Masterclass: Amazon EC2

Ian Massingham

Chief Evangelist (EMEA)

Amazon Web Services

@IanMmmm

Adam Boeglin

Solutions Architect

Amazon Web Services

27 July 2016

amazon web services | Webinars

Masterclass

1. A technical deep dive that goes beyond the basics

2. Intended to educate you on how to get the best from AWS services

3. Show you how things work and how to get things done

Amazon EC2

- Provides resizable compute capacity in the Cloud
- Designed to make web-scale cloud computing easier
- A true virtual computing environment
- Launch instances with a variety of operating systems
- Run as many or few systems as you desire

Completely Controlled

Flexible

Elastic

Web-Scale

Amazon EC2

Easy to Start

Reliable

Inexpensive

Secure

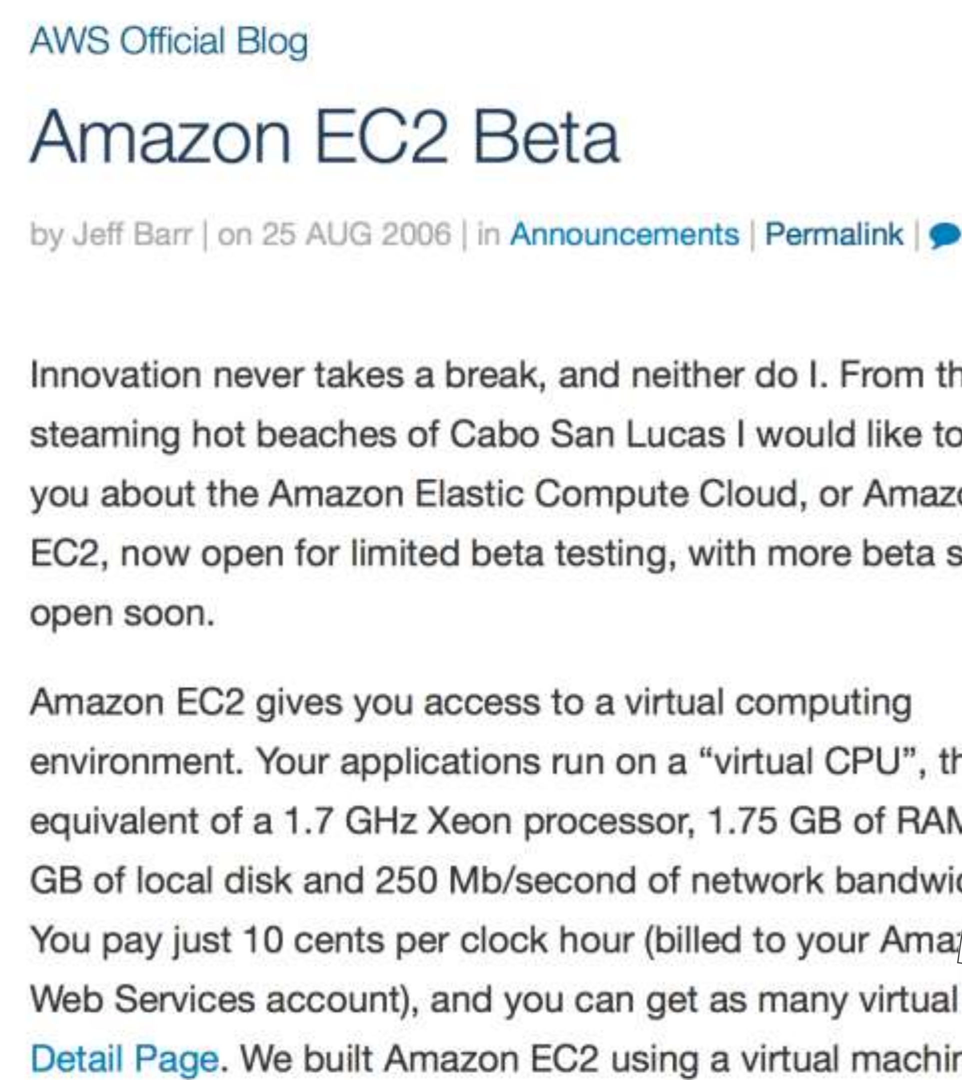
AWS Official Blog

Amazon EC2 Beta

by Jeff Barr | on 25 AUG 2006 | in Announcements | Permalink | 💬 Comments

Innovation never takes a break, and neither do I. From the steaming hot beaches of Cabo San Lucas I would like to tell you about the Amazon Elastic Compute Cloud, or Amazon EC2, now open for limited beta testing, with more beta slots to open soon.

Amazon EC2 gives you access to a virtual computing environment. Your applications run on a "virtual CPU", the equivalent of a 1.7 GHz Xeon processor, 1.75 GB of RAM, 160 GB of local disk and 250 Mb/second of network bandwidth. You pay just 10 cents per clock hour (billed to your Amazon Web Services account), and you can get as many virtual CPUs as you need. You can learn more on the EC2 Detail Page. We built Amazon EC2 using a virtual machine monitor by the name of Xen.



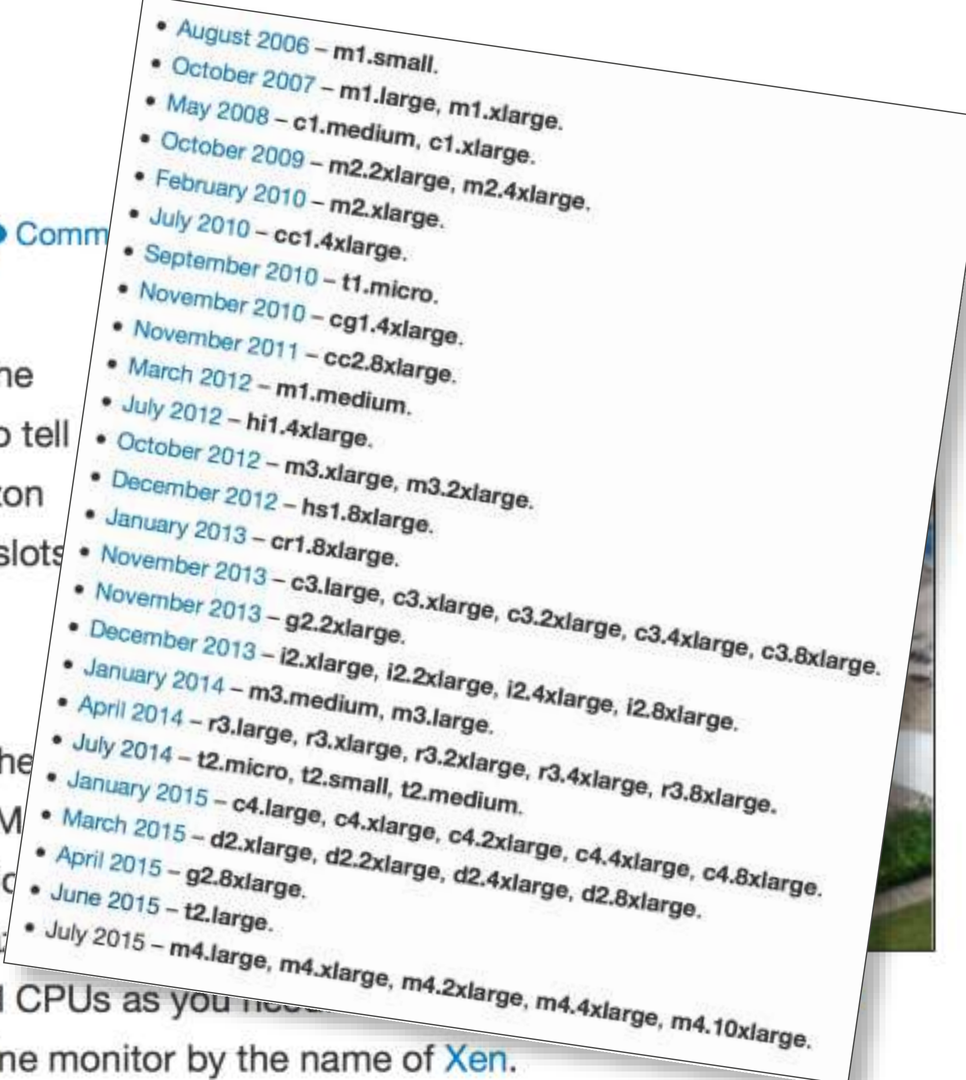
AWS Official Blog

Amazon EC2 Beta

by Jeff Barr | on 25 AUG 2006 | in Announcements | Permalink | Comm

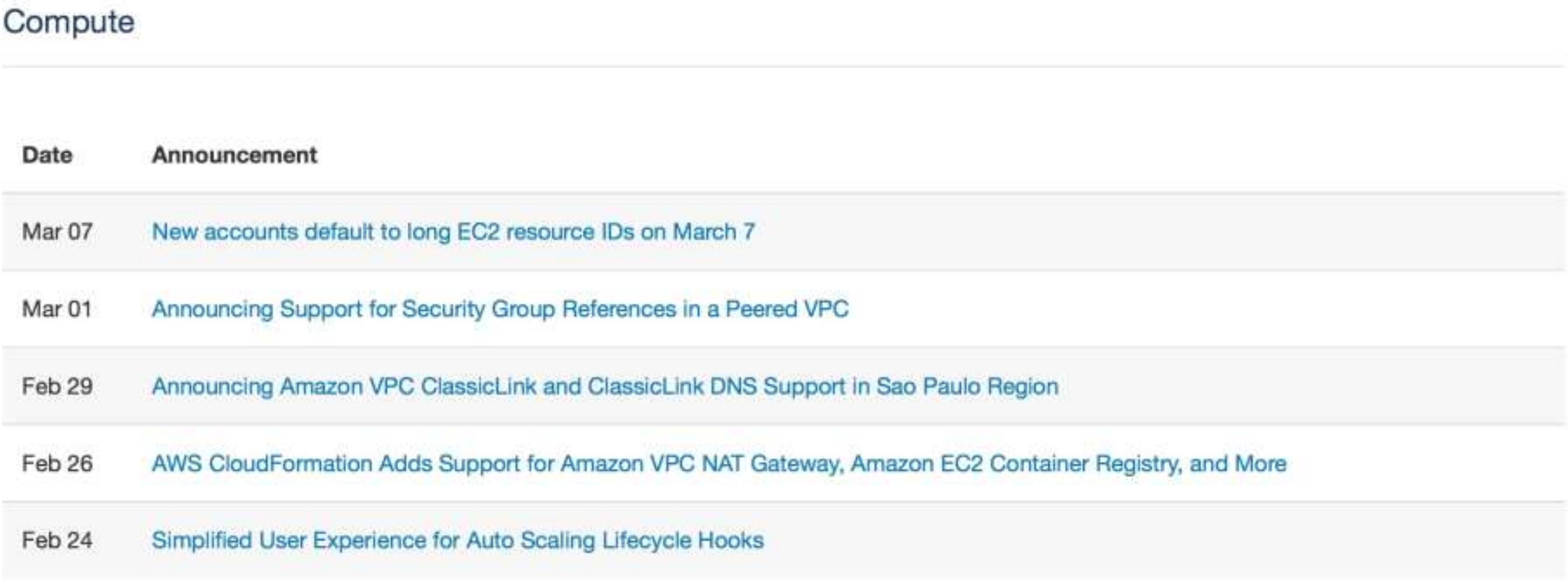
Innovation never takes a break, and neither do I. From the steaming hot beaches of Cabo San Lucas I would like to tell you about the Amazon Elastic Compute Cloud, or Amazon EC2, now open for limited beta testing, with more beta slots open soon.

Amazon EC2 gives you access to a virtual computing environment. Your applications run on a “virtual CPU”, the equivalent of a 1.7 GHz Xeon processor, 1.75 GB of RAM GB of local disk and 250 Mb/second of network bandwid You pay just 10 cents per clock hour (billed to your Ama Web Services account), and you can get as many virtual CPUs as you nee Detail Page. We built Amazon EC2 using a virtual machine monitor by the name of Xen.



- August 2006 – m1.small.
- October 2007 – m1.large, m1.xlarge.
- May 2008 – c1.medium, c1.xlarge.
- October 2009 – m2.2xlarge, m2.4xlarge.
- February 2010 – m2.xlarge.
- July 2010 – cc1.4xlarge.
- September 2010 – t1.micro.
- November 2010 – cg1.4xlarge.
- November 2011 – cc2.8xlarge.
- March 2012 – m1.medium.
- July 2012 – hi1.4xlarge.
- October 2012 – m3.xlarge, m3.2xlarge.
- December 2012 – hs1.8xlarge.
- January 2013 – cr1.8xlarge.
- November 2013 – c3.large, c3.xlarge, c3.2xlarge, c3.4xlarge, c3.8xlarge.
- November 2013 – g2.2xlarge.
- December 2013 – i2.xlarge, i2.2xlarge, i2.4xlarge, i2.8xlarge.
- January 2014 – m3.medium, m3.large.
- April 2014 – r3.large, r3.xlarge, r3.2xlarge, r3.4xlarge, r3.8xlarge.
- July 2014 – t2.micro, t2.small, t2.medium.
- January 2015 – c4.large, c4.xlarge, c4.2xlarge, c4.4xlarge, c4.8xlarge.
- March 2015 – d2.xlarge, d2.2xlarge, d2.4xlarge, d2.8xlarge.
- April 2015 – g2.8xlarge.
- June 2015 – t2.large.
- July 2015 – m4.large, m4.xlarge, m4.2xlarge, m4.4xlarge, m4.10xlarge.

