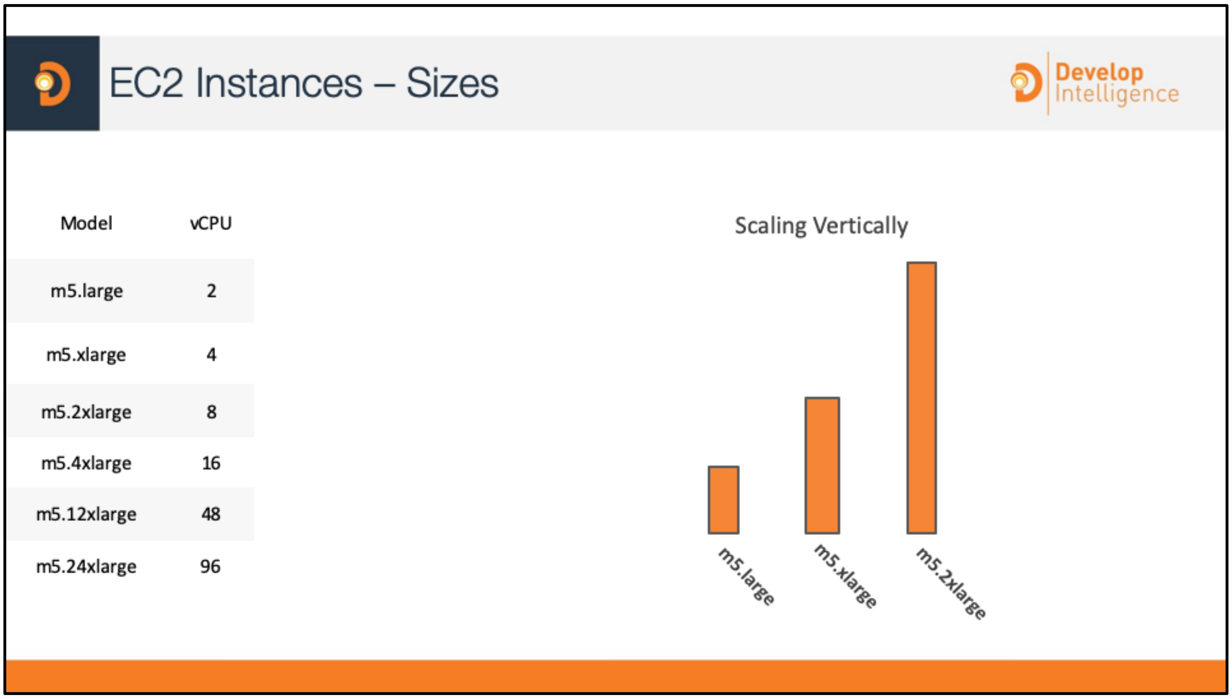
p3.2xlarge

When looking at an instance type, you will see that the model has a few parts to its name—as an example, take the M type.

M is the family name, which is then followed up by a number. Here, that number is 5. The number is the generation number of that type. So, an M5 instance is the 5th generation of the M family. In general, instances of a higher generation are more powerful and provide a better value for the price.

The next part of the name is the size portion of the instance. When comparing sizes, it's important to look at the coefficient portion of the size category. For example, an m5.2xlarge is twice as big as a m5.xlarge. This m5.xlarge is in turn twice as big as the m5.large. An m4.10xlarge (which appears in a chart later in this course) is 10 times as powerful as the m5.xlarge.

It is also important to note that network bandwidth is also tied to the size of your EC2 instance. If you are performing a highly network-intensive task, you might be required to increase your instance specs in order to meet those needs.



EC2 Instances – Sizes

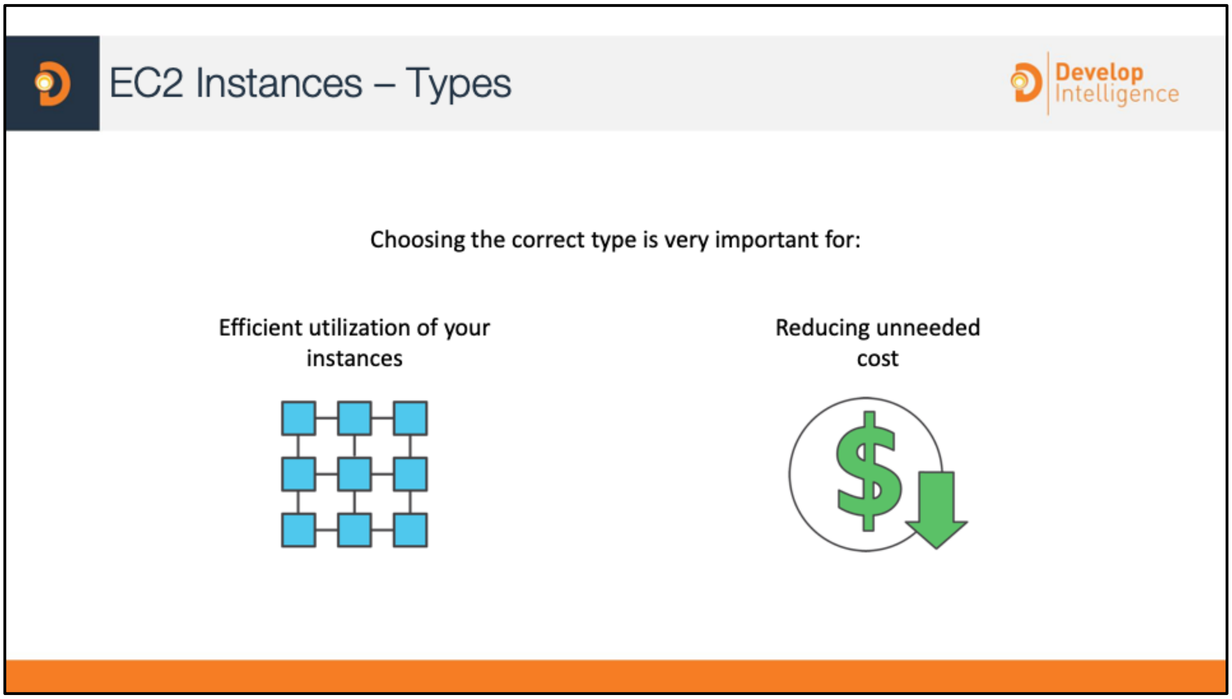
Model	vCPU
m5.large	2
m5.xlarge	4
m5.2xlarge	8
m5.4xlarge	16
m5.12xlarge	48
m5.24xlarge	96

The next part of the name is the size portion of the instance. When comparing sizes, it’s important to look at the coefficient portion of the size category.

For example a m5.2xlarge is twice as big as a m5.xlarge. This m5.xlarge is in turn twice as big as the m5.large.

You will notice later on in the chart there is a m5.12xlarge. This instance is 12 times as powerful as the m5.xlarge.

It is also important to note that network bandwidth is also tied to the size of your ec2 instance. If you are performing a task that is very network intensive you might be required to increase your instance specs in order to meet those needs.



EC2 Instances – Types

Choosing the correct type is very important for:

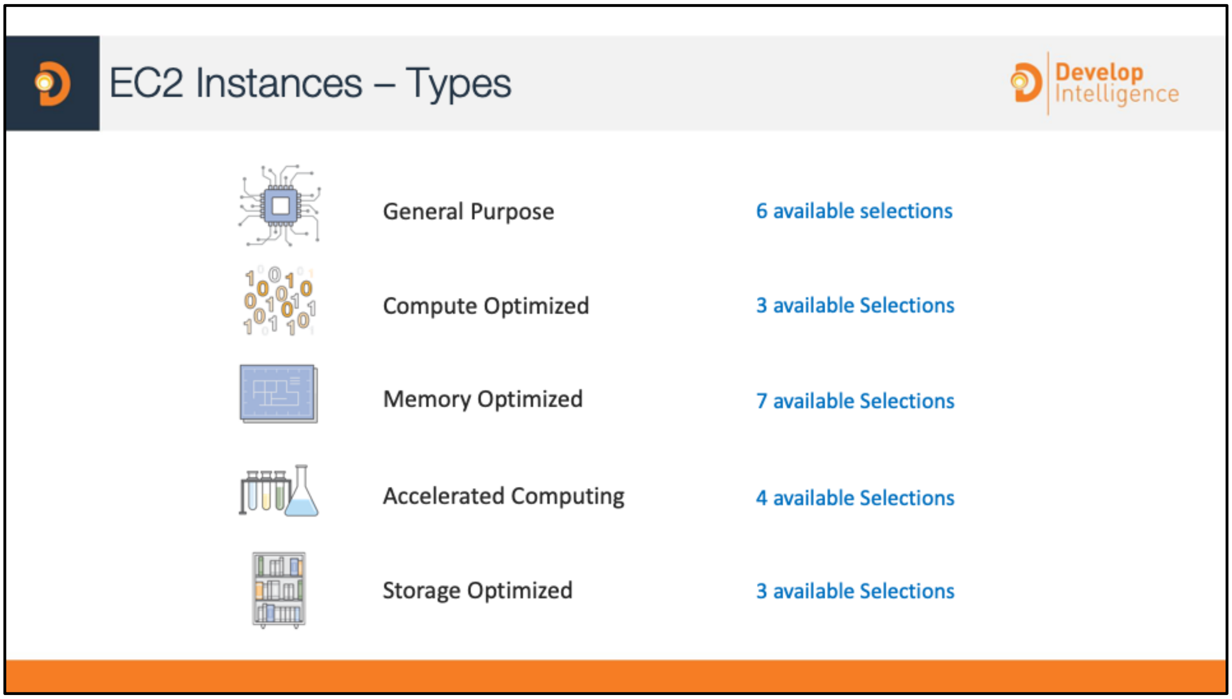
Efficient utilization of your instances

Reducing unneeded cost

Choosing the correct instance type is very important for reducing unneeded cost and increasing utilization of an instance.

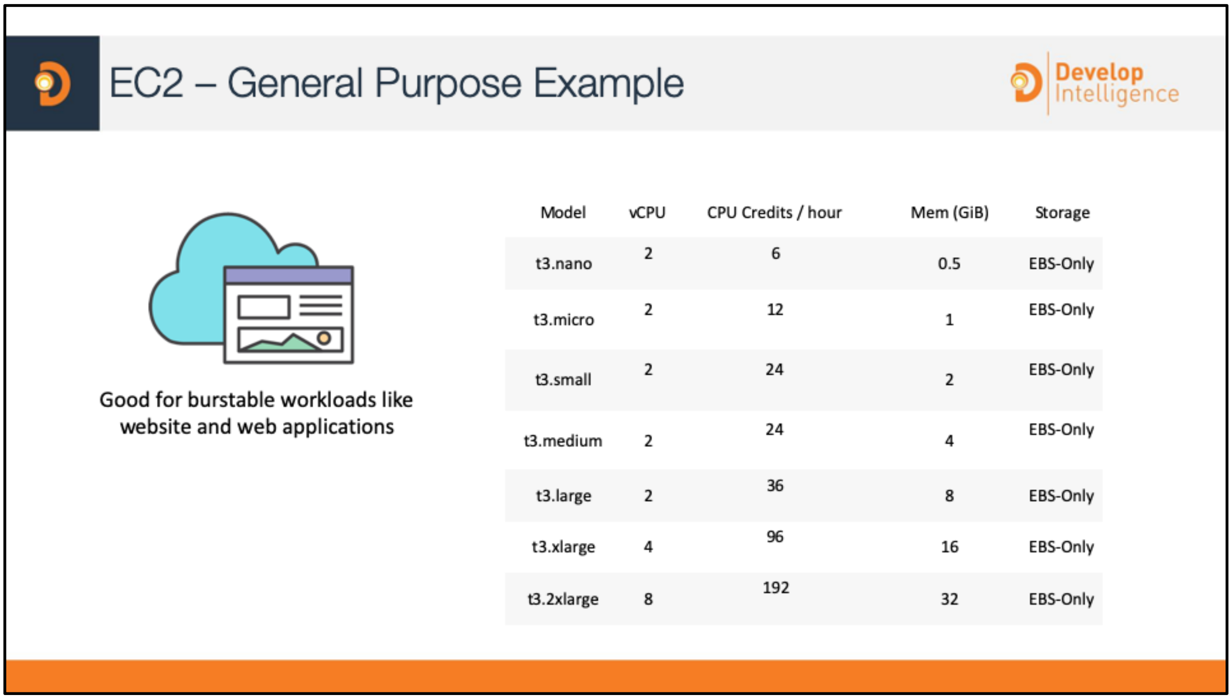
Each instance family has its own positives that need to be addressed when deciding how you are going to architect your solution.

Let’s take a look at all of the instance families and see what their recommended workloads are.



EC2 Instances – Types

General Purpose	6 available selections
Compute Optimized	3 available Selections
Memory Optimized	7 available Selections
Accelerated Computing	4 available Selections
Storage Optimized	3 available Selections



EC2 – General Purpose Example

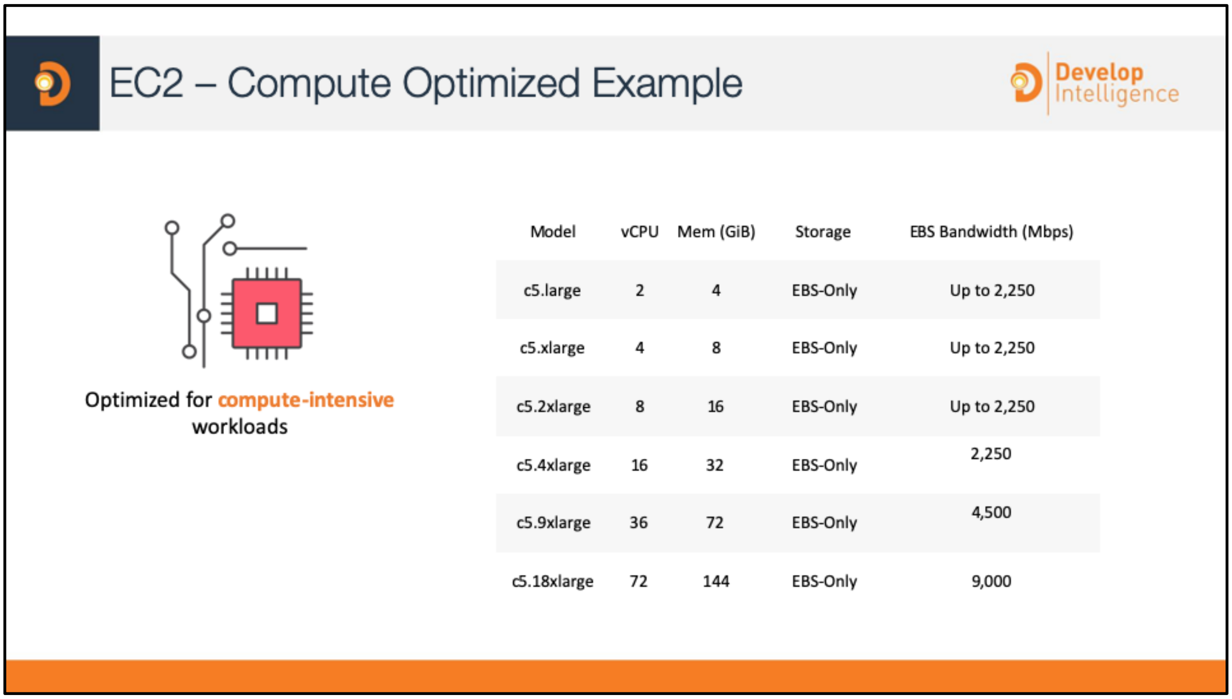
Develop Intelligence

Good for burstable workloads like website and web applications

Model	vCPU	CPU Credits / hour	Mem (GiB)	Storage
t3.nano	2	6	0.5	EBS-Only
t3.micro	2	12	1	EBS-Only
t3.small	2	24	2	EBS-Only
t3.medium	2	24	4	EBS-Only
t3.large	2	36	8	EBS-Only
t3.xlarge	4	96	16	EBS-Only
t3.2xlarge	8	192	32	EBS-Only

T2 instances are burstable performance instances that provide a baseline level of CPU performance with the ability to burst above the baseline.

Use cases for this type of instance include websites and web applications, development environments, build servers, code repositories, micro services, test and staging environments, and line of business applications.



EC2 – Compute Optimized Example

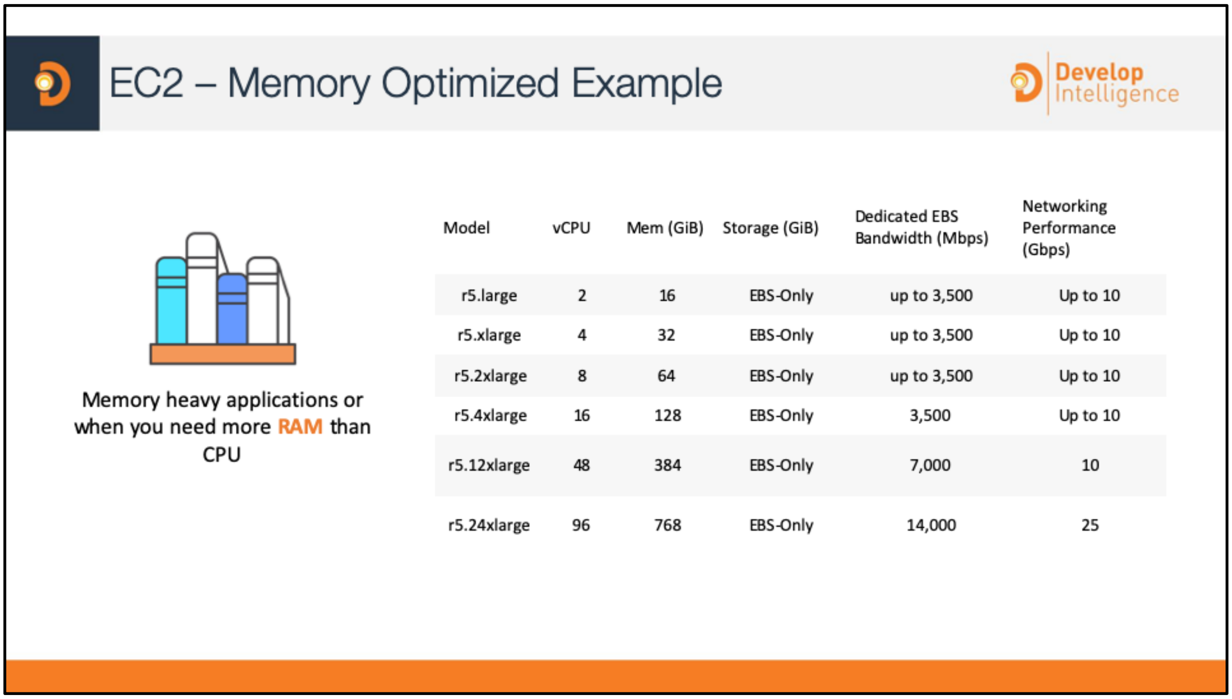
Develop Intelligence

Optimized for **compute-intensive** workloads

Model	vCPU	Mem (GiB)	Storage	EBS Bandwidth (Mbps)
c5.large	2	4	EBS-Only	Up to 2,250
c5.xlarge	4	8	EBS-Only	Up to 2,250
c5.2xlarge	8	16	EBS-Only	Up to 2,250
c5.4xlarge	16	32	EBS-Only	2,250
c5.9xlarge	36	72	EBS-Only	4,500
c5.18xlarge	72	144	EBS-Only	9,000

C5 instances are optimized for compute-intensive workloads and deliver very cost-effective high performance at a low price per compute ratio.

Use cases include high-performance web servers, scientific modelling, batch processing, distributed analytics, high-performance computing (HPC), machine/deep learning inference, ad serving, highly scalable multiplayer gaming, and video encoding.



EC2 – Memory Optimized Example

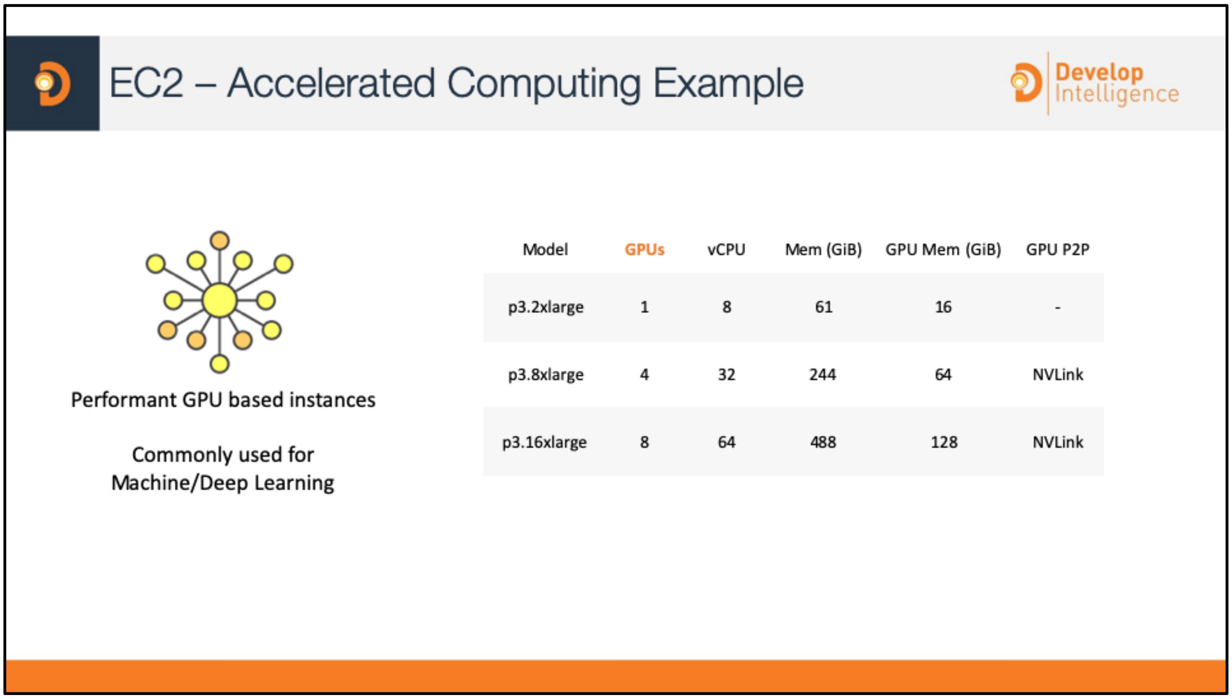
Develop Intelligence

Memory heavy applications or when you need more RAM than CPU

Model	vCPU	Mem (GiB)	Storage (GiB)	Dedicated EBS Bandwidth (Mbps)	Networking Performance (Gbps)
r5.large	2	16	EBS-Only	up to 3,500	Up to 10
r5.xlarge	4	32	EBS-Only	up to 3,500	Up to 10
r5.2xlarge	8	64	EBS-Only	up to 3,500	Up to 10
r5.4xlarge	16	128	EBS-Only	3,500	Up to 10
r5.12xlarge	48	384	EBS-Only	7,000	10
r5.24xlarge	96	768	EBS-Only	14,000	25

R4 instances are optimized for memory-intensive applications.

Use cases include high-performance databases, data mining and analysis, in-memory databases, distributed web scale in-memory caches, applications performing real-time processing of unstructured big data, Hadoop/Spark clusters, and other enterprise applications.



EC2 – Accelerated Computing Example

Develop Intelligence

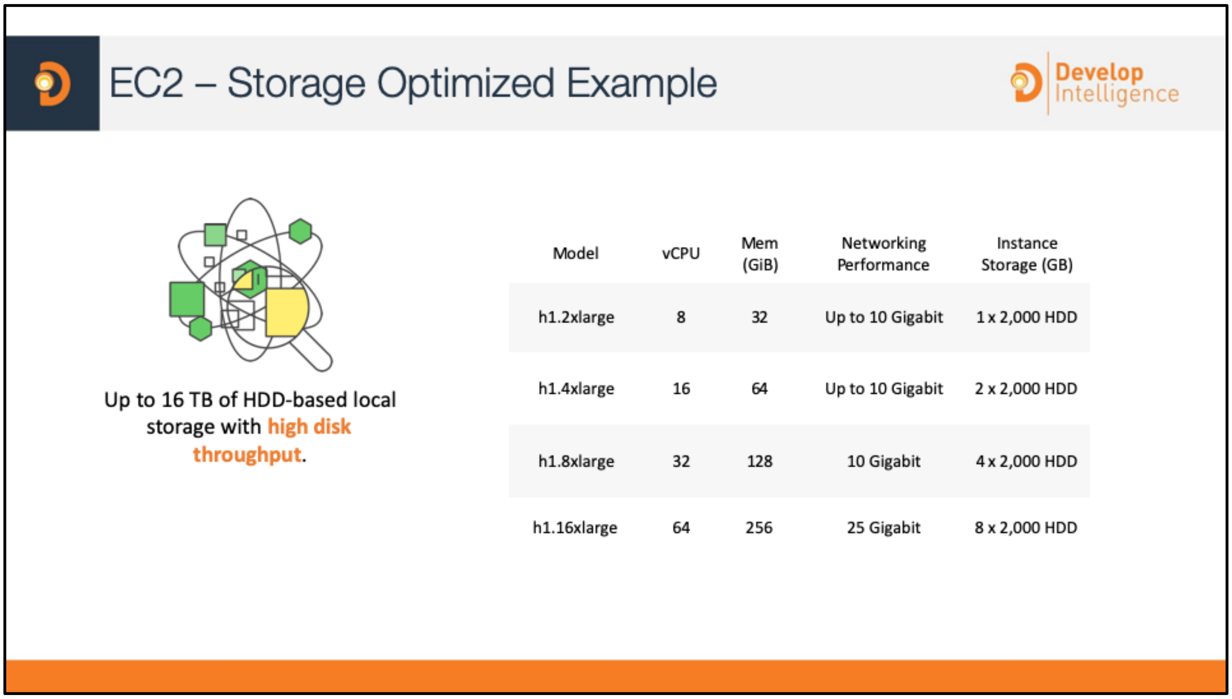
Performant GPU based instances

Commonly used for Machine/Deep Learning

Model	GPUs	vCPU	Mem (GiB)	GPU Mem (GiB)	GPU P2P
p3.2xlarge	1	8	61	16	-
p3.8xlarge	4	32	244	64	NVLink
p3.16xlarge	8	64	488	128	NVLink

P3 instances are intended for general-purpose GPU compute applications.

Use cases include machine learning, deep learning, high-performance computing, computational fluid dynamics, computational finance, seismic analysis, speech recognition, autonomous vehicles, and drug discovery.