New Amazon EC2/Compute Services Features



Compute

Date	Announcement
Mar 07	New accounts default to long EC2 resource IDs on March 7
Mar 01	Announcing Support for Security Group References in a Peered VPC
Feb 29	Announcing Amazon VPC ClassicLink and ClassicLink DNS Support in Sao Paulo Region
Feb 26	AWS CloudFormation Adds Support for Amazon VPC NAT Gateway, Amazon EC2 Container Registry, and More
Feb 24	Simplified User Experience for Auto Scaling Lifecycle Hooks

Agenda

- Amazon EC2 Concepts & Fundamentals
- Storage & Networking
- Monitoring, Metrics & Logs

Security & Access Control

- Deployment
- Cost Optimization

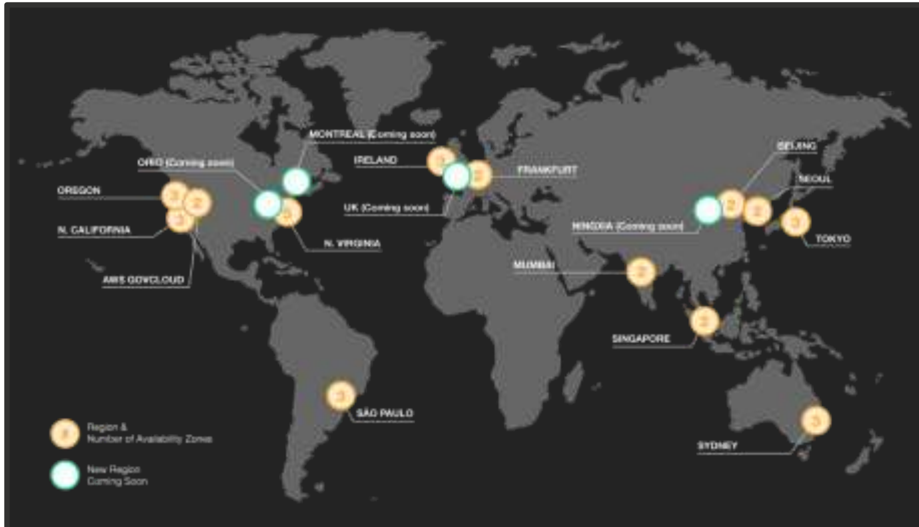
Amazon EC2 Concepts

Regions

- The geographical area where Amazon EC2 will launch the instances that you create

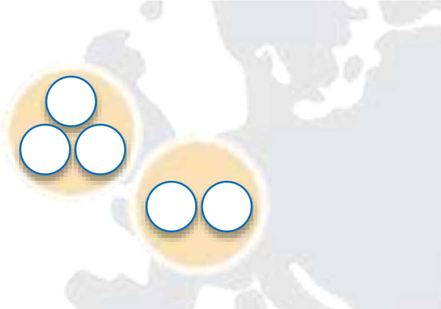
- Choose a region to optimize latency, minimize costs, or address regulatory requirements

- 13 regions around the world



Availability Zones

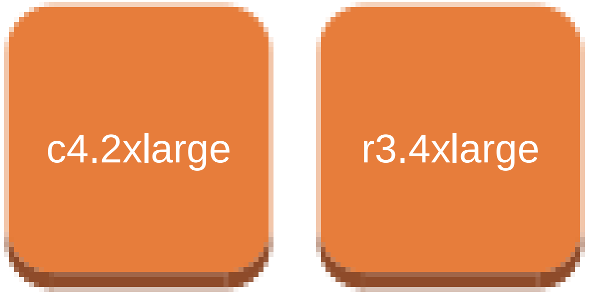
- Distinct locations that are engineered to be insulated from failures in other Availability Zones
- Provide inexpensive, low latency network connectivity to other Availability Zones in the same Region
- Regions contain between 2 & 5 EC2 availability zones



Instances

- Amazon EC2 provides a wide selection of instance types optimized to fit different use cases

- Instance types comprise varying combinations of CPU, memory, storage, and networking capacity



Extensive list of supported operating systems & software

Available in different locations

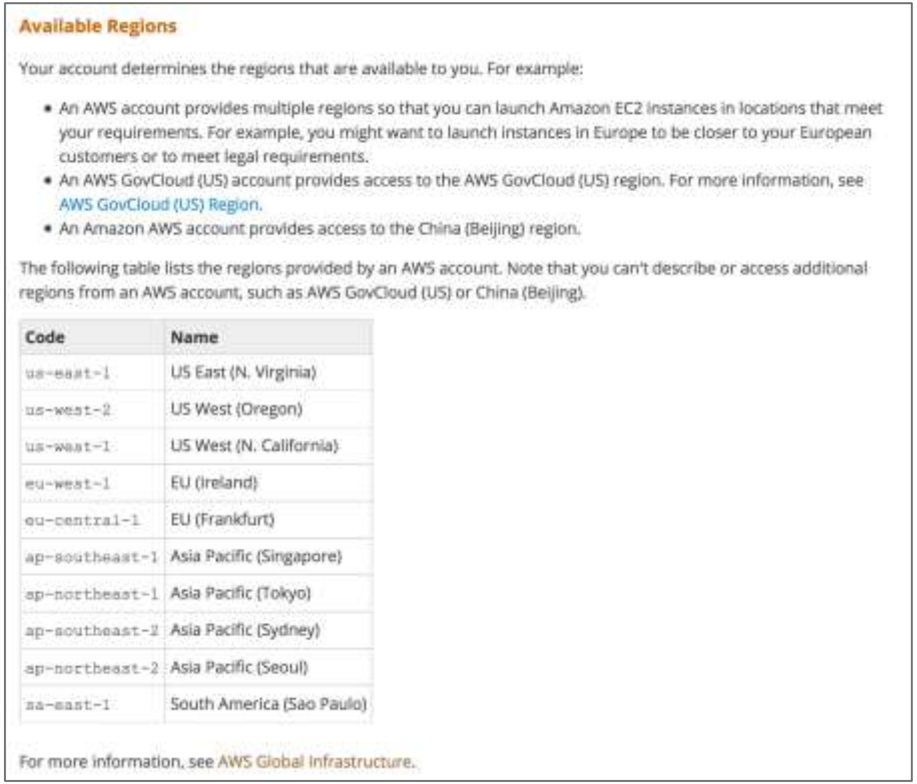
Integrated with other AWS Services

Amazon EC2 Fundamentals

Choice of instance families with differing resource ratios

Purchasing options for cost optimization

Import and export virtual machines



Available Regions

Your account determines the regions that are available to you. For example:

- An AWS account provides multiple regions so that you can launch Amazon EC2 instances in locations that meet your requirements. For example, you might want to launch instances in Europe to be closer to your European customers or to meet legal requirements.
- An AWS GovCloud (US) account provides access to the AWS GovCloud (US) region. For more information, see AWS GovCloud (US) Region.
- An Amazon AWS account provides access to the China (Beijing) region.

The following table lists the regions provided by an AWS account. Note that you can't describe or access additional regions from an AWS account, such as AWS GovCloud (US) or China (Beijing).

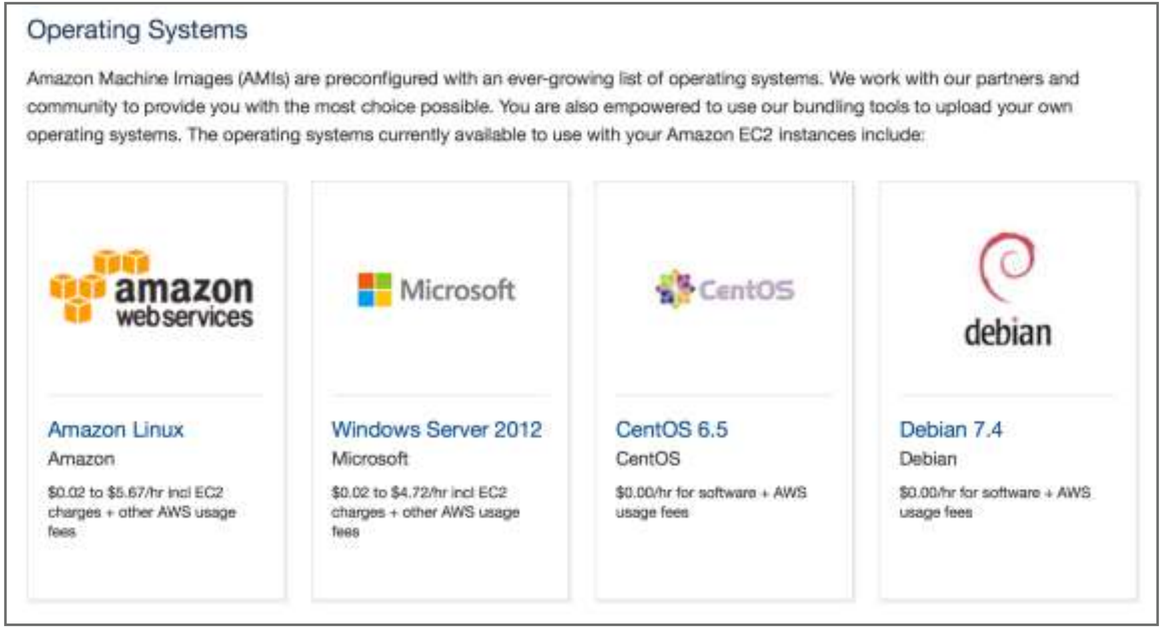
Code	Name
us-east-1	US East (N. Virginia)
us-west-2	US West (Oregon)
us-west-1	US West (N. California)
eu-west-1	EU (Ireland)
eu-central-1	EU (Frankfurt)
ap-southeast-1	Asia Pacific (Singapore)
ap-northeast-1	Asia Pacific (Tokyo)
ap-southeast-2	Asia Pacific (Sydney)
ap-northeast-2	Asia Pacific (Seoul)
sa-east-1	South America (Sao Paulo)

For more information, see AWS Global Infrastructure.

Available in Different Locations

docs.aws.amazon.com/AWSEC2/latest/UserGuide/using-regions-availability-zones.html

Extensive list of supported operating systems & software



Operating Systems

Amazon Machine Images (AMIs) are preconfigured with an ever-growing list of operating systems. We work with our partners and community to provide you with the most choice possible. You are also empowered to use our bundling tools to upload your own operating systems. The operating systems currently available to use with your Amazon EC2 instances include:

Amazon Linux
Amazon
$0.02 to $5.67/hr incl EC2 charges + other AWS usage fees

Windows Server 2012
Microsoft
$0.02 to $4.72/hr incl EC2 charges + other AWS usage fees

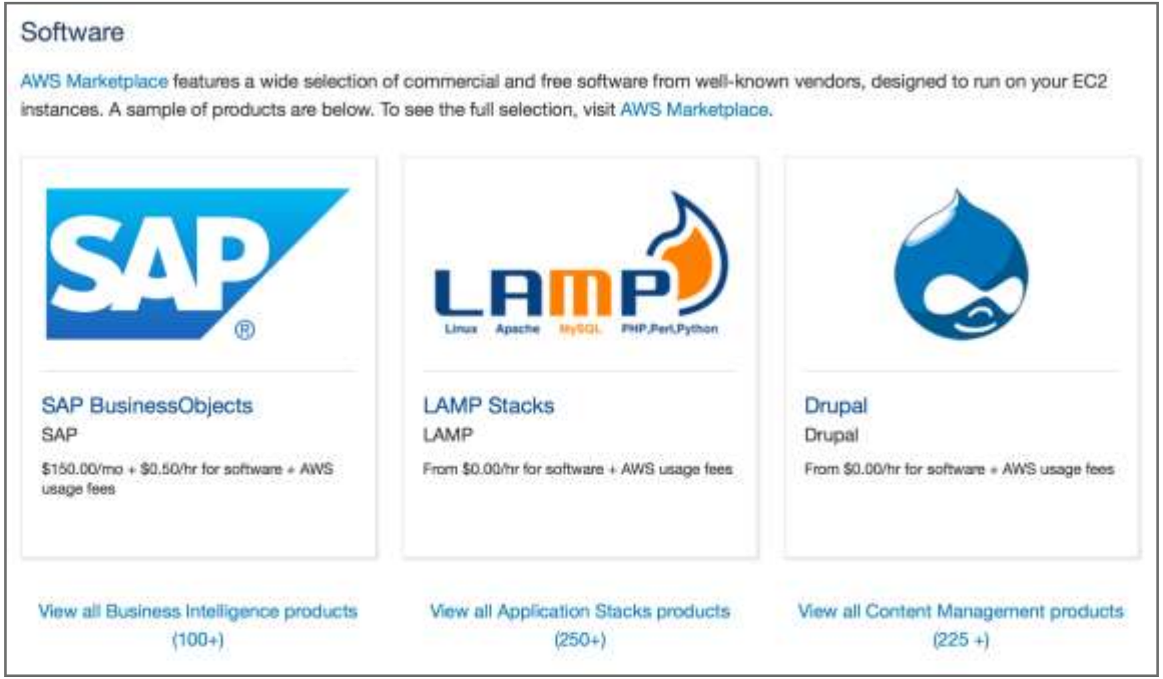
CentOS 6.5
CentOS
$0.00/hr for software + AWS usage fees

Debian 7.4
Debian
$0.00/hr for software + AWS usage fees

RedHat Linux, Windows Server, SuSE Linux, Ubuntu, Fedora, Debian, Cent OS, Gentoo Linux, Oracle Linux, and FreeBSD

aws.amazon.com/ec2/faqs/

Extensive list of supported operating systems & software



Software

AWS Marketplace features a wide selection of commercial and free software from well-known vendors, designed to run on your EC2 instances. A sample of products are below. To see the full selection, visit AWS Marketplace.

SAP BusinessObjects
SAP
$150.00/mo + $0.50/hr for software + AWS usage fees

LAMP Stacks
LAMP
From $0.00/hr for software + AWS usage fees

Drupal
Drupal
From $0.00/hr for software + AWS usage fees

View all Business Intelligence products (100+)

View all Application Stacks products (250+)

View all Content Management products (225 +)

aws.amazon.com/marketplace

Integrated with other AWS Services

- Amazon Elastic Block Store
- Amazon CloudWatch
- Amazon Virtual Private Cloud
- AWS Identity and Access Management

Purchasing options for cost optimisation

On-Demand Instances

Pay for compute capacity by the hour with no long-term commitments or upfront payments

Reserved Instances

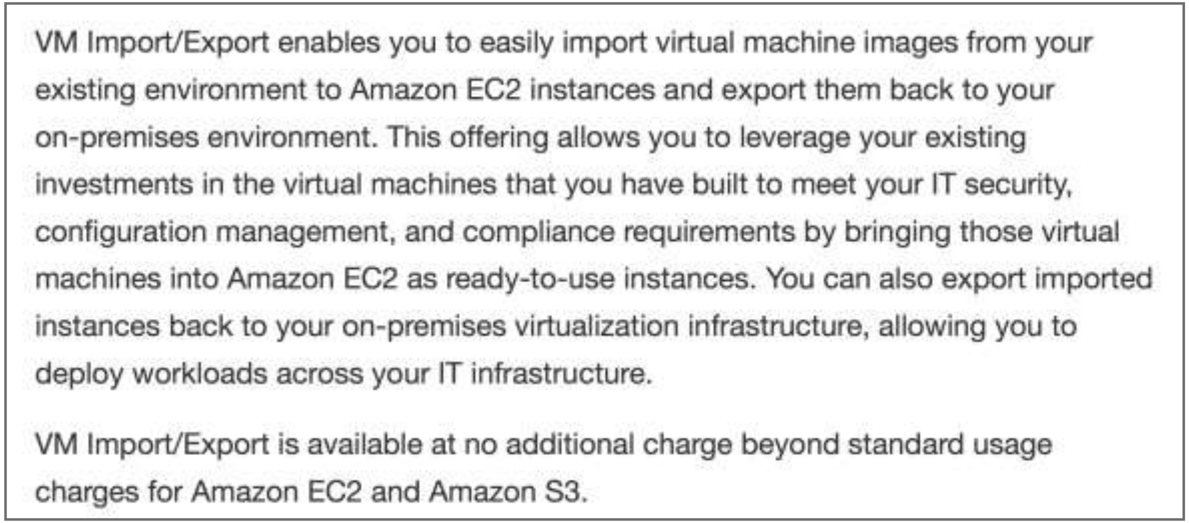
Provide you with a significant discount (up to 75%) compared to On-Demand Instance pricing

Spot Instances

Purchase compute capacity with no upfront commitment and at hourly rates usually lower than the On-Demand rate

aws.amazon.com/ec2/purchasing-options/

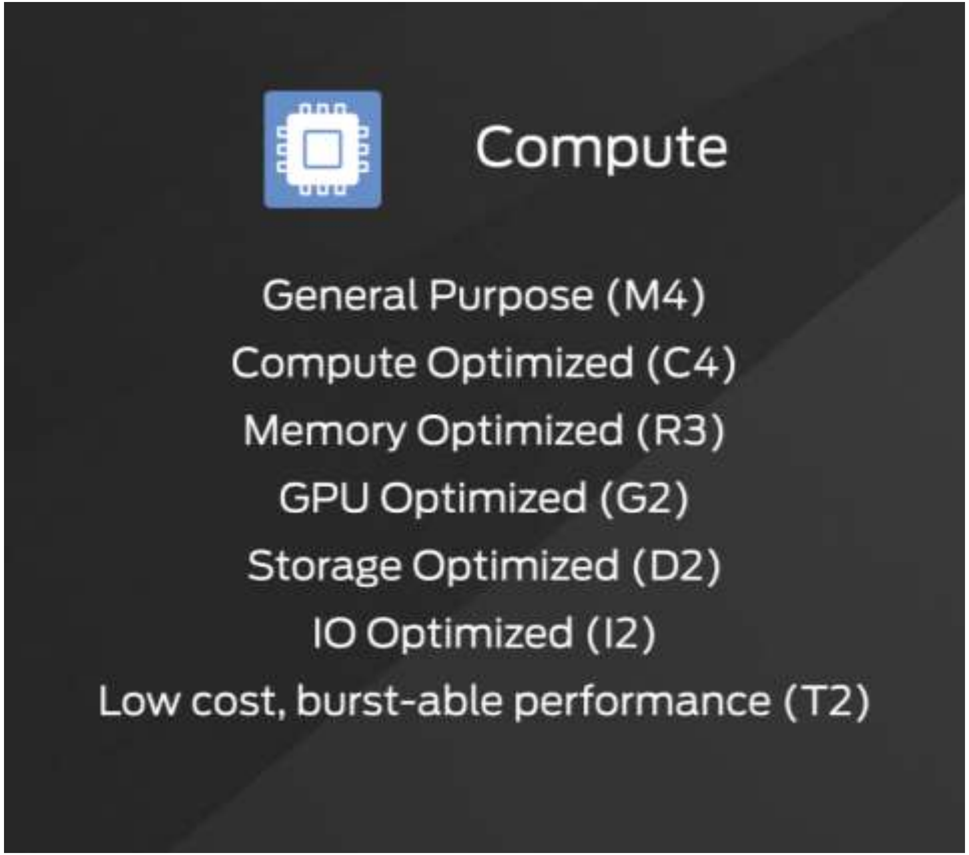
Import and export virtual machines



VM Import/Export enables you to easily import virtual machine images from your existing environment to Amazon EC2 instances and export them back to your on-premises environment. This offering allows you to leverage your existing investments in the virtual machines that you have built to meet your IT security, configuration management, and compliance requirements by bringing those virtual machines into Amazon EC2 as ready-to-use instances. You can also export imported instances back to your on-premises virtualization infrastructure, allowing you to deploy workloads across your IT infrastructure.

VM Import/Export is available at no additional charge beyond standard usage charges for Amazon EC2 and Amazon S3.

aws.amazon.com/ec2/vm-import/



Compute

General Purpose (M4)

Compute Optimized (C4)

Memory Optimized (R3)

GPU Optimized (G2)

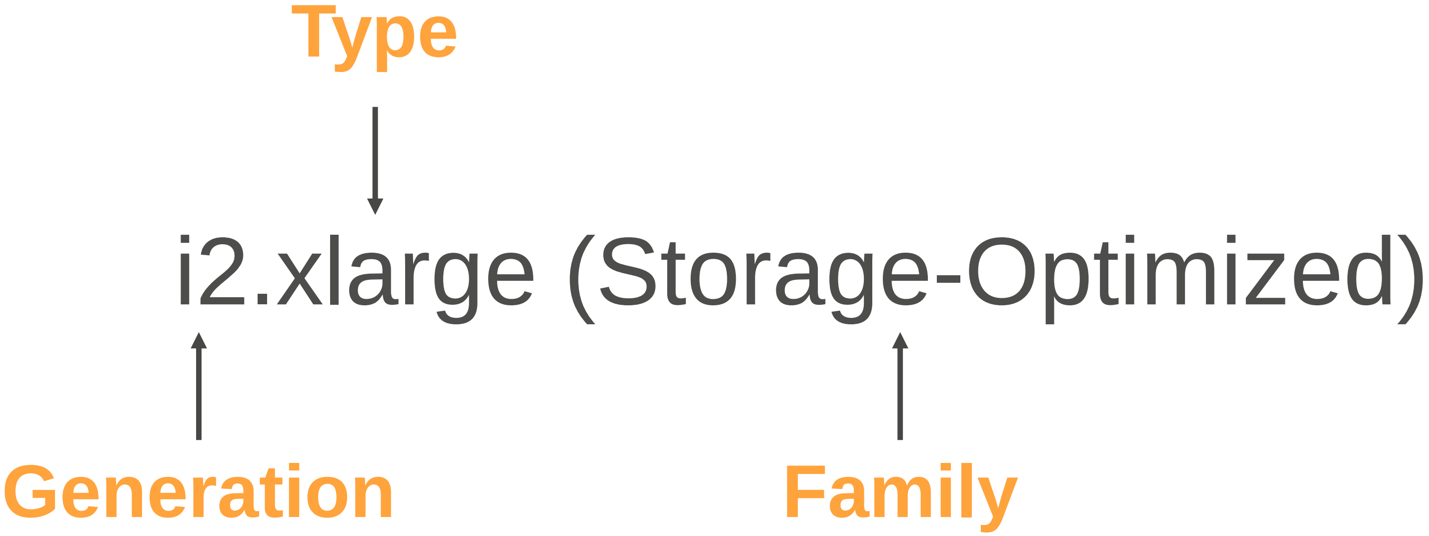
Storage Optimized (D2)

IO Optimized (I2)

Low cost, burst-able performance (T2)

Choice of instance families with differing resource ratios

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



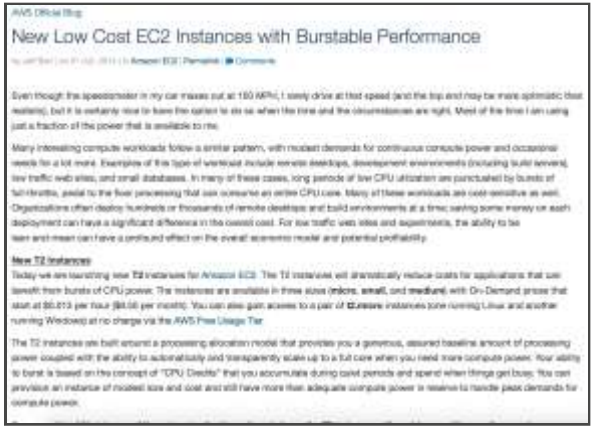
Type

i2.xlarge (Storage-Optimized)

Generation

Family

T2 Instances : Low Cost EC2 Instances with Burstable Performance



AWS Official Blog

New Low Cost EC2 Instances with Burstable Performance

by Jeff Barr | on 01 JUL 2014 | in Amazon EC2 | Permalink | Comments

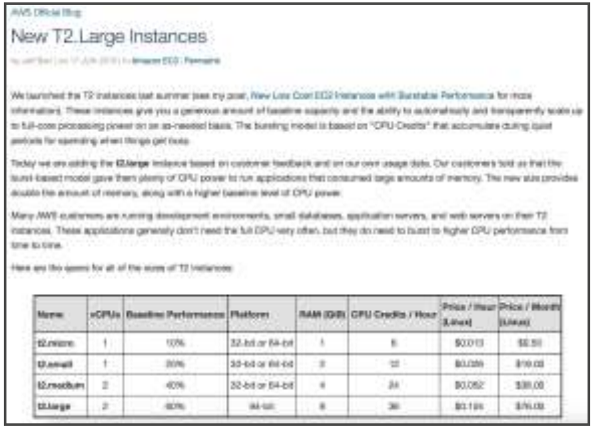
Even though the speedometer in my car maxes out at 150 MPH, I rarely drive at that speed (and the top end may be more optimistic than realistic), but it is certainly nice to have the option to do so when the time and the circumstances are right. Most of the time I am using just a fraction of the power that is available to me.

Many interesting compute workloads follow a similar pattern, with modest demands for continuous compute power and occasional needs for a lot more. Examples of this type of workload include remote desktops, development environments (including build servers), low traffic web sites, and small databases. In many of these cases, long periods of low CPU utilization are punctuated by bursts of full-throttle, pedal to the floor processing that can consume an entire CPU core. Many of these workloads are cost-sensitive as well. Organizations often deploy hundreds or thousands of remote desktops and build environments at a time; saving some money on each deployment can have a significant difference in the overall cost. For low traffic web sites and experiments, the ability to be lean and mean can have a profound effect on the overall economic model and potential profitability.