EC2 – Storage Optimized Example

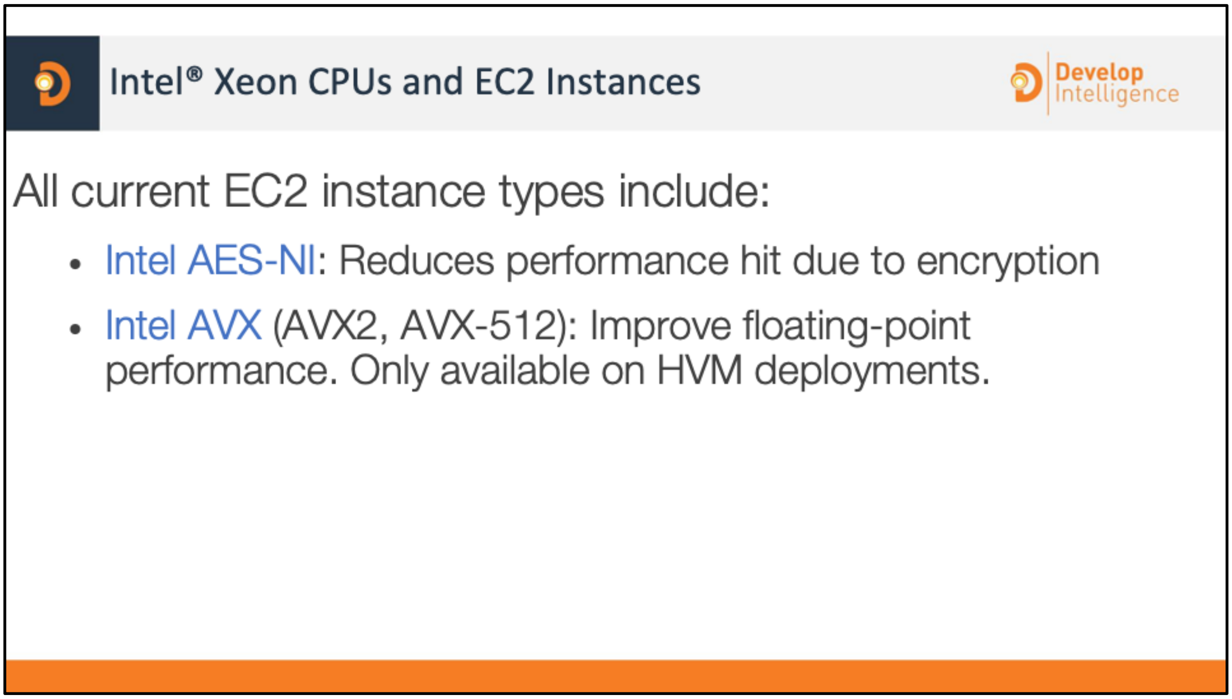
Develop Intelligence

Up to 16 TB of HDD-based local storage with **high disk throughput**.

Model	vCPU	Mem (GiB)	Networking Performance	Instance Storage (GB)
h1.2xlarge	8	32	Up to 10 Gigabit	1 x 2,000 HDD
h1.4xlarge	16	64	Up to 10 Gigabit	2 x 2,000 HDD
h1.8xlarge	32	128	10 Gigabit	4 x 2,000 HDD
h1.16xlarge	64	256	25 Gigabit	8 x 2,000 HDD

H1 instances feature up to 16 TB of HDD-based local storage, deliver high disk throughput, and a balance of compute and memory.

Use cases include Amazon EMR-based workloads, distributed file systems such as HDFS and MapR-FS, network file systems, log or data processing applications such as Apache Kafka, and big data workload clusters.



Intel® Xeon CPUs and EC2 Instances

Develop Intelligence

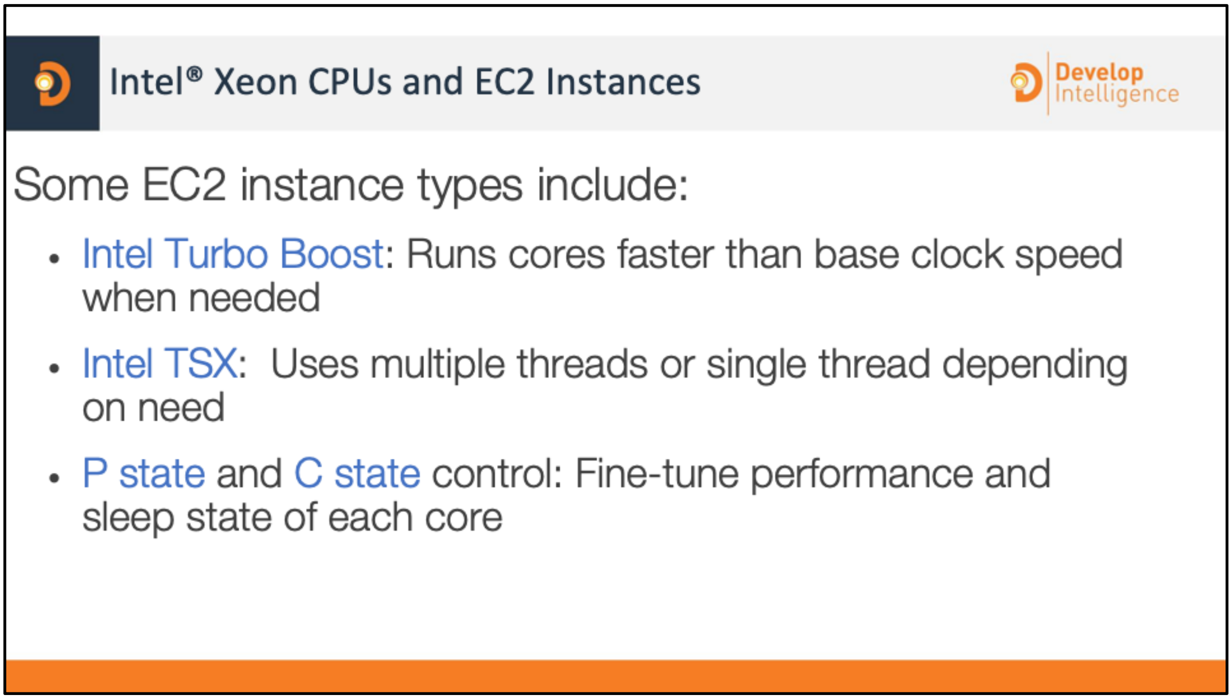
All current EC2 instance types include:

- Intel AES-NI: Reduces performance hit due to encryption
- Intel AVX (AVX2, AVX-512): Improve floating-point performance. Only available on HVM deployments.

All current EC2 instance types that use Intel processors include Intel's Advanced Encryption Standard New Instructions (AES-NI), which reduces the performance hit your processor takes when you enable encryption.

All instance types also include some form of Intel Advanced Vector Extension (AVX), which is Intel's instructions custom-built for floating-point intensive workloads. AVX2 provides twice the floating point performance of AVX, and AVX-512, available only on the new Intel Xeon Scalable Processor family of CPUs, doubles the performance of AVX2.

Intel Transactional Synchronization Extensions (TSX): Provides workload optimized performance specific to the applications, multi-threaded when needed and single threaded when needed



Intel® Xeon CPUs and EC2 Instances

Some EC2 instance types include:

- Intel Turbo Boost: Runs cores faster than base clock speed when needed
- Intel TSX: Uses multiple threads or single thread depending on need
- P state and C state control: Fine-tune performance and sleep state of each core

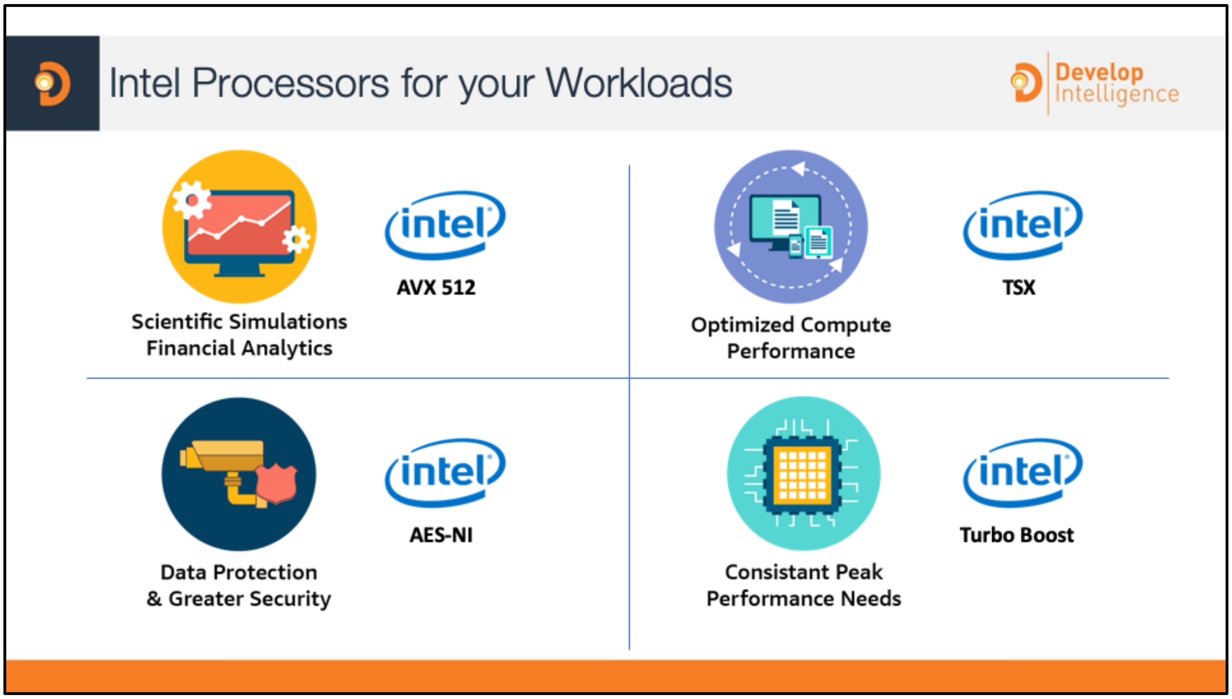
Some instance types also include Intel Turbo Boost, Intel TSX, and P State and C State control.

Intel Turbo Boost intelligently boosts the clock speed of cores based on need.

Intel Transactional Synchronization Extensions (TSX): Provides workload optimized performance specific to the applications, multi-threaded when needed and single threaded when needed.

P state and C state control allows you to tune the performance and sleep state of each core to your own needs.

To find out which instance types currently support these options, see the AWS instance types page: https://aws.amazon.com/ec2/instance-types/



Intel Processors for your Workloads

Develop Intelligence

Scientific Simulations
Financial Analytics

intel
AVX 512

Optimized Compute
Performance

intel
TSX

Data Protection
& Greater Security

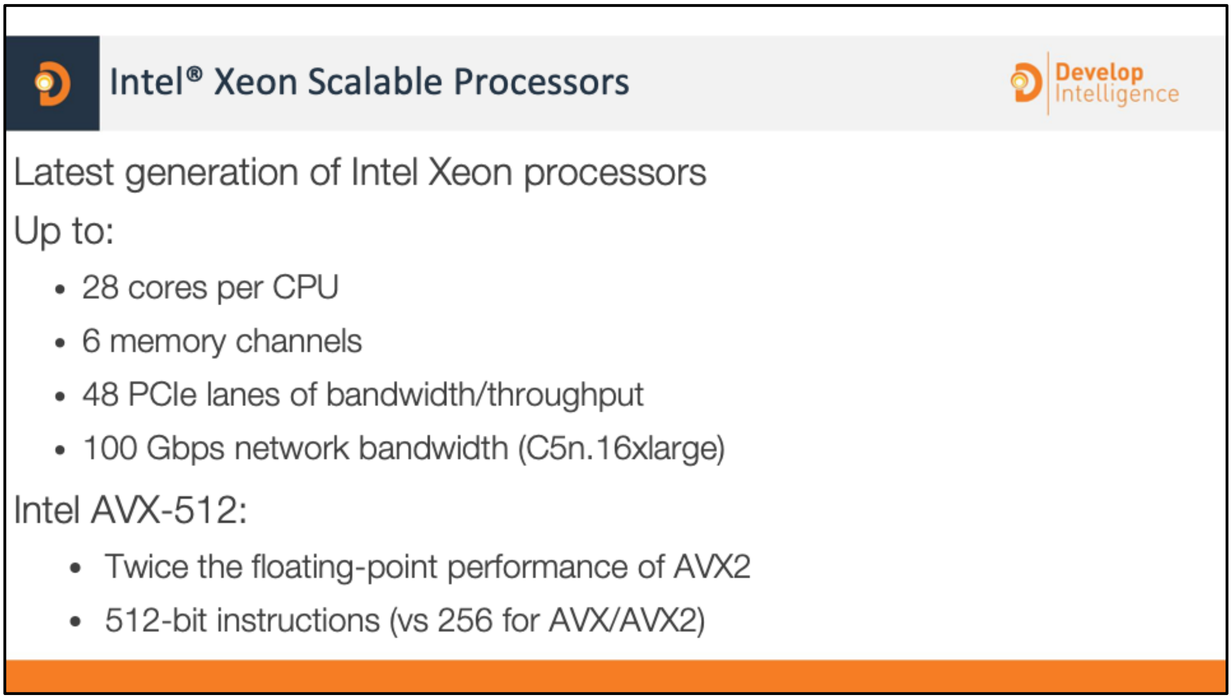
intel
AES-NI

Consistant Peak
Performance Needs

intel
Turbo Boost

There are several different Intel processors to fit different workloads.