New T2 Instances

Today we are launching new **T2** instances for Amazon EC2. The T2 instances will dramatically reduce costs for applications that can benefit from bursts of CPU power. The instances are available in three sizes (**micro**, **small**, and **medium**) with On-Demand prices that start at $0.013 per hour ($9.50 per month). You can also gain access to a pair of **t2.micro** instances (one running Linux and another running Windows) at no charge via the AWS Free Usage Tier.

The T2 instances are built around a processing allocation model that provides you a generous, assured baseline amount of processing power coupled with the ability to automatically and transparently scale up to a full core when you need more compute power. Your ability to burst is based on the concept of "CPU Credits" that you accumulate during quiet periods and spend when things get busy. You can provision an instance of modest size and cost and still have more than adequate compute power in reserve to handle peak demands for compute power.



AWS Official Blog

New T2.Large Instances

by Jeff Barr | on 17 JUN 2015 | in Amazon EC2 | Permalink

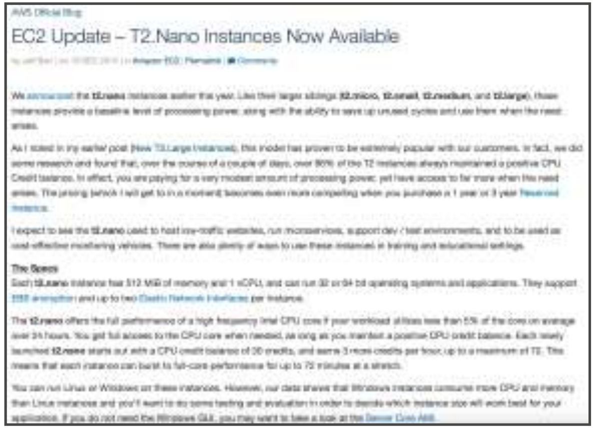
We launched the T2 instances last summer (see my post, New Low Cost EC2 Instances with Burstable Performance for more information). These instances give you a generous amount of baseline capacity and the ability to automatically and transparently scale up to full-core processing power on an as-needed basis. The bursting model is based on "CPU Credits" that accumulate during quiet periods for spending when things get busy.

Today we are adding the **t2.large** instance based on customer feedback and on our own usage data. Our customers told us that the burst-based model gave them plenty of CPU power to run applications that consumed large amounts of memory. The new size provides double the amount of memory, along with a higher baseline level of CPU power.

Many AWS customers are running development environments, small databases, application servers, and web servers on their T2 instances. These applications generally don't need the full CPU very often, but they do need to burst to higher CPU performance from time to time.

Here are the specs for all of the sizes of T2 instances:

Name	vCPUs	Baseline Performance	Platform	RAM (GiB)	CPU Credits / Hour	Price / Hour (Linux)	Price / Month (Linux)
t2.micro	1	10%	32-bit or 64-bit	1	6	$0.013	$9.50
t2.small	1	20%	32-bit or 64-bit	2	12	$0.026	$19.00
t2.medium	2	40%	32-bit or 64-bit	4	24	$0.052	$38.00
t2.large	2	60%	64-bit	8	36	$0.104	$76.00



AWS Official Blog

EC2 Update – T2.Nano Instances Now Available

by Jeff Barr | on 15 DEC 2015 | in Amazon EC2 | Permalink | Comments

We announced the **t2.nano** instances earlier this year. Like their larger siblings (**t2.micro**, **t2.small**, **t2.medium**, and **t2.large**), these instances provide a baseline level of processing power, along with the ability to save up unused cycles and use them when the need arises.

As I noted in my earlier post (New T2.Large Instances), this model has proven to be extremely popular with our customers. In fact, we did some research and found that, over the course of a couple of days, over 96% of the T2 instances always maintained a positive CPU Credit balance. In effect, you are paying for a very modest amount of processing power, yet have access to far more when the need arises. The pricing (which I will get to in a moment) becomes even more compelling when you purchase a 1 year or 3 year Reserved Instance.

I expect to see the **t2.nano** used to host low-traffic websites, run microservices, support dev / test environments, and to be used as cost-effective monitoring vehicles. There are also plenty of ways to use these instances in training and educational settings.

The Specs

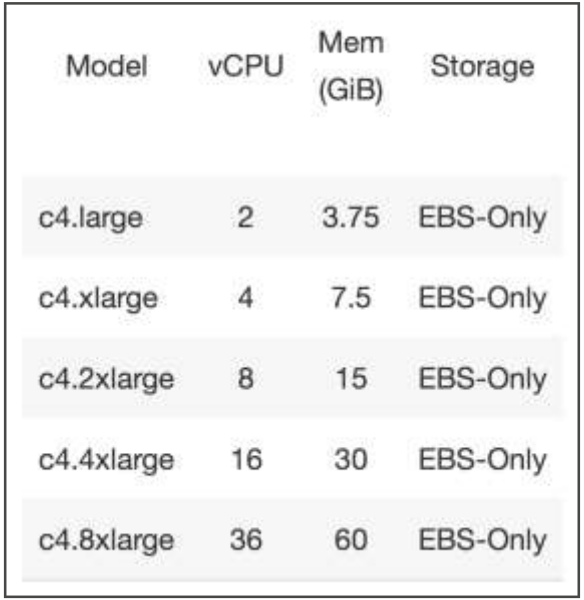
Each **t2.nano** instance has 512 MiB of memory and 1 vCPU, and can run 32 or 64 bit operating systems and applications. They support EBS encryption and up to two Elastic Network Interfaces per instance.

The **t2.nano** offers the full performance of a high frequency Intel CPU core if your workload utilizes less than 5% of the core on average over 24 hours. You get full access to the CPU core when needed, as long as you maintain a positive CPU credit balance. Each newly launched **t2.nano** starts out with a CPU credit balance of 30 credits, and earns 3 more credits per hour, up to a maximum of 72. This means that each instance can burst to full-core performance for up to 72 minutes at a stretch.

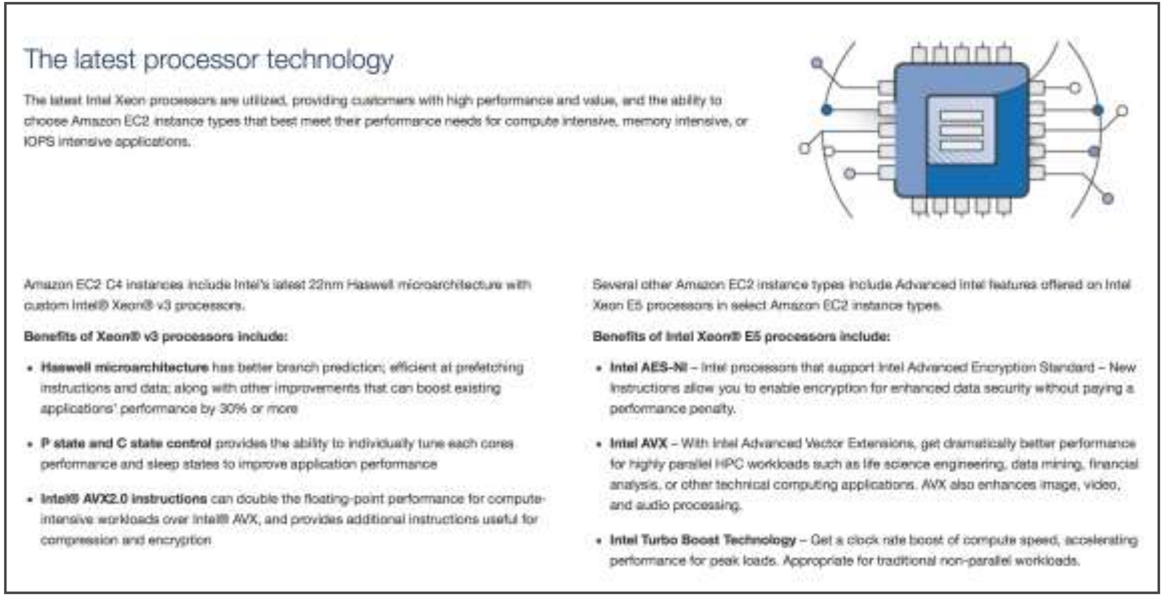
You can run Linux or Windows on these instances. However, our data shows that Windows instances consume more CPU and memory than Linux instances and you'll want to do some testing and evaluation in order to decide which instance size will work best for your application. If you do not need the Windows GUI, you may want to take a look at the Server Core AMI.

docs.aws.amazon.com/AWSEC2/latest/UserGuide/t2-instances.html

C4 Instances : Highest Compute Performance on Amazon EC2



Model	vCPU	Mem (GiB)	Storage
c4.large	2	3.75	EBS-Only
c4.xlarge	4	7.5	EBS-Only
c4.2xlarge	8	15	EBS-Only
c4.4xlarge	16	30	EBS-Only
c4.8xlarge	36	60	EBS-Only



The latest processor technology

The latest Intel Xeon processors are utilized, providing customers with high performance and value, and the ability to choose Amazon EC2 instance types that best meet their performance needs for compute intensive, memory intensive, or IOPS intensive applications.

Amazon EC2 C4 instances include Intel's latest 22nm Haswell microarchitecture with custom Intel® Xeon® v3 processors.

Benefits of Xeon® v3 processors include:

- **Haswell microarchitecture** has better branch prediction; efficient at prefetching instructions and data; along with other improvements that can boost existing applications' performance by 30% or more
- **P state and C state control** provides the ability to individually tune each cores performance and sleep states to improve application performance
- **Intel® AVX2.0 instructions** can double the floating-point performance for compute-intensive workloads over Intel® AVX, and provides additional instructions useful for compression and encryption

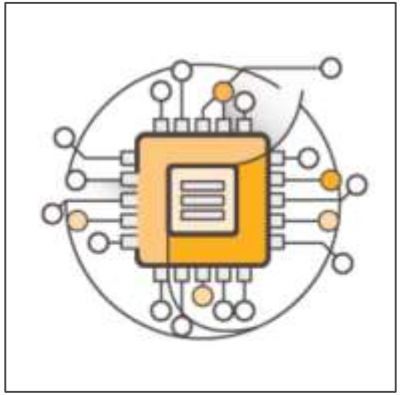
Several other Amazon EC2 instance types include Advanced Intel features offered on Intel Xeon E5 processors in select Amazon EC2 instance types.

Benefits of Intel Xeon® E5 processors include:

- **Intel AES-NI** – Intel processors that support Intel Advanced Encryption Standard – New Instructions allow you to enable encryption for enhanced data security without paying a performance penalty.
- **Intel AVX** – With Intel Advanced Vector Extensions, get dramatically better performance for highly parallel HPC workloads such as life science engineering, data mining, financial analysis, or other technical computing applications. AVX also enhances image, video, and audio processing.
- **Intel Turbo Boost Technology** – Get a clock rate boost of compute speed, accelerating performance for peak loads. Appropriate for traditional non-parallel workloads.

docs.aws.amazon.com/AWSEC2/latest/UserGuide/c4-instances.html

Now Available: Amazon EC2 X1 Instances



X1 Instances are a new addition to the Amazon EC2 memory-optimized instance family and are designed for running large-scale, in-memory applications and in-memory databases in the AWS cloud. X1 instances offer 1,952 GiB of DDR4 based memory, 8x the memory offered by any other Amazon EC2 instance. Each X1 instance is powered by four Intel® Xeon® E7 8880 v3 (Haswell) processors and offers 128 vCPUs.

Demo: Getting Started with EC2

Creating EC2 Instances with the AWS CLI

Amazon Machine Image ID

```
aws ec2 run-instances \
--image-id ami-c229c0a2 \
--instance-type m3.medium \
--count 5 \
--security-group-ids sg-9d5c2dfa \
--subnet-id subnet-608ebb14 \
--key-name ec2-masterclass
```

docs.aws.amazon.com/cli/latest/reference/ec2/index.html

Creating EC2 Instances with the AWS CLI

EC2 Instance type to run

```
aws ec2 run-instances \
--image-id ami-c229c0a2 \
--instance-type m3.medium \
--count 5 \
--security-group-ids sg-9d5c2dfa \
--subnet-id subnet-608ebb14 \
--key-name ec2-masterclass
```

docs.aws.amazon.com/cli/latest/reference/ec2/index.html

Creating EC2 Instances with the AWS CLI

```
aws ec2 run-instances \
--image-id ami-c229c0a2 \
--instance-type m3.medium \
--count 5 \
--security-group-ids sg-9d5c2dfa \
--subnet-id subnet-608ebb14 \
--key-name ec2-masterclass
```

Number of instances to run

docs.aws.amazon.com/cli/latest/reference/ec2/index.html

Creating EC2 Instances with the AWS CLI

```
aws ec2 run-instances \
--image-id ami-c229c0a2 \
--instance-type m3.medium \
--count 5 \
--security-group-ids sg-9d5c2dfa \
--subnet-id subnet-608ebb14 \
--key-name ec2-masterclass
```

Security group to apply to instance(s)

docs.aws.amazon.com/cli/latest/reference/ec2/index.html

Creating EC2 Instances with the AWS CLI

```
aws ec2 run-instances \
--image-id ami-c229c0a2 \
--instance-type m3.medium \
--count 5 \
--security-group-ids sg-9d5c2dfa \
--subnet-id subnet-608ebb14 \
--key-name ec2-masterclass
```

Subnet ID to launch instances within

docs.aws.amazon.com/cli/latest/reference/ec2/index.html

Creating EC2 Instances with the AWS CLI

```
aws ec2 run-instances \
--image-id ami-c229c0a2 \
--instance-type m3.medium \
--count 5 \
--security-group-ids sg-9d5c2dfa \
--subnet-id subnet-608ebb14 \
--key-name ec2-masterclass
```

Key pair to secure access to instance(s)

docs.aws.amazon.com/cli/latest/reference/ec2/index.html

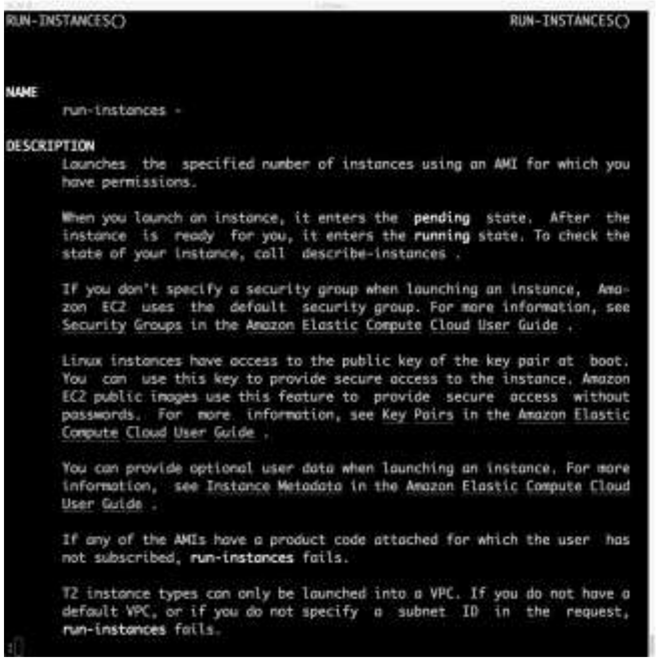
Managing EC2 via the AWS CLI

AWS CLI

Detailed help on a
specific command

In this case:

`aws ec2 run-instances`



```
RUN-INSTANCES()                                                RUN-INSTANCES()



NAME
       run-instances -

DESCRIPTION
       Launches  the  specified number of instances using an AMI for which you
       have permissions.

       When you launch an instance, it enters the  pending  state.  After  the
       instance  is  ready  for you, it enters the running state. To check the
       state of your instance, call  describe-instances .

       If you don't specify a security group when launching an instance,  Ama-
       zon  EC2  uses  the  default  security group. For more information, see
       Security Groups in the Amazon Elastic Compute Cloud User Guide .

       Linux instances have access to the public key of the key pair at  boot.
       You  can  use this key to provide secure access to the instance. Amazon
       EC2 public images use this feature to  provide  secure  access  without
       passwords.  For  more  information, see Key Pairs in the Amazon Elastic
       Compute Cloud User Guide .

       You can provide optional user data when launching an instance. For more
       information,  see Instance Metadata in the Amazon Elastic Compute Cloud
       User Guide .

       If any of the AMIs have a product code attached for which the user  has
       not subscribed, run-instances fails.

       T2 instance types can only be launched into a VPC. If you do not have a
       default VPC, or if you do not specify  a  subnet  ID  in  the  request,
       run-instances fails.
:
```

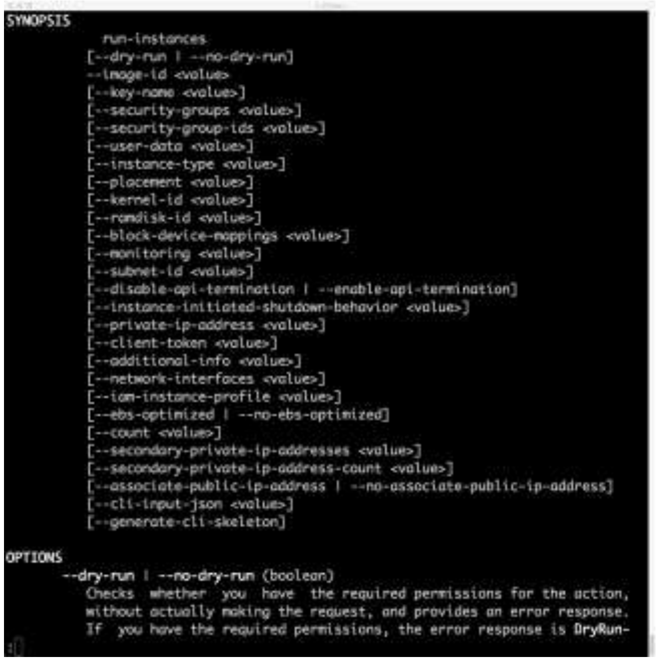
Managing EC2 via the AWS CLI

AWS CLI

Detailed help on a specific command

In this case:

`aws ec2 run-instances`



```
SYNOPSIS
            run-instances
          [--dry-run | --no-dry-run]
          --image-id <value>
          [--key-name <value>]
          [--security-groups <value>]
          [--security-group-ids <value>]
          [--user-data <value>]
          [--instance-type <value>]
          [--placement <value>]
          [--kernel-id <value>]
          [--ramdisk-id <value>]
          [--block-device-mappings <value>]
          [--monitoring <value>]
          [--subnet-id <value>]
          [--disable-api-termination | --enable-api-termination]
          [--instance-initiated-shutdown-behavior <value>]
          [--private-ip-address <value>]
          [--client-token <value>]
          [--additional-info <value>]
          [--network-interfaces <value>]
          [--iam-instance-profile <value>]
          [--ebs-optimized | --no-ebs-optimized]
          [--count <value>]
          [--secondary-private-ip-addresses <value>]
          [--secondary-private-ip-address-count <value>]
          [--associate-public-ip-address | --no-associate-public-ip-address]
          [--cli-input-json <value>]
          [--generate-cli-skeleton]

OPTIONS
       --dry-run | --no-dry-run (boolean)
          Checks  whether  you  have  the required permissions for the action,
          without actually making the request, and provides an error response.
          If  you have the required permissions, the error response is DryRun-
:
```

docs.aws.amazon.com/cli/latest/reference/ec2/index.html

Launching and Terminating Instances

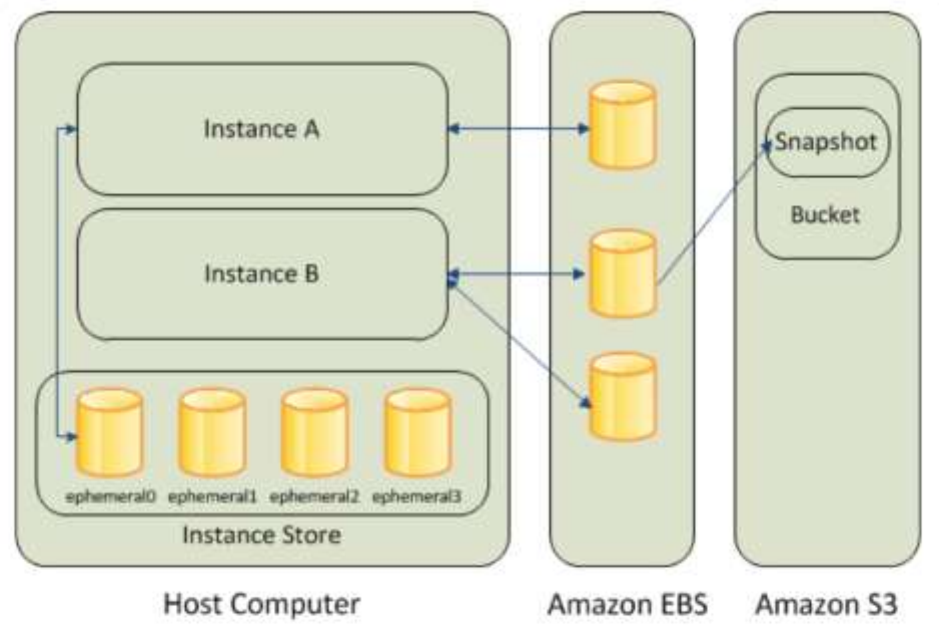
Python Boto3

```
>>> import boto3
>>> client = boto3.client('ec2')
>>> response=client.run_instances(DryRun=True, ImageId='ami-c229c0a2', MinCount=5, MaxCount=5,
InstanceType='m3.medium', KeyName='ec2-masterclass', SecurityGroupIds=['sg-9d5c2dfa'],
SubnetId='subnet-608ebb14')

>>> response = client.terminate_instances(InstanceIds=['i-d4116c0c'])
>>> print response
{u'TerminatingInstances': [{u'InstanceId': 'i-d4116c0c', u'CurrentState': {u'Code': 32,
u'Name': 'shutting-down'}, u'PreviousState': {u'Code': 16, u'Name': 'running'}}],
'ResponseMetadata': {'HTTPStatusCode': 200, 'RequestId': 'e7b021e8-dc9a-4757-8698-
9c747a5a702a'}}

>>> instance=boto3.resource('ec2')
>>> instance = ec2.instance('i-d3116c0b')
>>> instance.terminate(){u'TerminatingInstances': [{u'InstanceId': 'i-d3116c0b',
u'CurrentState': {u'Code': 32, u'Name': 'shutting-down'}, u'PreviousState': {u'Code': 16,
u'Name': 'running'}}], 'ResponseMetadata': {'HTTPStatusCode': 200, 'RequestId': 'a7ca0aeb-b35e-
4555-8d89-faacacb3c2ca'}}
>>>
```

github.com/boto/boto3

Storage

Data Storage Options



Data Storage Options

Instance Store

Physically attached
to the host computer

Type and amount differs
by instance type

Data **dependent** upon instance lifecycle

Amazon EBS

Persistent block level storage volumes

Magnetic

General Purpose (SSD)

Provisioned IOPS (SSD)

Data **independent** of instance lifecycle

Instance Store

- Physically attached to the host computer
- Type and amount differs by instance type
- Data **dependent** upon instance lifecycle

Instance store data persists if:

- The OS in the instance is rebooted
- The instance is restarted

Instance store data is lost when:

- An underlying instance drive fails
- An EBS-backed instance is stopped
- The instance is terminated

Amazon EBS

- **Persistent** block level storage volumes

Magnetic

- General Purpose (SSD)
- Provisioned IOPS (SSD)

- Data **independent** of instance lifecycle

EBS Volumes

- EBS volumes **automatically replicated** within the **Availability Zone (AZ)** in which they are created

- Use EBS-optimized instances to deliver **dedicated throughput** between Amazon EC2 and Amazon EBS, with options between 500 and 4,000 Mbps, depending on the instance type

Amazon EBS

- **Persistent** block level storage volumes

Magnetic

- General Purpose (SSD)
- Provisioned IOPS (SSD)

- Data **independent** of instance lifecycle

EBS Volumes

- EBS volumes **attached** to a running instance automatically detach from the instance with their data intact when that instance is terminated.

- EBS volumes created and attached to an instance at **launch** are deleted when that instance is terminated. You can modify this behavior by changing the value of the flag **DeleteOnTermination**.

Amazon EBS

- **Persistent** block level storage volumes

Magnetic

- General Purpose (SSD)
- Provisioned IOPS (SSD)

- Data **independent** of instance lifecycle

EBS Snapshots

- An EBS snapshot is a **point-in-time backup copy** of an EBS volume that is stored in Amazon S3

- Snapshots are **incremental**, only the blocks that have changed after your most recent snapshot are saved

Amazon EBS

- **Persistent** block level storage volumes

  Magnetic

- General Purpose (SSD)
- Provisioned IOPS (SSD)

- Data **independent** of instance lifecycle

EBS Snapshots

- When you delete a snapshot, only the data **exclusive** to that snapshot is removed

- Can be **shared** across AWS accounts or **copied** across AWS regions

Amazon EBS

- **Persistent** block level storage volumes

Magnetic

- General Purpose (SSD)
- Provisioned IOPS (SSD)

- Data **independent** of instance lifecycle

EBS Encryption

- Data stored at rest on the volume, disk I/O, and snapshots created from the volume are **all** encrypted

- The encryption occurs on the servers that **host** Amazon EC2 instances, providing encryption of **data-in-transit** from EC2 instances to EBS storage

Amazon EBS

- **Persistent** block level storage volumes

Magnetic

- General Purpose (SSD)
- Provisioned IOPS (SSD)

- Data **independent** of instance lifecycle

EBS Encryption

- Uses **AWS Key Management Service** (AWS KMS) master keys unless you select a **Customer Master Key** (CMK).