- **Intel® AVX 512:** Optimized for: scientific simulations, financial analytics, artificial intelligence (AI)/deep learning, 3D modeling and analysis, image and audio/video processing, cryptography and data compression.
- **Intel® AES-NI:** Intel® AES-NI provides faster data protection and greater security; making pervasive encryption feasible in areas where previously it was not.
- **Intel® TSX:** Intel® Transactional Synchronization Extensions (Intel® TSX) allows the processor to determine dynamically whether threads need to serialize through lock-protected critical sections, and to perform serialization only when required. Optimizing compute performance for business applications dynamically
- **Intel® Turbo Boost:** Intel® Turbo Boost Technology 2.0 accelerates processor and graphics performance for peak loads, automatically allowing processor cores to run faster than the rated operating frequency if they're operating below power, current, and temperature specification limits.



Intel® Xeon Scalable Processors

Latest generation of Intel Xeon processors

Up to:

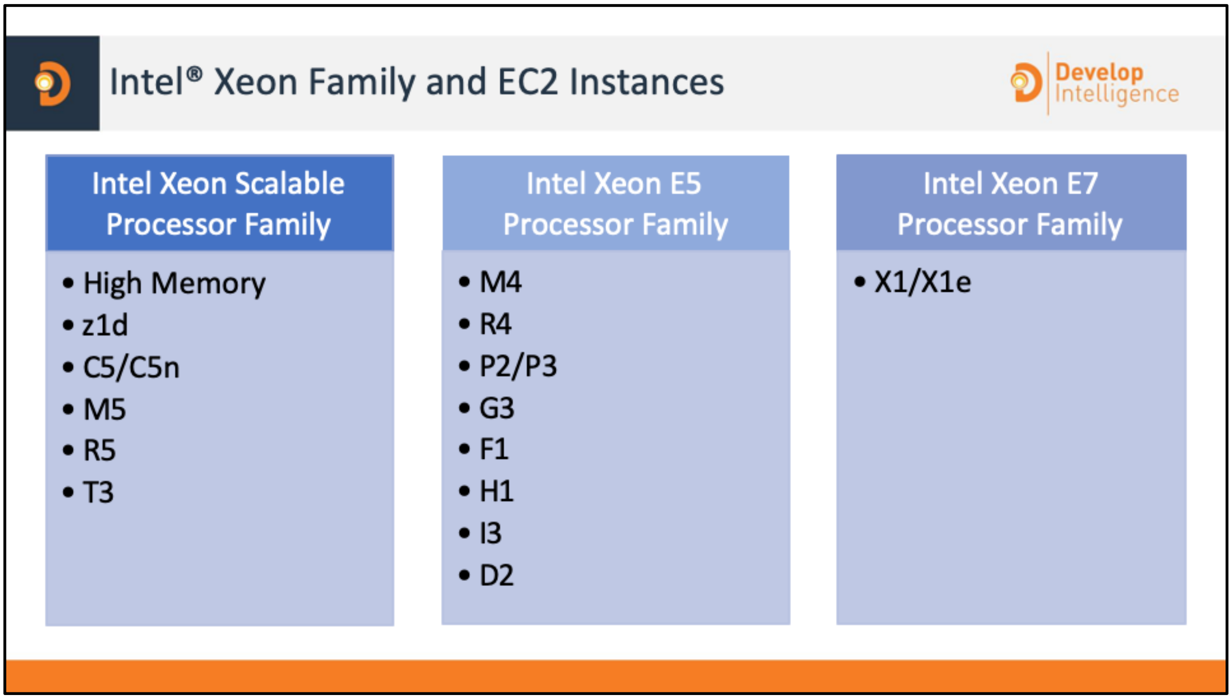
- 28 cores per CPU
- 6 memory channels
- 48 PCIe lanes of bandwidth/throughput
- 100 Gbps network bandwidth (C5n.16xlarge)

Intel AVX-512:

- Twice the floating-point performance of AVX2
- 512-bit instructions (vs 256 for AVX/AVX2)

The latest generation of Intel Xeon processors is the Intel Xeon Scalable Processor Family. This group provides substantial performance improvement over the prior generation, with up to 28 cores delivering enhanced per core performance, and significant increases in memory bandwidth (6 memory channels) and I/O bandwidth and throughput (48 PCIe lanes), your most data-hungry, latency-sensitive applications such as in-memory databases and high-performance computing will see notable improvements enabled by denser compute and faster access to large data volumes.

This family also includes the latest version of Intel's AVX instructions, which double the floating point performance of processors using AVX2.

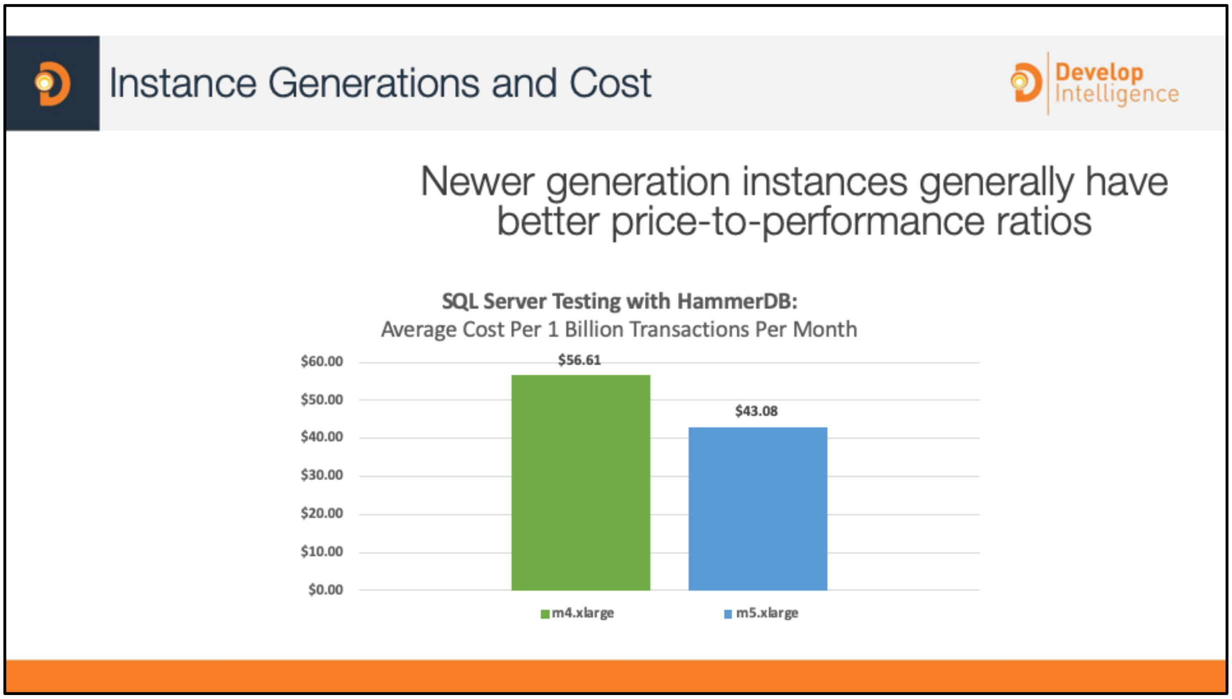


Intel® Xeon Family and EC2 Instances

Intel Xeon Scalable Processor Family	Intel Xeon E5 Processor Family	Intel Xeon E7 Processor Family
• High Memory • z1d • C5/C5n • M5 • R5 • T3	• M4 • R4 • P2/P3 • G3 • F1 • H1 • I3 • D2	• X1/X1e

Here are the most current (as of March 2019) EC2 instance types and their associated processor families. For the latest list of EC2 instance types, see the AWS instance type information page:

https://aws.amazon.com/ec2/instance-types/



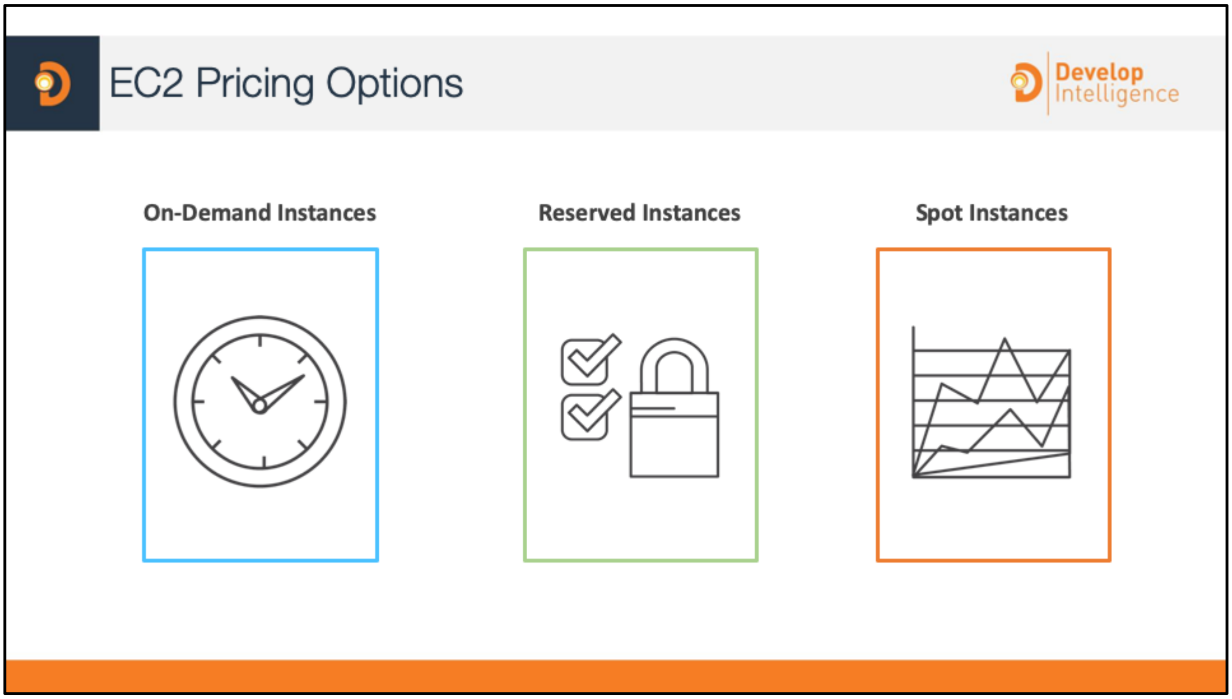
Instance Generations and Cost

Develop Intelligence

Newer generation instances generally have better price-to-performance ratios

Even though newer generation instances cost more than prior generation instances, the price to performance ratio of newer instances is typically higher than older generations. Here's one example, of a test performed using HammerDB on several SQL Server deployments on m4.xlarge and m5.xlarge instances. In this test, they compared the number of transactions that could be performed per month by each instance based on a set number of users (from 3 to 233) with the monthly cost of operating the instances, and averaged all of the results for each instance type. You can read all the details in the link provided below.

Source: https://www.dbbest.com/blog/validating-aws-ec2-sql-server-deployments-using-benchmark-tools/



EC2 Pricing Options

On-Demand Instances

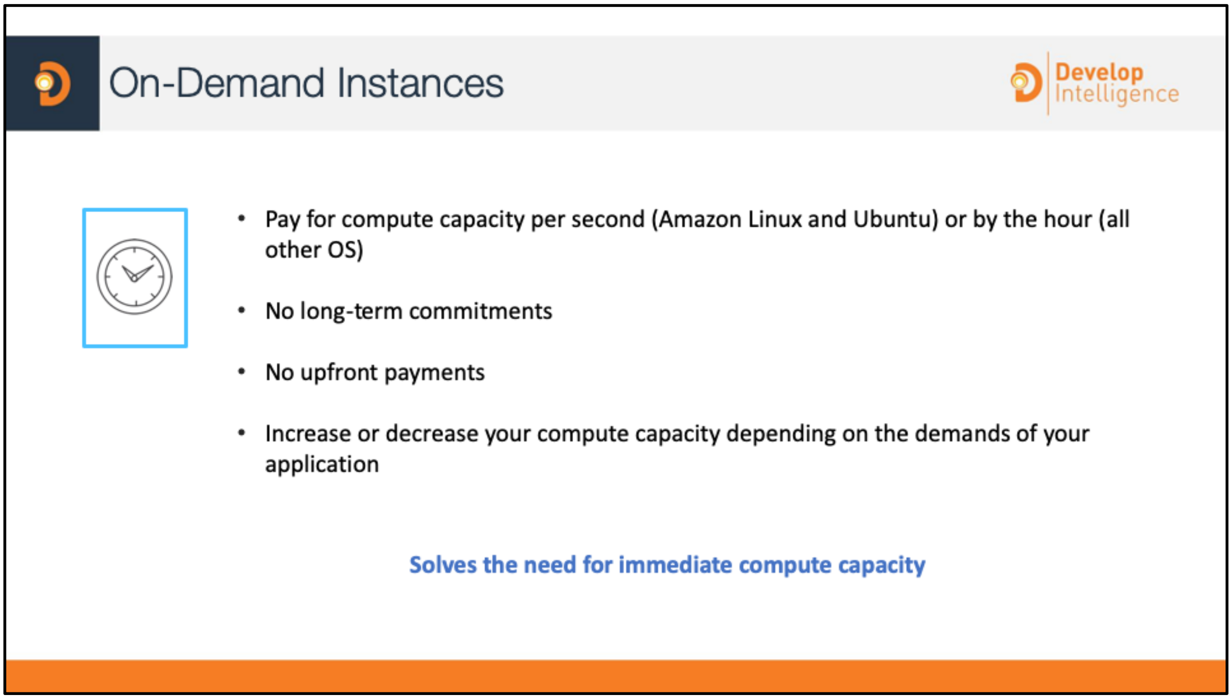
Reserved Instances

Spot Instances

As part of the Free Tier from AWS, new AWS customers can get started with Amazon EC2 t2.micro instances, S3 bucket capacity, and many other AWS service offerings for free for up to one year after sign-up. What's available in the free tier varies from service to service. Please visit https://aws.amazon.com/free/ for details.