- Creating **your own CMK** gives you the ability to create, rotate, disable, define access controls, and audit the encryption keys.

EBS Volumes: Larger & Faster

General Purpose (SSD)	Provisioned IOPS (SSD)
Up to 16TB	Up to 16TB
10,000 IOPS (burst)	20,000 IOPS
Up to 160 MBps	Up to 320 MBps

Demo: Working with EBS Volumes

Networking

Amazon VPC

Amazon VPC

A virtual network in your own logically isolated area within the AWS cloud populated by infrastructure, platform, and application services that share common security and interconnection

aws.amazon.com/vpc/

VPC Networking

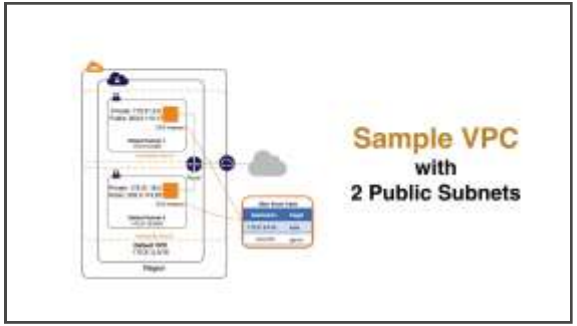
- Elastic Network Interface (ENI)
- Subnet
- Network Access Control List (NACL)
- Route Table
- Internet Gateway
- Virtual Private Gateway
- Route 53 Private Hosted Zone

VPC Network Topology

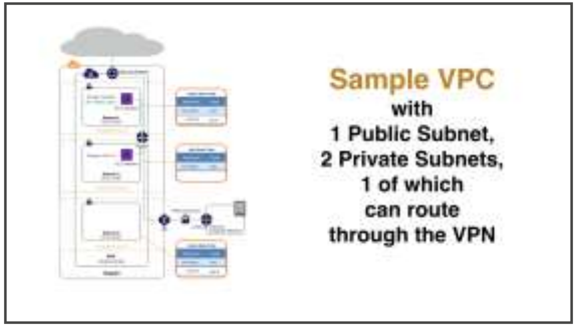
- A VPC can span multiple AZs, but each subnet must reside entirely within one AZ

- Use at least 2 subnets in different AZs for each layer of your network

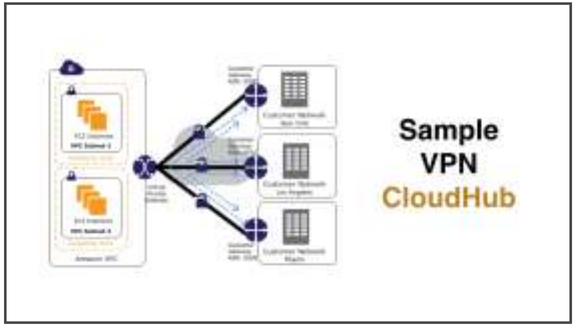
Control of Subnets and Routing Tables



Sample VPC
with
2 Public Subnets

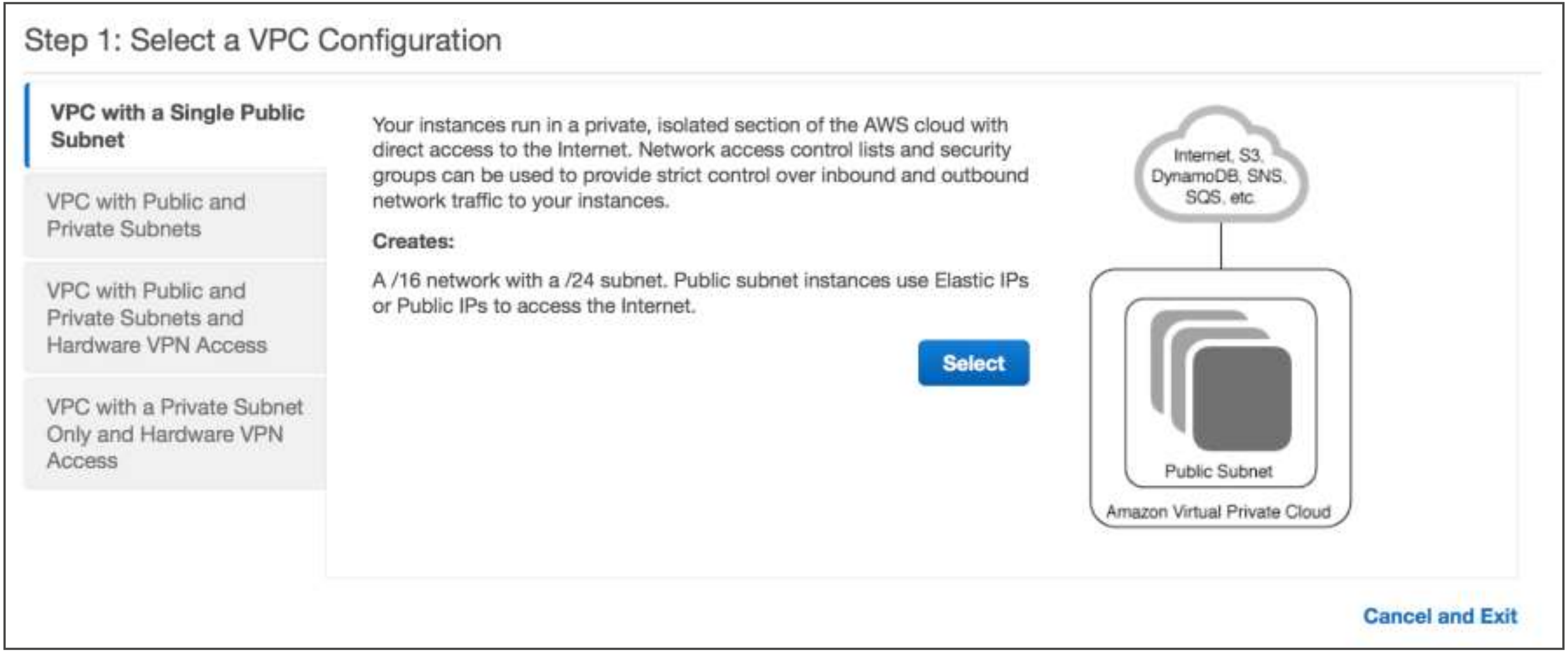


Sample VPC
with
1 Public Subnet,
2 Private Subnets,
1 of which
can route
through the VPN



Sample
VPN
CloudHub

VPC Creation with the VPC Wizard



Step 1: Select a VPC Configuration

VPC with a Single Public Subnet

VPC with Public and Private Subnets

VPC with Public and Private Subnets and Hardware VPN Access

VPC with a Private Subnet Only and Hardware VPN Access

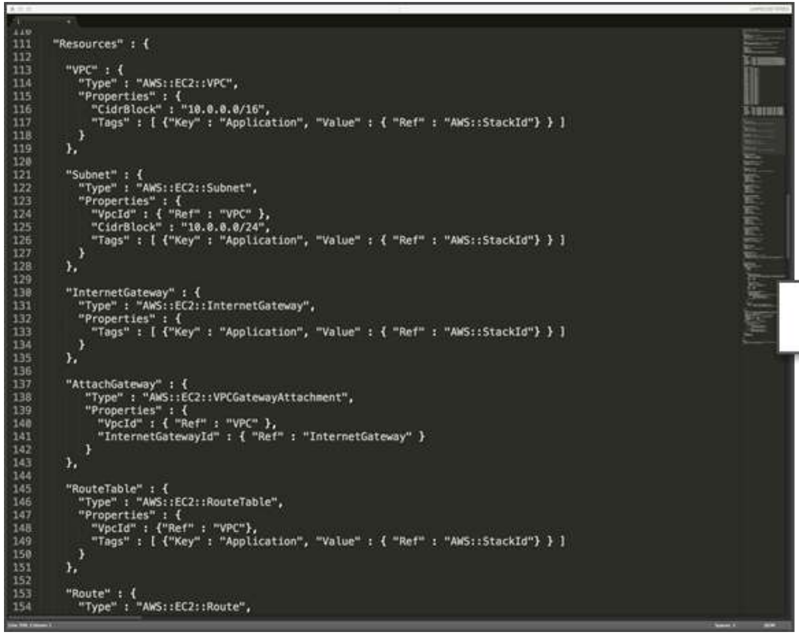
Your instances run in a private, isolated section of the AWS cloud with direct access to the Internet. Network access control lists and security groups can be used to provide strict control over inbound and outbound network traffic to your instances.

Creates:

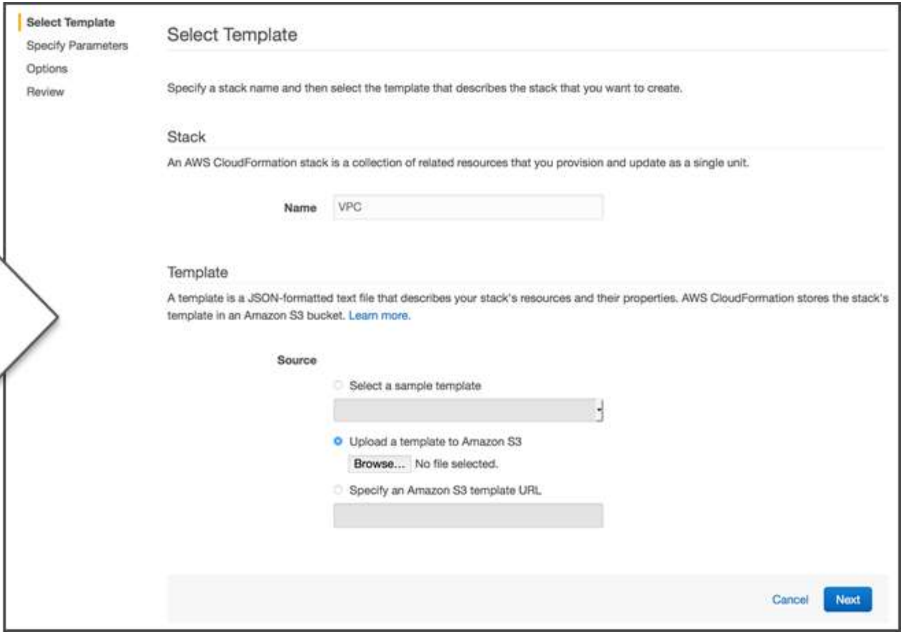
A /16 network with a /24 subnet. Public subnet instances use Elastic IPs or Public IPs to access the Internet.

Select
Internet, S3, DynamoDB, SNS, SQS, etc.
Public Subnet
Amazon Virtual Private Cloud
Cancel and Exit

VPC Creation with AWS CloudFormation



```
  "Resources" : {

    "VPC" : {
      "Type" : "AWS::EC2::VPC",
      "Properties" : {
        "CidrBlock" : "10.0.0.0/16",
        "Tags" : [ {"Key" : "Application", "Value" : { "Ref" : "AWS::StackId"} } ]
      }
    },

    "Subnet" : {
      "Type" : "AWS::EC2::Subnet",
      "Properties" : {
        "VpcId" : { "Ref" : "VPC" },
        "CidrBlock" : "10.0.0.0/24",
        "Tags" : [ {"Key" : "Application", "Value" : { "Ref" : "AWS::StackId"} } ]
      }
    },

    "InternetGateway" : {
      "Type" : "AWS::EC2::InternetGateway",
      "Properties" : {
        "Tags" : [ {"Key" : "Application", "Value" : { "Ref" : "AWS::StackId"} } ]
      }
    },

    "AttachGateway" : {
       "Type" : "AWS::EC2::VPCGatewayAttachment",
       "Properties" : {
         "VpcId" : { "Ref" : "VPC" },
         "InternetGatewayId" : { "Ref" : "InternetGateway" }
       }
    },

    "RouteTable" : {
      "Type" : "AWS::EC2::RouteTable",
      "Properties" : {
        "VpcId" : {"Ref" : "VPC"},
        "Tags" : [ {"Key" : "Application", "Value" : { "Ref" : "AWS::StackId"} } ]
      }
    },

    "Route" : {
      "Type" : "AWS::EC2::Route",
```



Select Template

Specify a stack name and then select the template that describes the stack that you want to create.

Stack

An AWS CloudFormation stack is a collection of related resources that you provision and update as a single unit.

Name VPC

Template

A template is a JSON-formatted text file that describes your stack's resources and their properties. AWS CloudFormation stores the stack's template in an Amazon S3 bucket. Learn more.

Source

○ Select a sample template

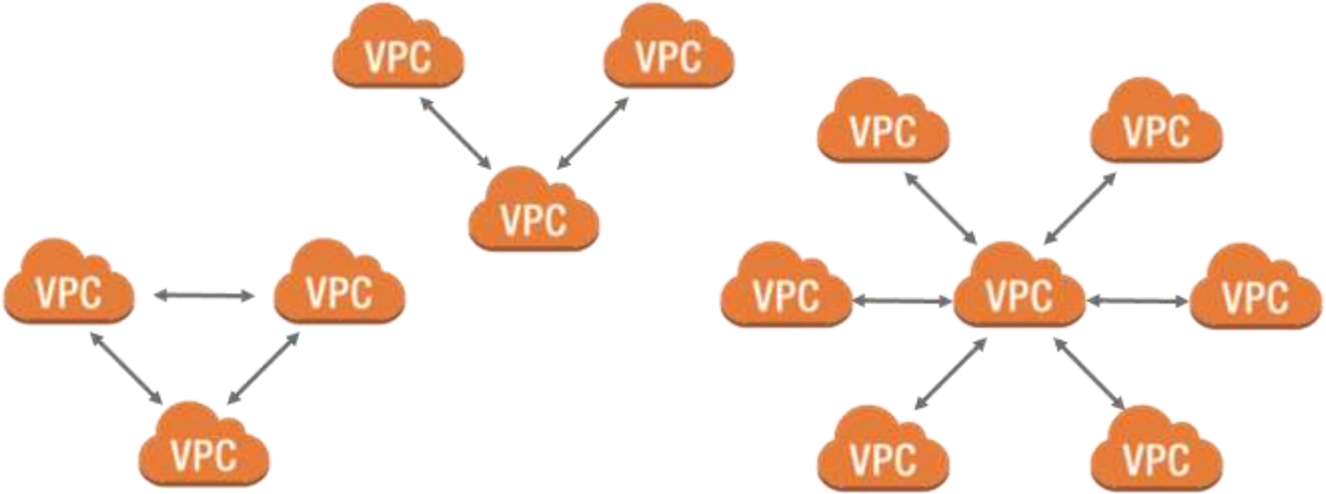
● Upload a template to Amazon S3

Browse... No file selected.

○ Specify an Amazon S3 template URL

VPC Peering

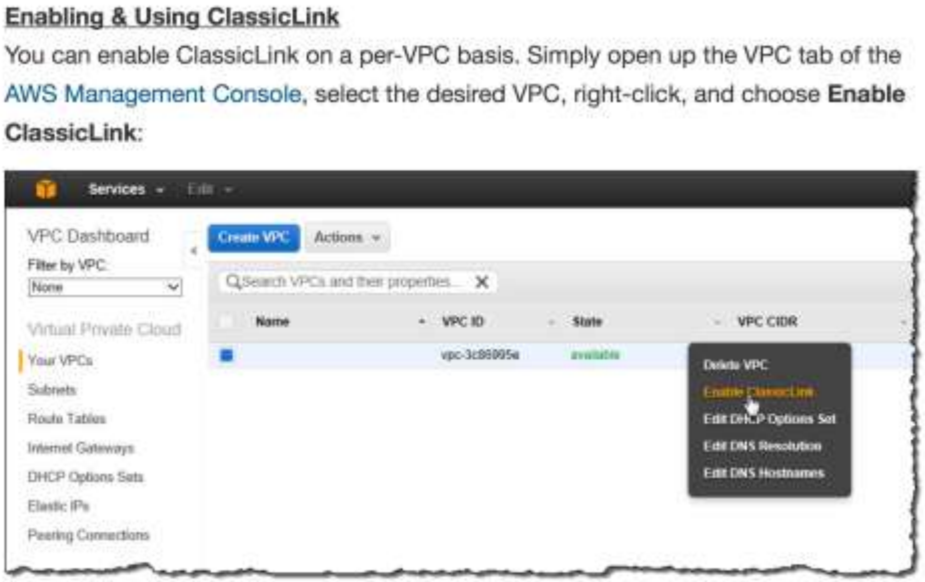
A networking connection between two VPCs



docs.aws.amazon.com/AmazonVPC/latest/UserGuide/vpc-peering.html

ClassicLink

Private Communication Between Classic EC2 Instances & VPC Resources



Enabling & Using ClassicLink

You can enable ClassicLink on a per-VPC basis. Simply open up the VPC tab of the AWS Management Console, select the desired VPC, right-click, and choose **Enable ClassicLink**:

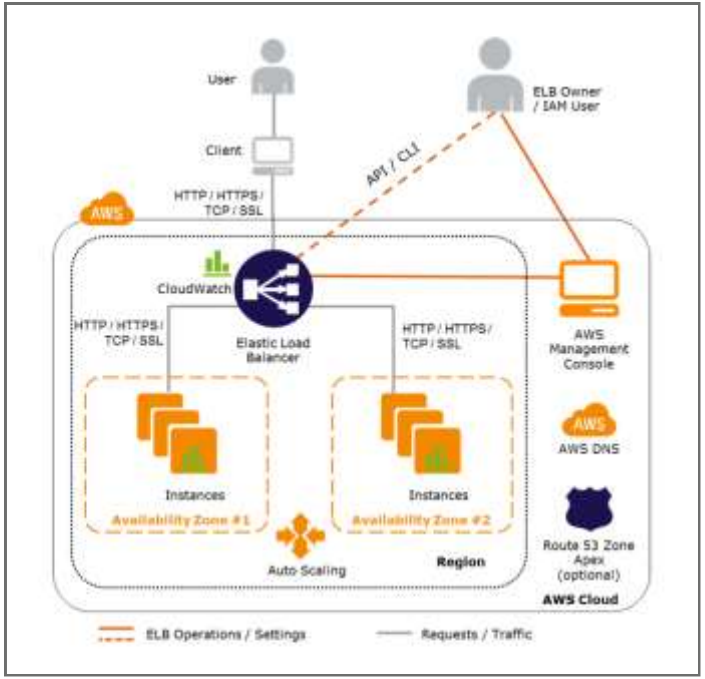
Name	VPC ID	State	VPC CIDR
	vpc-3c86095e	available	

aws.amazon.com/blogs/aws/classiclink-private-communication-between-classic-ec2-instances-vpc-resources/

Elastic Load Balancing

aws.amazon.com/elasticloadbalancing/

Elastic Load Balancing



- Timeout Configuration
- Connection Draining
- Cross-zone Load Balancing

aws.amazon.com/elasticloadbalancing/

Monitoring, Metrics & Logs

Amazon CloudWatch

aws.amazon.com/cloudwatch/

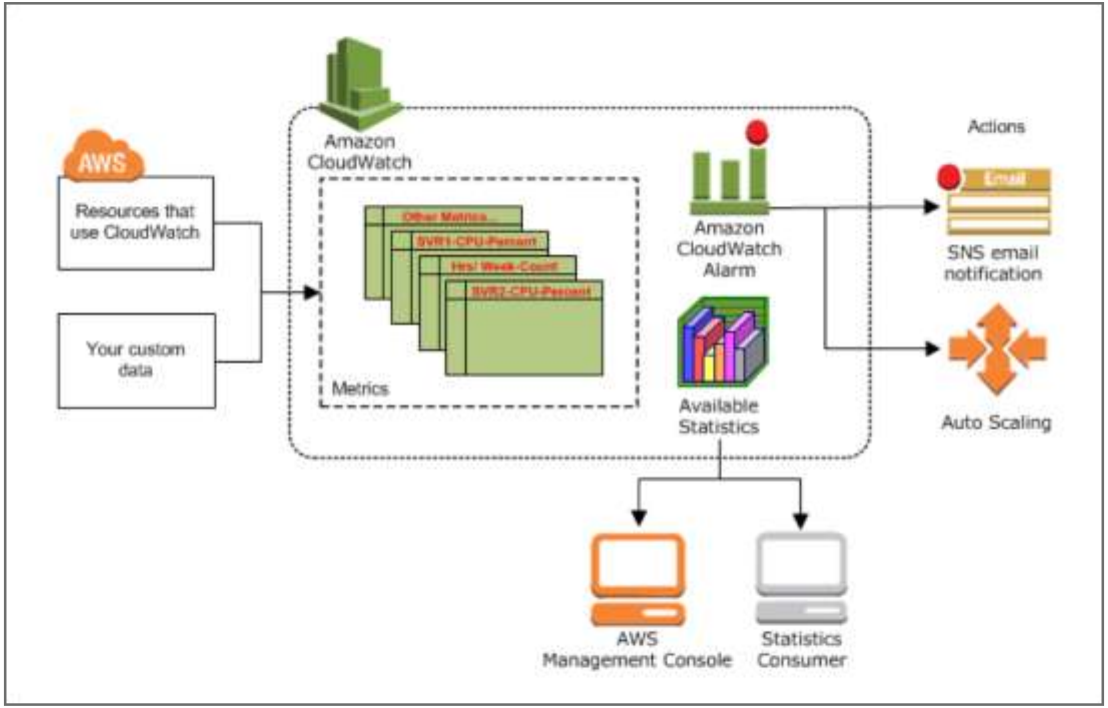
Amazon CloudWatch

- A monitoring service for AWS cloud resources and the applications you run on AWS.

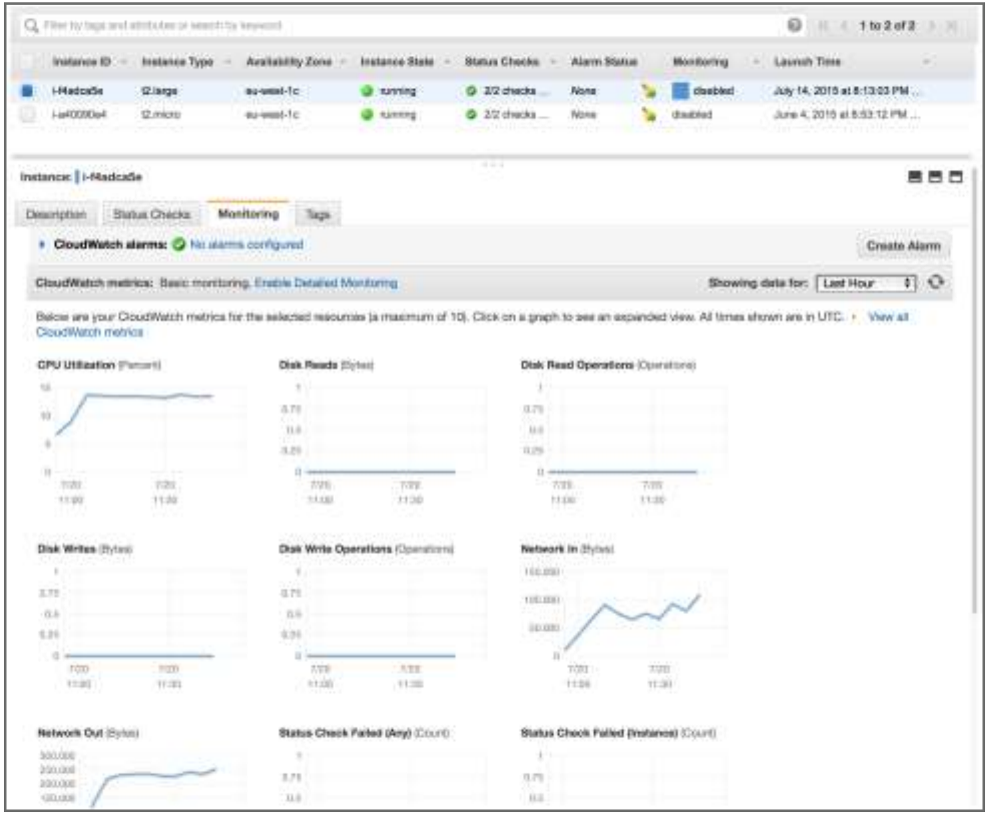
- Use Amazon CloudWatch to collect and track metrics, collect and monitor log files, and set alarms.

aws.amazon.com/cloudwatch/

Amazon CloudWatch



CloudWatch Metrics in the EC2 Console



Filter by tags and attributes or search by keyword

1 to 2 of 2

	Instance ID	Instance Type	Availability Zone	Instance State	Status Checks	Alarm Status	Monitoring	Launch Time
■	i-f4adca5e	t2.large	eu-west-1c	● running	✓ 2/2 checks ...	None	disabled	July 14, 2015 at 8:13:03 PM ...
□	i-a40090e4	t2.micro	eu-west-1c	● running	✓ 2/2 checks ...	None	disabled	June 4, 2015 at 8:53:12 PM ...