Amazon EC2 usage of Amazon Linux- and Ubuntu-based instances that are launched in On-Demand, Reserved and Spot form will be billed on one-second increments, with a minimum of 60 seconds. All other operating systems are billed in one-hour increments, and are billed hour forward, that is, billed at the start of the hour whether you use the full hour or not.

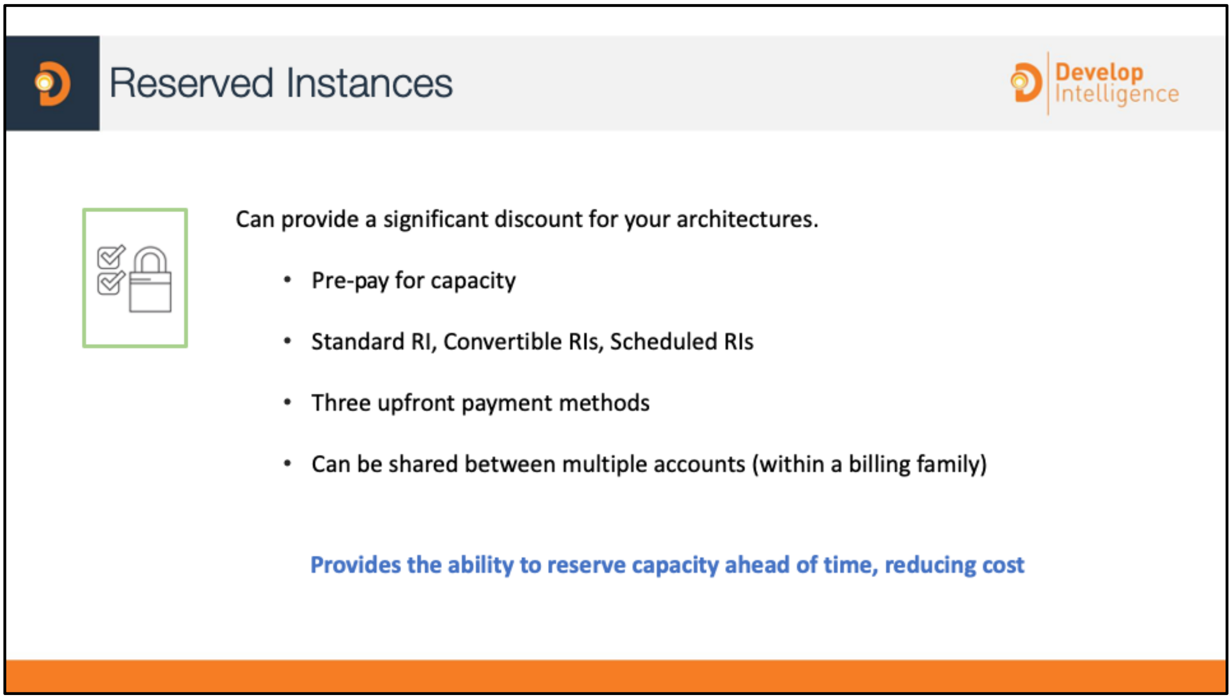
For more information about how AWS pricing works, see https://d0.awsstatic.com/whitepapers/aws_pricing_overview.pdf



On-Demand Instances

- Pay for compute capacity per second (Amazon Linux and Ubuntu) or by the hour (all other OS)
- No long-term commitments
- No upfront payments
- Increase or decrease your compute capacity depending on the demands of your application

Solves the need for immediate compute capacity



Reserved Instances

Develop Intelligence

Can provide a significant discount for your architectures.

- Pre-pay for capacity
- Standard RI, Convertible RIs, Scheduled RIs
- Three upfront payment methods
- Can be shared between multiple accounts (within a billing family)

Provides the ability to reserve capacity ahead of time, reducing cost

Reserve Instances (RI) are a great tool to help reduce cost in your architecture. If you know what the baseline level of usage is going to be for your EC2 instances, an RI can provide significant discounts.

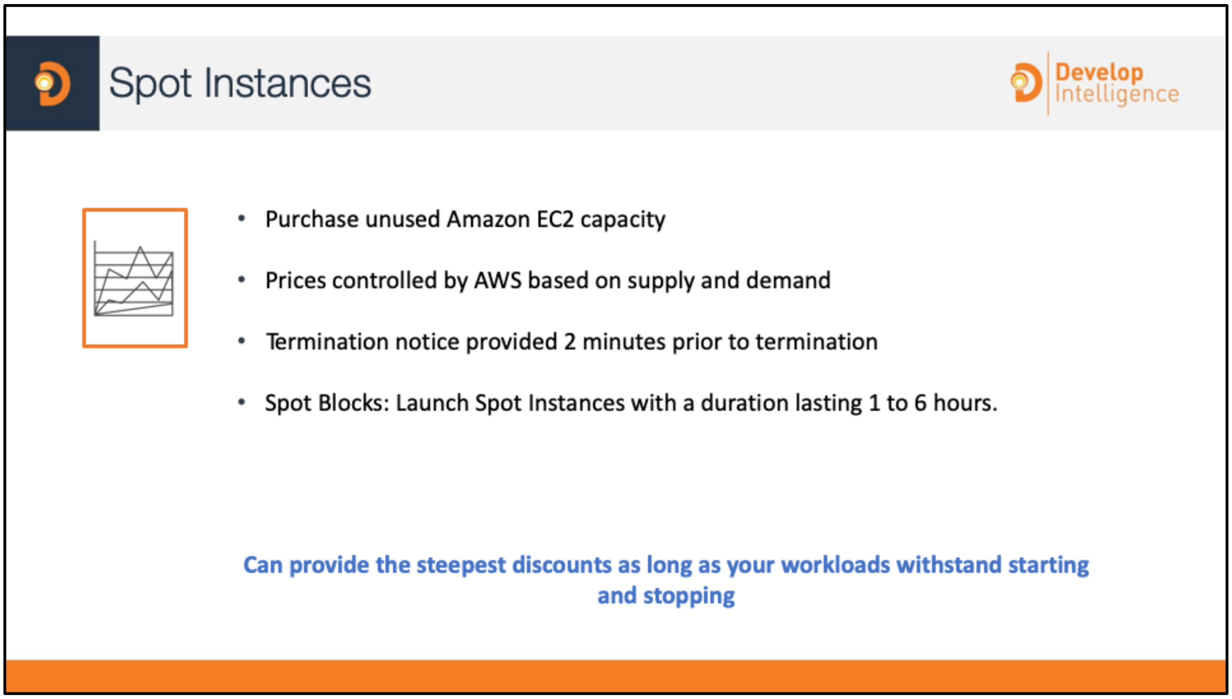
You can set up an RI in multiple ways:

- Standard RIs: Provide the most significant discount (up to 75% off the On-Demand price) and are best suited for ready state usage
- Convertible RIs: Provide a discount (up to 54% off On-Demand price) and are able to change the attributes of the RI as long as the change results in the creation of Ris of equal or greater value
- Schedule RIs: These RIs launch in the time window of your choice, allowing you to match your capacity needs.

Term: AWS offers Standard RIs for 1-year or 3-year terms. Reserved Instance Marketplace sellers also offer RIs with shorter terms. AWS offers Convertible RIs for 1-year or 3-year terms.

Payment option: You can choose between three payment options: All Upfront, Partial Upfront, and No Upfront. If you choose the Partial or No Upfront payment option, the remaining balance will be due in monthly increments over the term.

For more information, see https://docs.aws.amazon.com/aws-technical-content/latest/cost-optimization-reservation-models/introduction.html



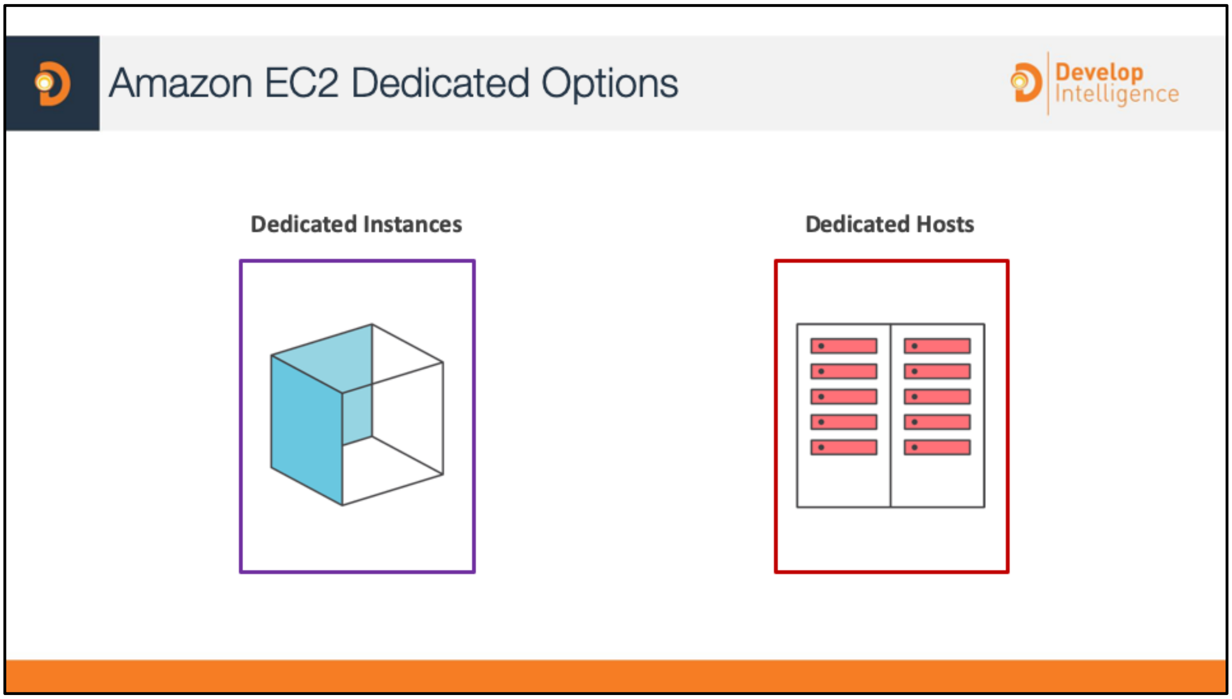
Spot Instances

Develop Intelligence

- Purchase unused Amazon EC2 capacity
- Prices controlled by AWS based on supply and demand
- Termination notice provided 2 minutes prior to termination
- Spot Blocks: Launch Spot Instances with a duration lasting 1 to 6 hours.

Can provide the steepest discounts as long as your workloads withstand starting and stopping

With Amazon EC2 Spot Instances, you don’t have to bid for Spot Instances in the new pricing model, and you just pay the Spot price that’s in effect for the current hour for the instances that you launch. You can request Spot capacity just like you would request On-Demand capacity, without having to spend time analyzing market prices or setting a maximum bid price.



Amazon EC2 Dedicated Options

Develop Intelligence

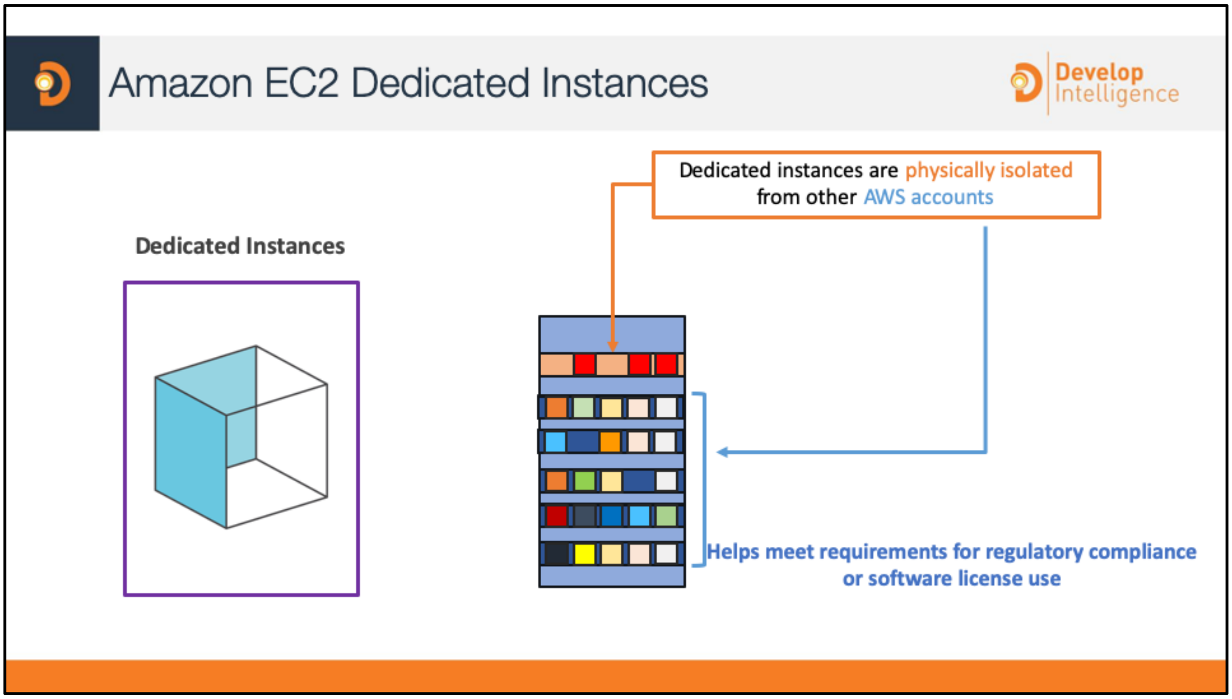
Dedicated Instances

Dedicated Hosts

In addition to these dedicated options, you might want to consider AWS License Manger for your license requirements:

AWS License Manager makes it easier for users to manage licenses in AWS and on-premises servers from various different software vendors (Microsoft, SAP, Oracle, etc...) It will let admins create customized licensing rules when an EC2 instance gets launched, and can use these rules to limit licensing violations such as using more licenses than an agreement allows or being able to reassign licenses to different servers on a short-term basis.

Admins gain control and visibility of all their licenses with the AWS License Manager dashboard.
https://aws.amazon.com/about-aws/whats-new/2018/11/announcing%20aws%20license%20manager/



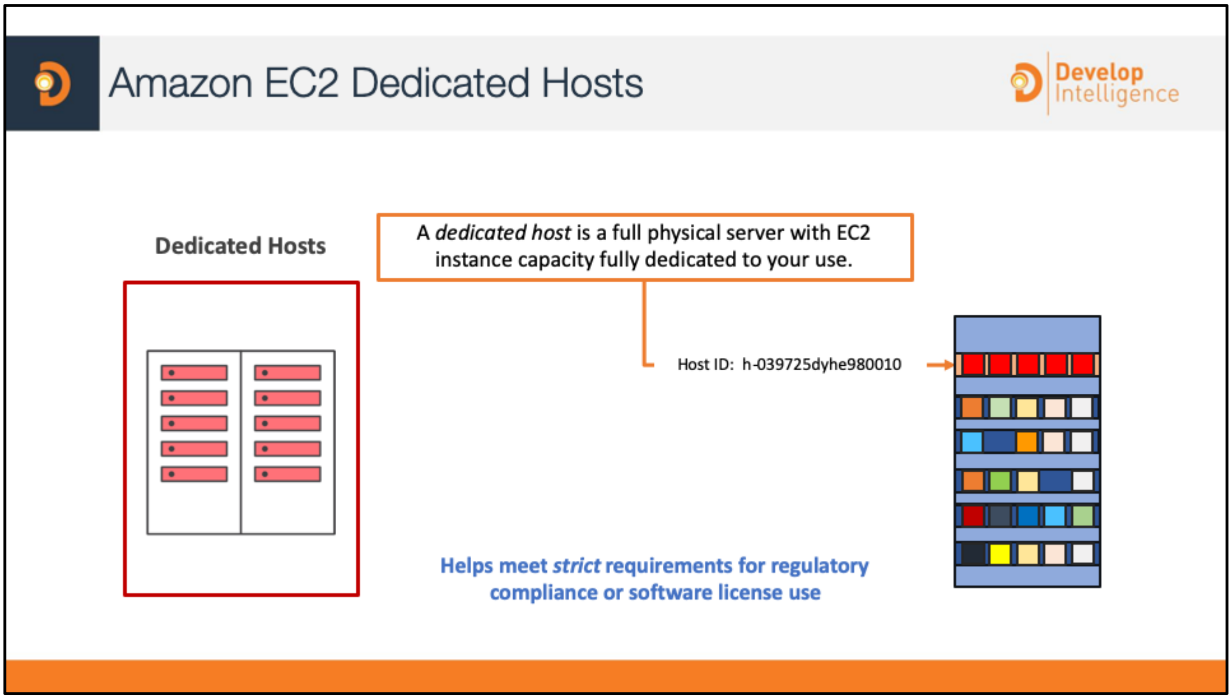
Amazon EC2 Dedicated Instances

Dedicated instances are physically isolated from other AWS accounts

Helps meet requirements for regulatory compliance or software license use

Dedicated Instances are Amazon EC2 instances that run in a VPC on hardware that's dedicated to a single customer. Your Dedicated Instances are physically isolated at the host hardware level from instances that belong to other AWS accounts. Dedicated Instance pricing has two components:

- An hourly per instance usage fee
- A dedicated per-region fee (note that you pay this once per hour, regardless of how many Dedicated Instances you're running)



Amazon EC2 Dedicated Hosts

Develop Intelligence

Dedicated Hosts

A *dedicated host* is a full physical server with EC2 instance capacity fully dedicated to your use.

Host ID: h-039725dyhe980010

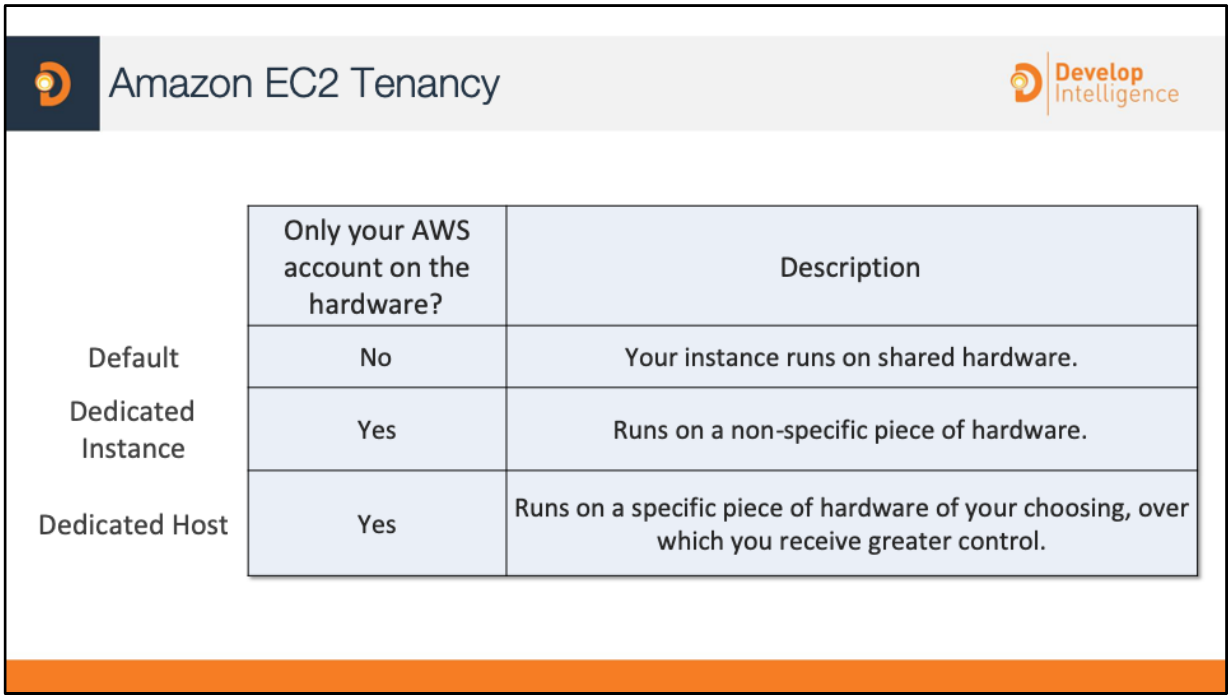
Helps meet *strict* requirements for regulatory compliance or software license use

A Dedicated Host is a physical EC2 server with instance capacity fully dedicated for your use. Dedicated Hosts can help you reduce costs by allowing you to use your existing server-bound software licenses, including Windows Server, SQL Server, and SUSE Linux Enterprise Server (subject to your license terms), and can also help you meet compliance requirements. Dedicated Hosts can be purchased On-Demand (hourly). Reservations can provide up to a 70% discount compared to the On-Demand price.

Dedicated Host benefits:

- **Save money on licensing costs**: Dedicated Hosts can enable you to save money by using your own per-socket or per-core software licenses in Amazon EC2.
- **Help meet compliance and regulatory requirements**: Dedicated Hosts allow you to place your instances in a VPC on a specific, physical server. This enables you to deploy instances using configurations that help address corporate compliance and regulatory requirements

For more information about Dedicated Hosts, see https://aws.amazon.com/ec2/dedicated-hosts/



Amazon EC2 Tenancy

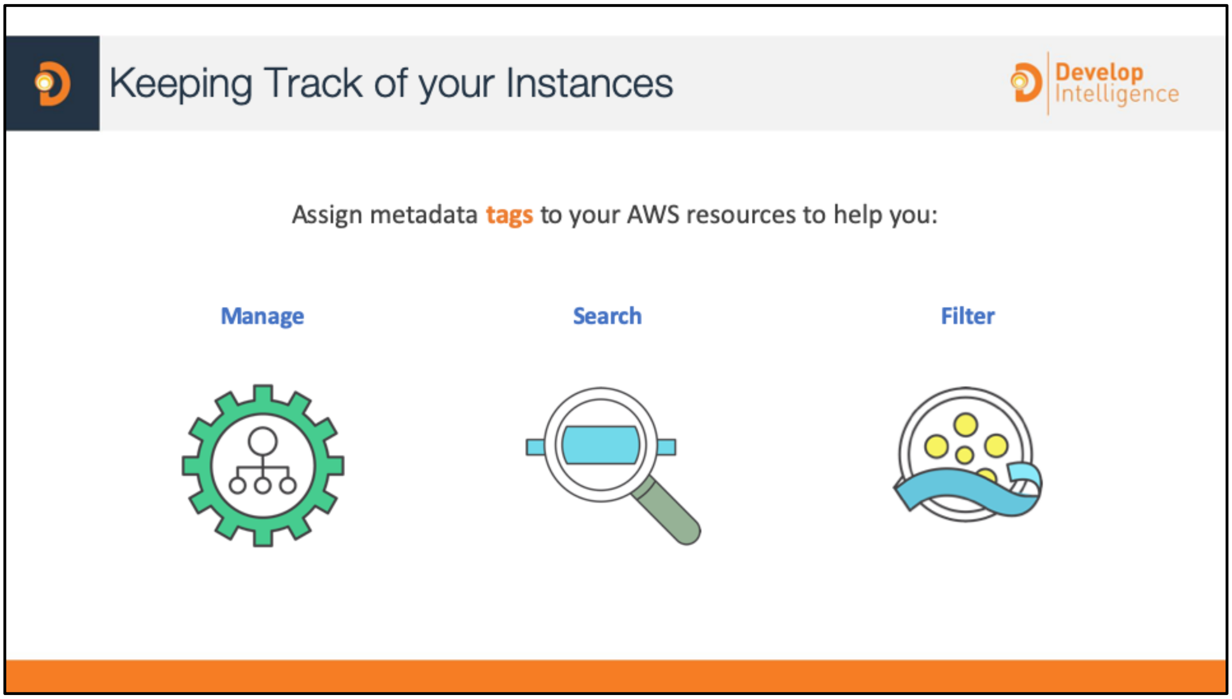
Develop Intelligence

	Only your AWS account on the hardware?	Description
Default	No	Your instance runs on shared hardware.
Dedicated Instance	Yes	Runs on a non-specific piece of hardware.
Dedicated Host	Yes	Runs on a specific piece of hardware of your choosing, over which you receive greater control.

After you launch an instance, there are some limitations to changing its tenancy.

- You cannot change the tenancy of an instance from default to dedicated or host after you've launched it.
- You cannot change the tenancy of an instance from dedicated or host to default after you've launched it.
- You *can* change the tenancy of an instance from dedicated to host, or from host to dedicated, after you've launched it.

For more information, see Changing the Tenancy of an Instance.



Keeping Track of your Instances

Develop Intelligence

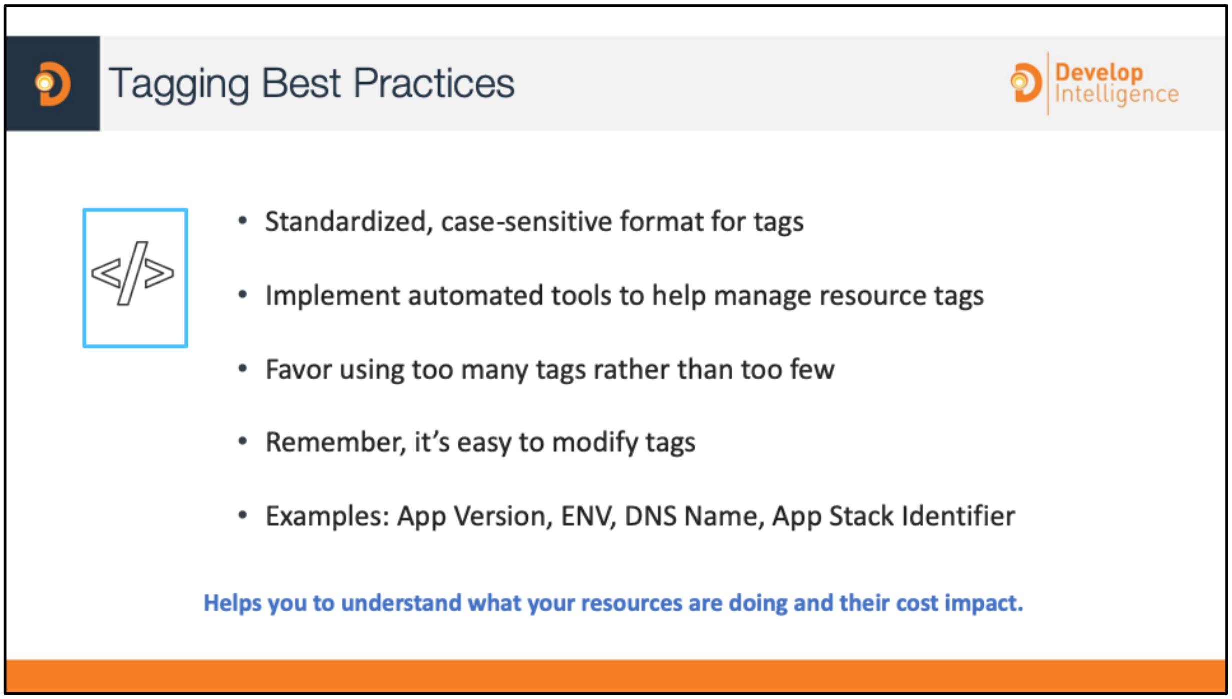
Assign metadata **tags** to your AWS resources to help you:

Manage **Search** **Filter**

AWS allows customers to assign metadata to their AWS resources in the form of *tags*. Each tag is a simple label consisting of a customer-defined key and an optional value that can make it easier to manage, search for, and filter resources.

Although there are no inherent types of tags, they enable customers to categorize resources by purpose, owner, environment, or other criteria. This webpage describes commonly used tagging categories and strategies to help AWS customers implement a consistent and effective tagging strategy. The following sections assume basic knowledge of AWS resources, tagging, detailed billing, and IAM.

For more information about AWS tagging strategies, see https://aws.amazon.com/answers/account-management/aws-tagging-strategies/.



Tagging Best Practices

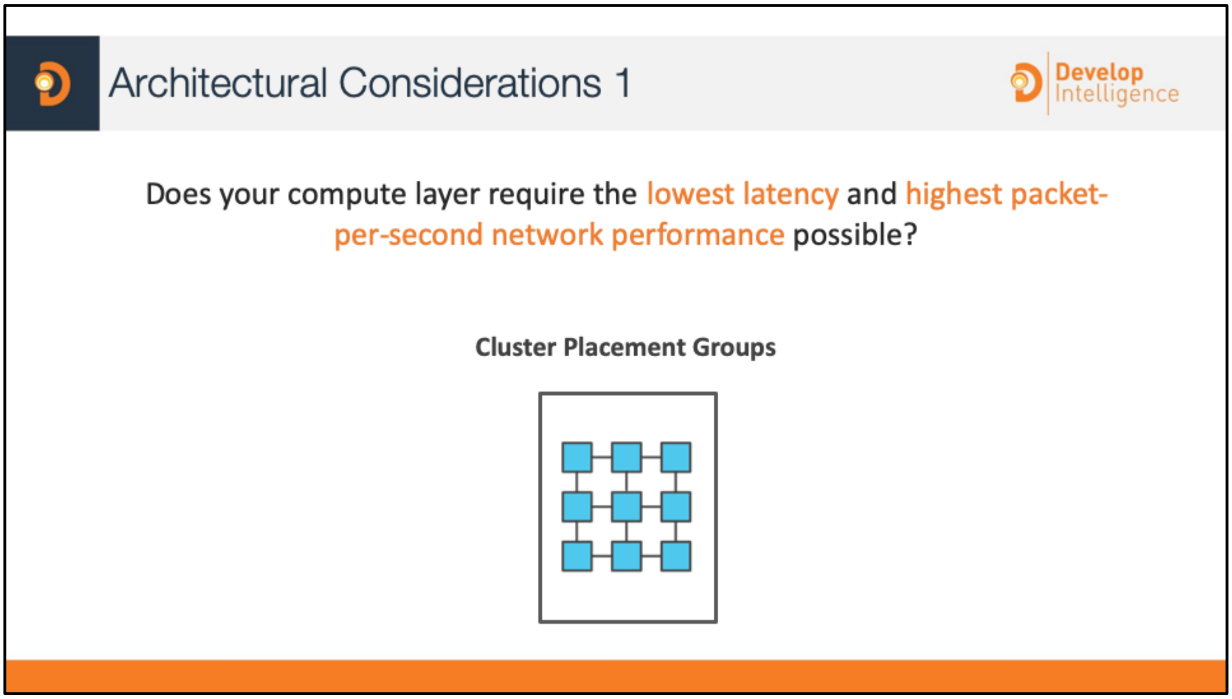
- Standardized, case-sensitive format for tags
- Implement automated tools to help manage resource tags
- Favor using too many tags rather than too few
- Remember, it's easy to modify tags
- Examples: App Version, ENV, DNS Name, App Stack Identifier

Helps you to understand what your resources are doing and their cost impact.

Always use a standardized, case-sensitive format for tags, and implement it consistently across all resource types.
Consider tag dimensions that support the ability to manage resource access control, cost tracking, automation, and organization.

Implement automated tools to help manage resource tags. The Resource Groups Tagging API enables programmatic control of tags, making it easier to automatically manage, search, and filter tags and resources. It also simplifies backups of tag data across all supported services with a single API call per AWS Region.
Err on the side of using too many tags rather than too few tags.

Remember that it is easy to modify tags to accommodate changing business requirements, but make sure to consider the ramifications of future changes, especially in relation to tag-based access control, automation, or upstream billing reports.



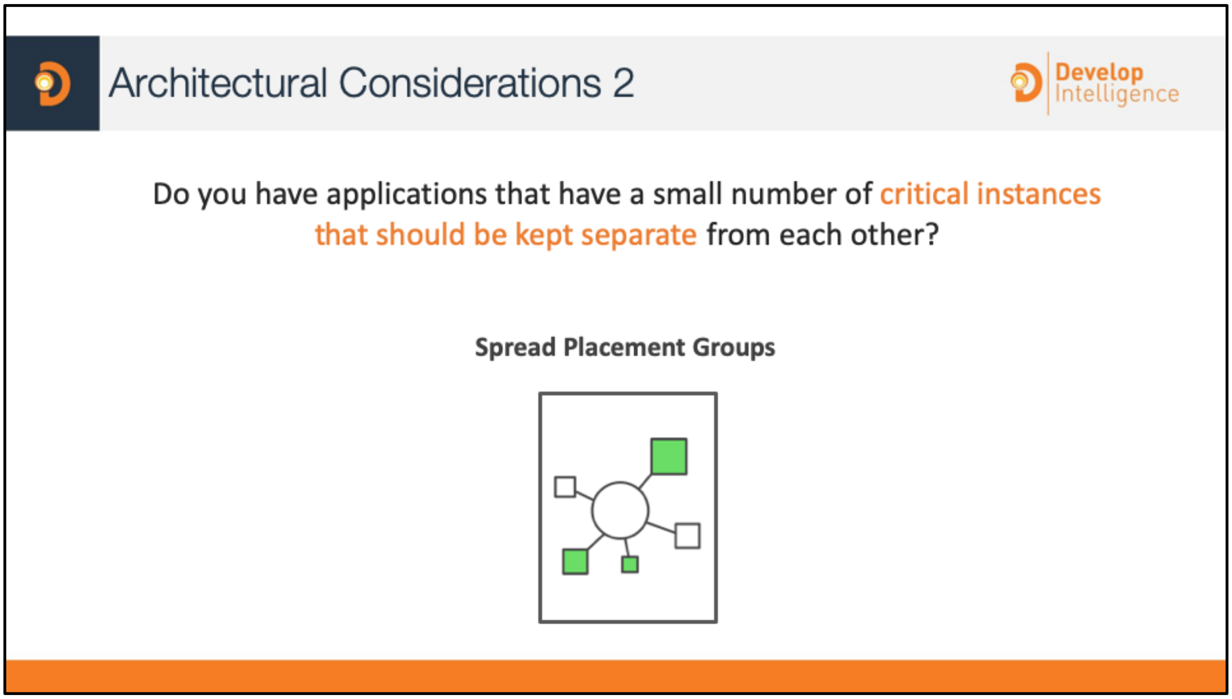
Architectural Considerations 1

Does your compute layer require the lowest latency and highest packet-per-second network performance possible?

Cluster Placement Groups

The cluster placement group is a logical grouping of instances within a single Availability Zone. This grouping provides the lowest latency and highest packet per second network performance possible.

We recommend that you launch all the instances you will need in this grouping *at one time*. If you try to add more instances into the group later, you will increase your chance of receiving an insufficient capacity error.



Architectural Considerations 2

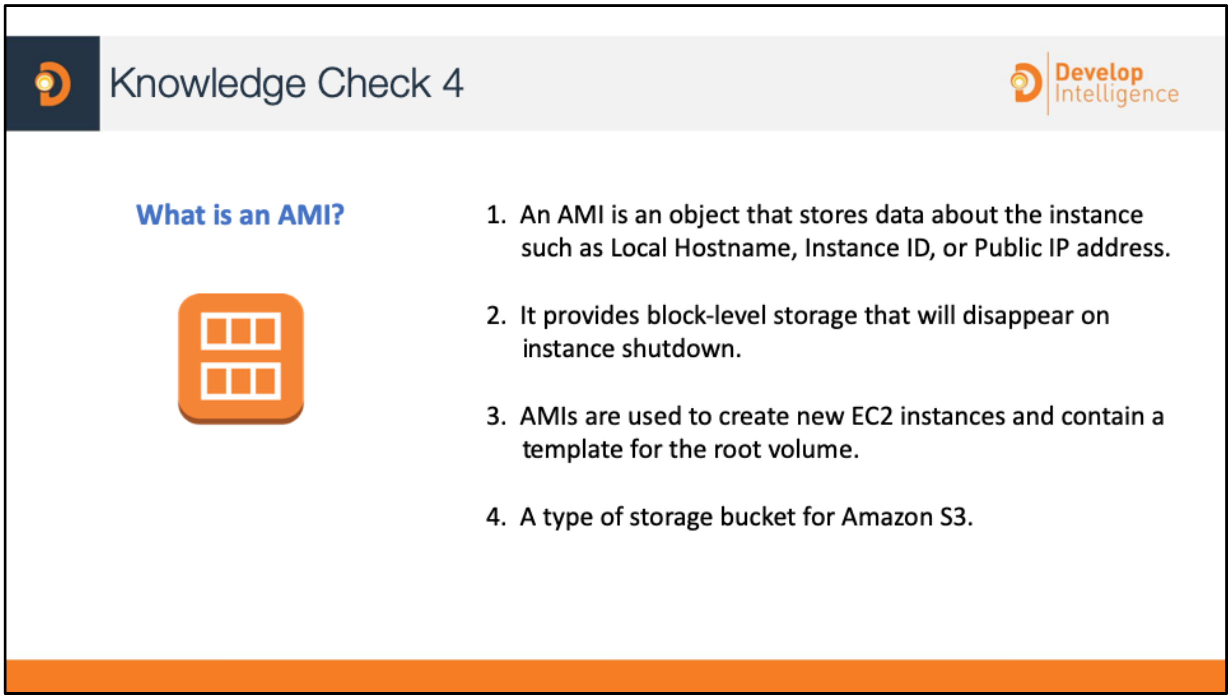
Develop Intelligence

Do you have applications that have a small number of critical instances that should be kept separate from each other?

Spread Placement Groups

A *spread placement group* is a grouping of instances that are purposely positioned on distinct underlying hardware. This grouping reduces the risk of simultaneous failures that could occur if instances where sharing underlying hardware.

This type of group can span multiple Availability Zones, up to a maximum of seven instances per Availability Zone per group.

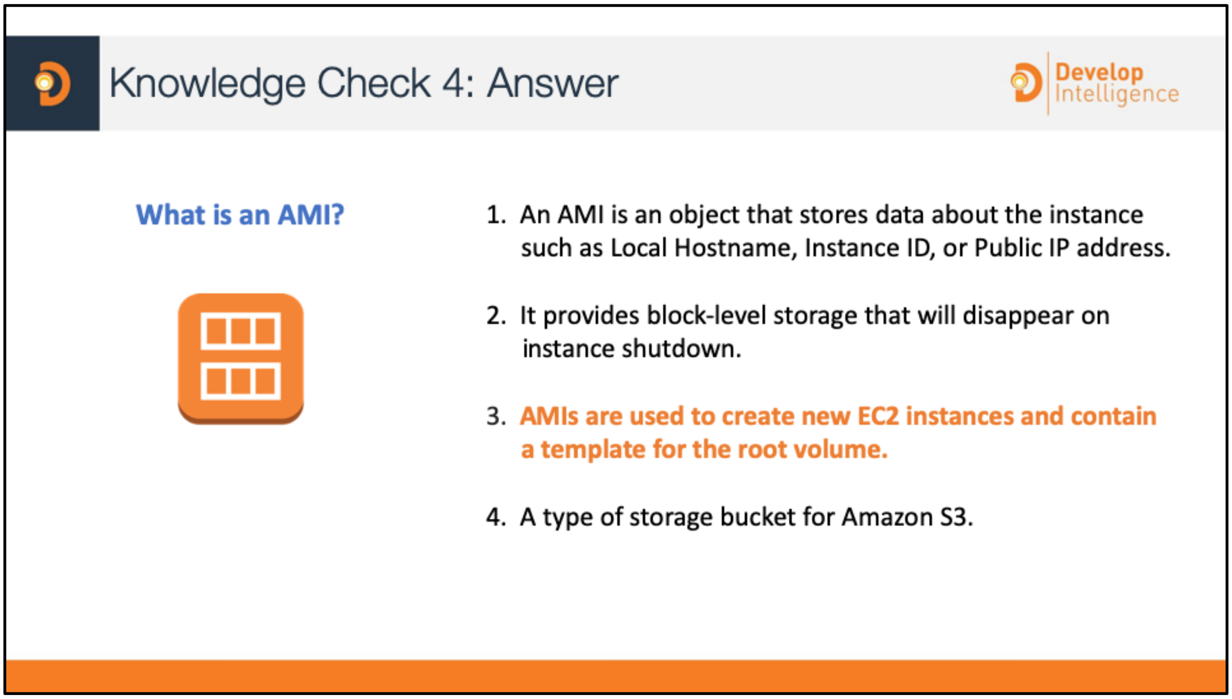


Knowledge Check 4

Develop Intelligence

What is an AMI?

1. An AMI is an object that stores data about the instance such as Local Hostname, Instance ID, or Public IP address.
2. It provides block-level storage that will disappear on instance shutdown.
3. AMIs are used to create new EC2 instances and contain a template for the root volume.
4. A type of storage bucket for Amazon S3.

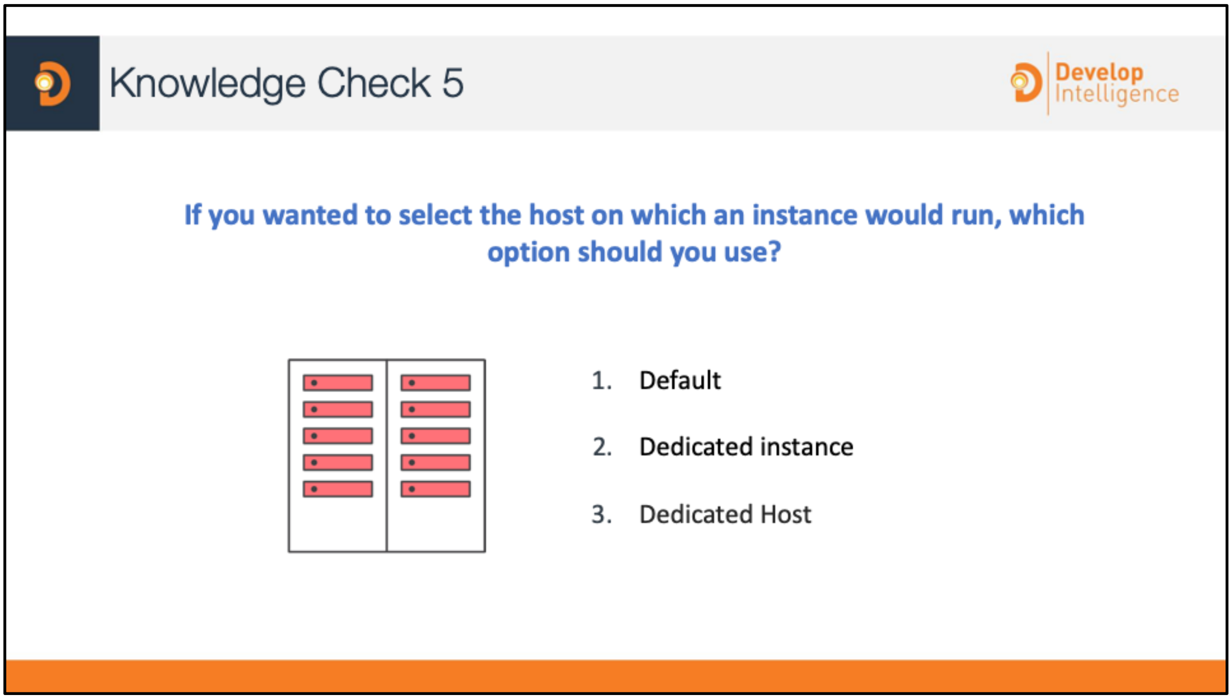


Knowledge Check 4: Answer

Develop Intelligence

What is an AMI?

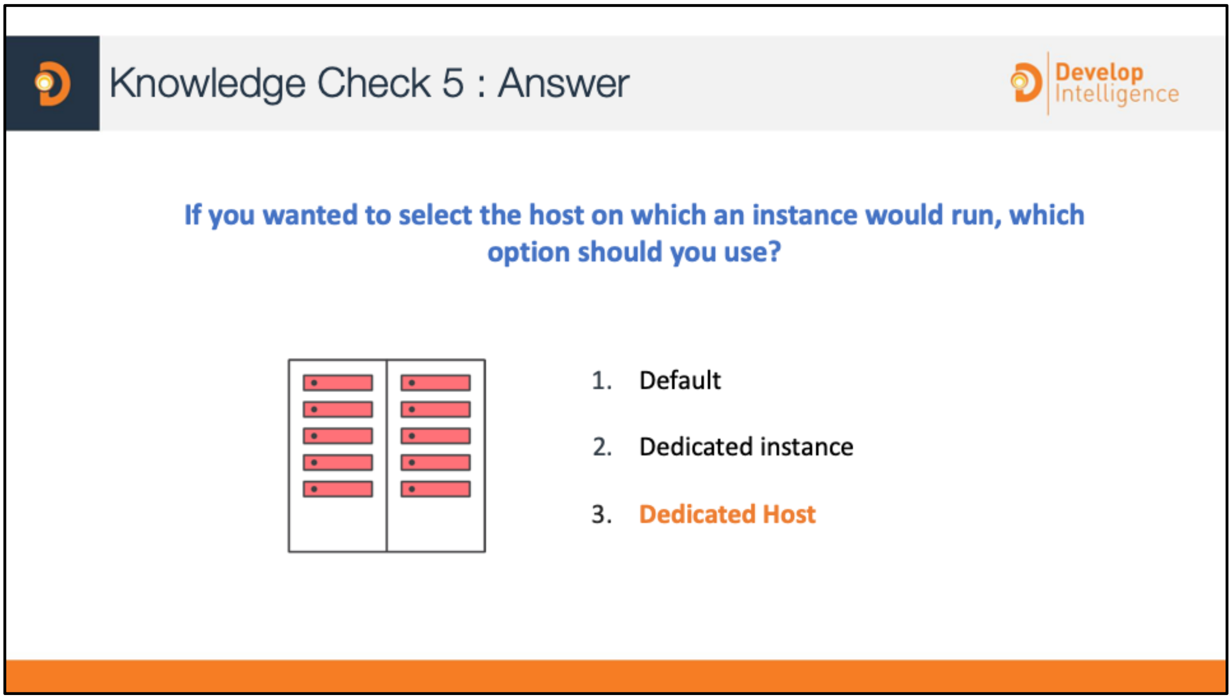
1. An AMI is an object that stores data about the instance such as Local Hostname, Instance ID, or Public IP address.
2. It provides block-level storage that will disappear on instance shutdown.
3. **AMIs are used to create new EC2 instances and contain a template for the root volume.**
4. A type of storage bucket for Amazon S3.



Knowledge Check 5

If you wanted to select the host on which an instance would run, which option should you use?

1. Default
2. Dedicated instance
3. Dedicated Host

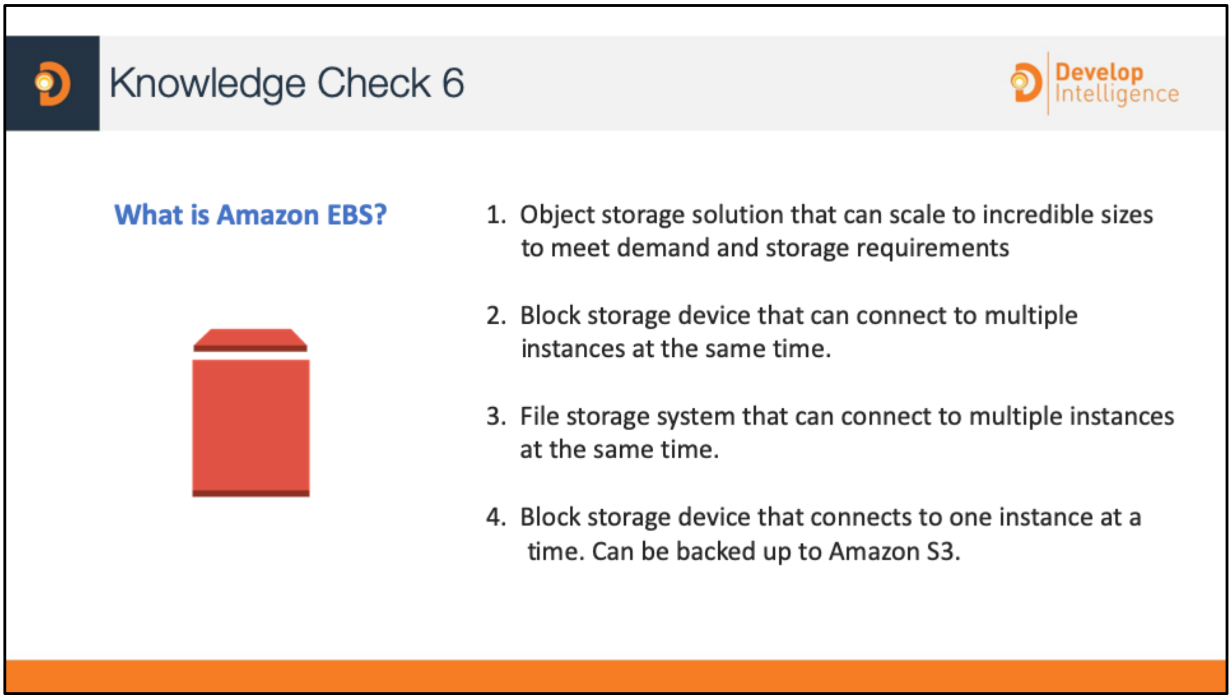


Knowledge Check 5 : Answer

Develop Intelligence

If you wanted to select the host on which an instance would run, which option should you use?

1. Default
2. Dedicated instance
3. **Dedicated Host**

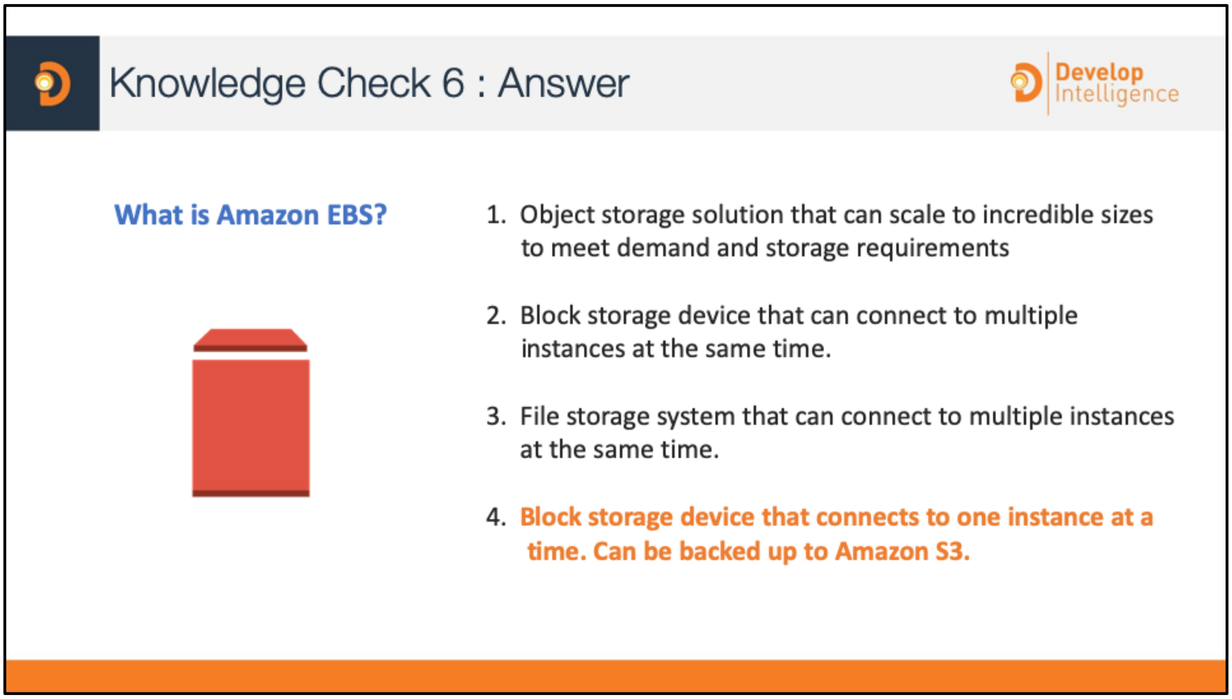


Knowledge Check 6

Develop Intelligence

What is Amazon EBS?

1. Object storage solution that can scale to incredible sizes to meet demand and storage requirements
2. Block storage device that can connect to multiple instances at the same time.
3. File storage system that can connect to multiple instances at the same time.
4. Block storage device that connects to one instance at a time. Can be backed up to Amazon S3.



Knowledge Check 6 : Answer

Develop Intelligence

What is Amazon EBS?

1. Object storage solution that can scale to incredible sizes to meet demand and storage requirements
2. Block storage device that can connect to multiple instances at the same time.
3. File storage system that can connect to multiple instances at the same time.
4. **Block storage device that connects to one instance at a time. Can be backed up to Amazon S3.**