Instance: i-f4adca5e

Description | Status Checks | Monitoring | Tags

CloudWatch alarms: ✓ No alarms configured

Create Alarm

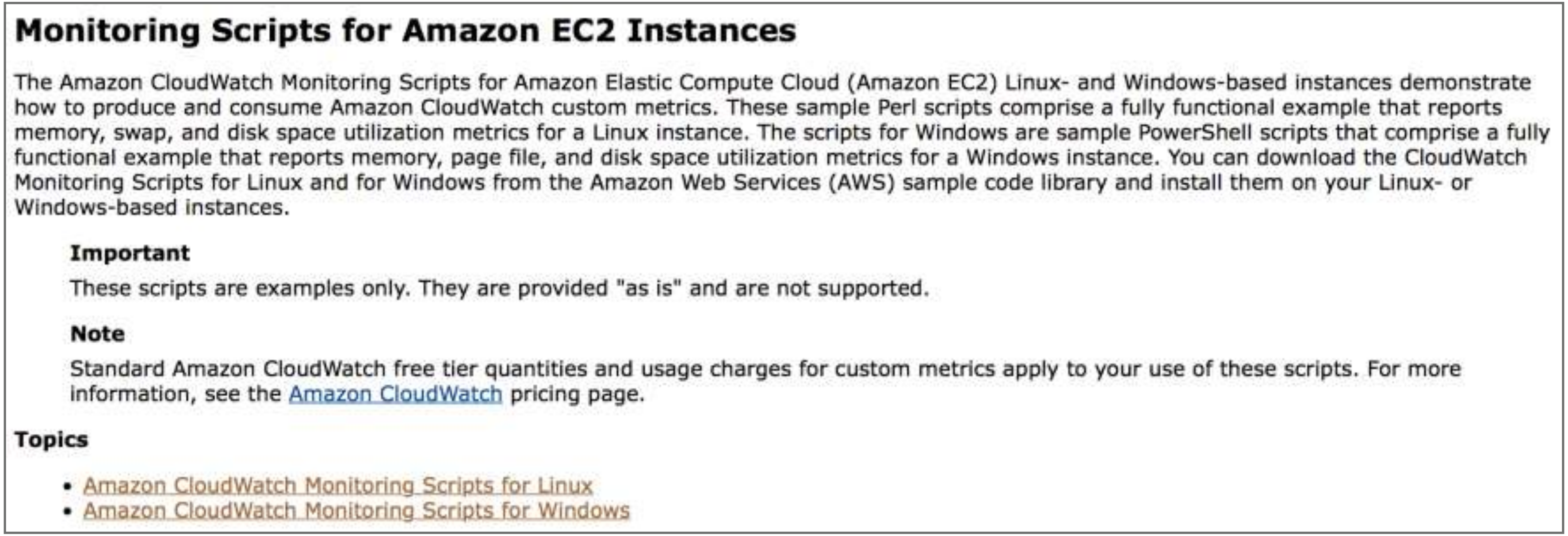
CloudWatch metrics: Basic monitoring. Enable Detailed Monitoring

Showing data for: Last Hour

Below are your CloudWatch metrics for the selected resources (a maximum of 10). Click on a graph to see an expanded view. All times shown are in UTC. View all CloudWatch metrics

CPU Utilization (Percent)
Disk Reads (Bytes)
Disk Read Operations (Operations)
Disk Writes (Bytes)
Disk Write Operations (Operations)
Network In (Bytes)
Network Out (Bytes)
Status Check Failed (Any) (Count)
Status Check Failed (Instance) (Count)
1
0.75
0.5
0.25
0
150,000
100,000
50,000
300,000
250,000
200,000
150,000
7/20
11:00
11:30

Monitoring Scripts for EC2 Instances



Monitoring Scripts for Amazon EC2 Instances

The Amazon CloudWatch Monitoring Scripts for Amazon Elastic Compute Cloud (Amazon EC2) Linux- and Windows-based instances demonstrate how to produce and consume Amazon CloudWatch custom metrics. These sample Perl scripts comprise a fully functional example that reports memory, swap, and disk space utilization metrics for a Linux instance. The scripts for Windows are sample PowerShell scripts that comprise a fully functional example that reports memory, page file, and disk space utilization metrics for a Windows instance. You can download the CloudWatch Monitoring Scripts for Linux and for Windows from the Amazon Web Services (AWS) sample code library and install them on your Linux- or Windows-based instances.

> **Important**
>
> These scripts are examples only. They are provided "as is" and are not supported.

> **Note**
>
> Standard Amazon CloudWatch free tier quantities and usage charges for custom metrics apply to your use of these scripts. For more information, see the Amazon CloudWatch pricing page.

Topics

- Amazon CloudWatch Monitoring Scripts for Linux
- Amazon CloudWatch Monitoring Scripts for Windows

docs.aws.amazon.com/AmazonCloudWatch/latest/DeveloperGuide/mon-scripts.html

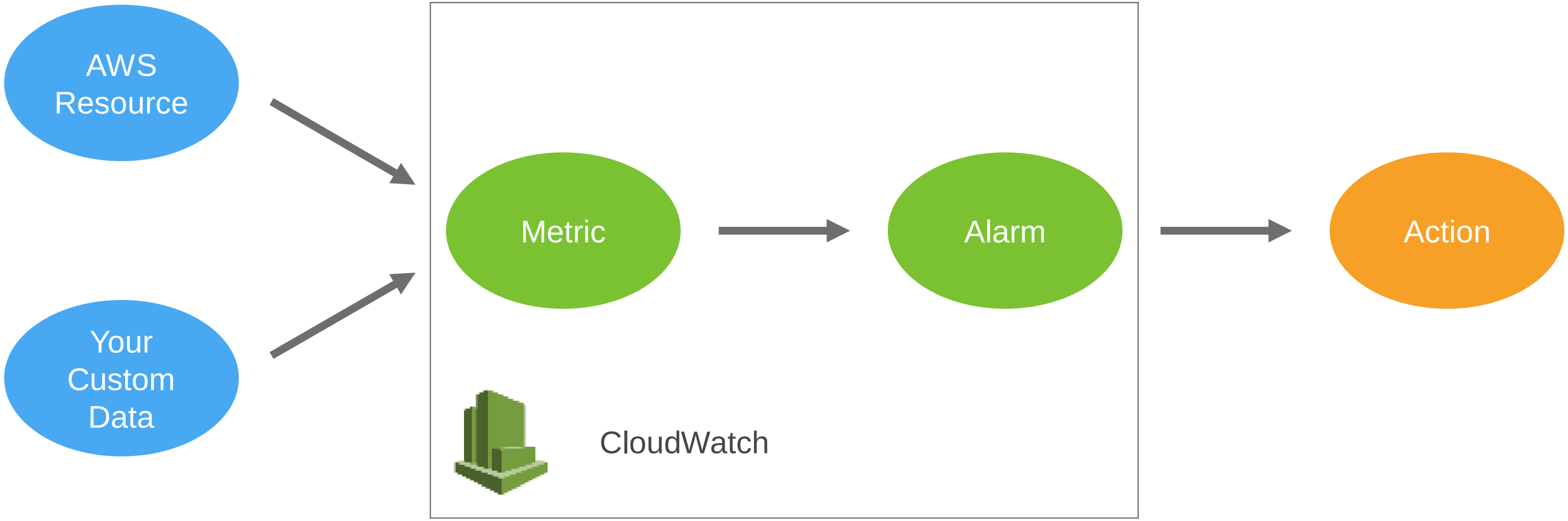
Amazon CloudWatch Logs

- Monitor applications and systems using log data
- Store in a highly durable storage & set retention policies
- Access your log files via Web, CLI or SDK

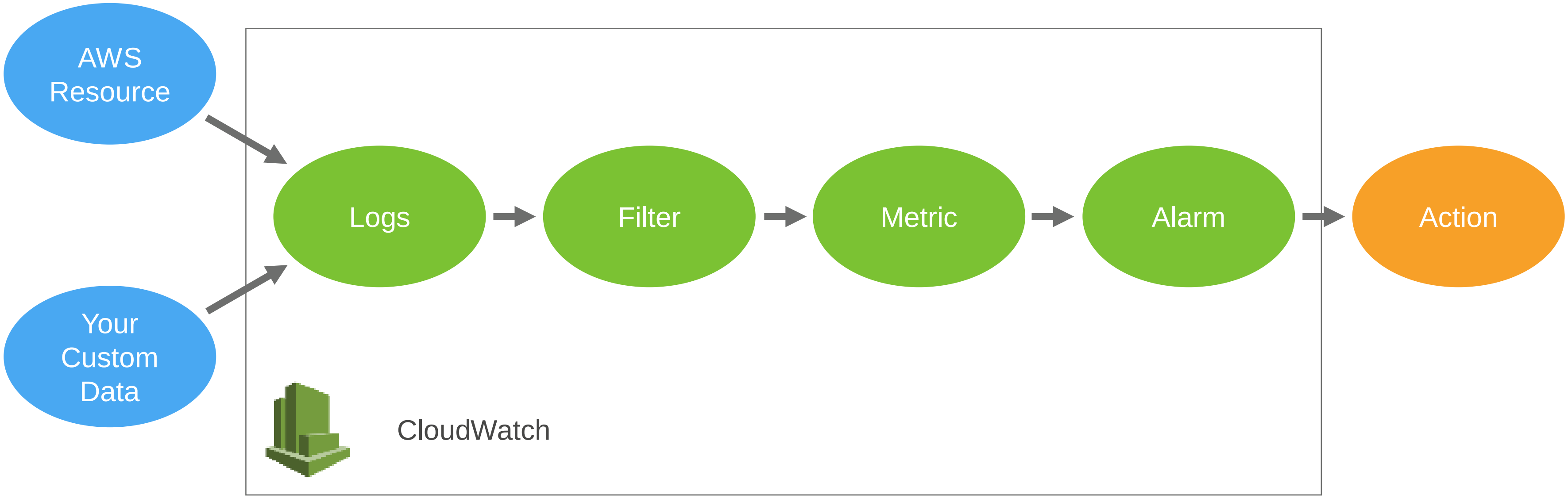
- Amazon EC2 (Linux & Windows)
- AWS Lambda
- …

docs.aws.amazon.com/AmazonCloudWatch/latest/DeveloperGuide/WhatIsCloudWatchLogs.html

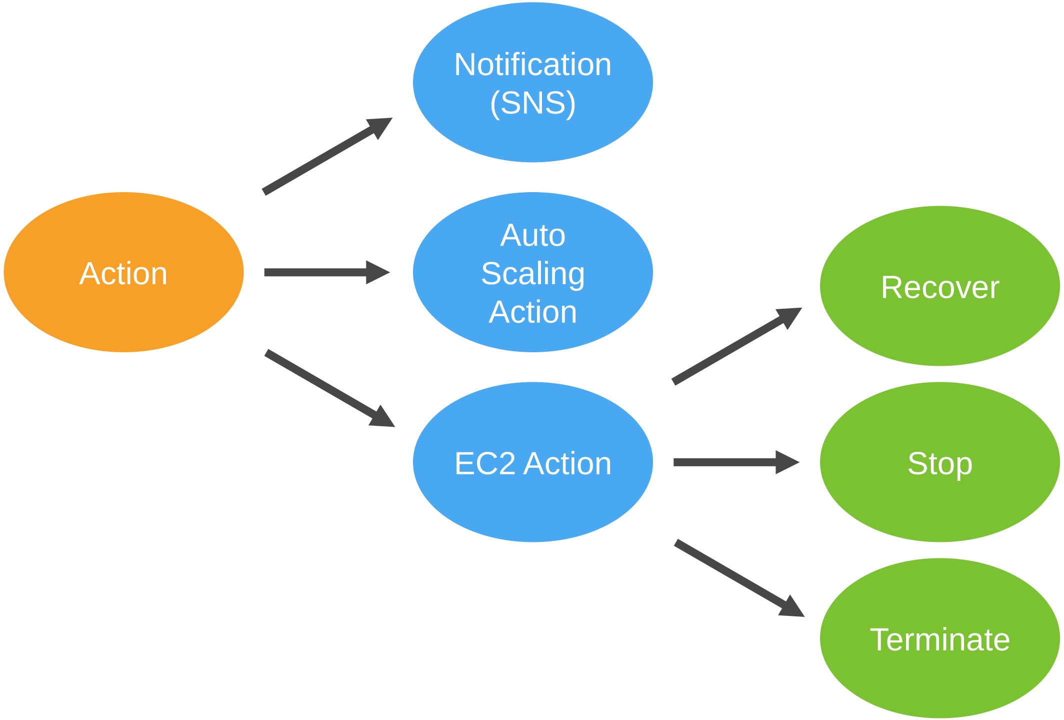
CloudWatch Metrics & Alarms



CloudWatch Logs + Filter



Alarm Actions



Amazon EC2 Auto Recovery

Use this action together with Status Checks to automate instance recovery

Security & Access Control

Access Credentials

Access key and secret key used to authenticate when accessing AWS APIs

Key Pairs

Public key and private key used to authenticate when accessing an Amazon EC2 instance

Use IAM Roles to pass access credentials to EC2 instances

Demo: Using IAM Roles

Deployment

Amazon Machine Images

Amazon maintained

Set of Linux and Windows images

Kept up to date by Amazon in each region

Community maintained

Images published by other AWS users

Managed and maintained by Marketplace partners

Your machine images

AMIs you have created from EC2 instances

Can be kept private or shared with other accounts

Create an AMI

- Start an instance
- Configure the instance
- Create an AMI from your instance
- Start new ones from the AMI

Create an AMI

- Start an instance
- Configure the instance
- Create an AMI from your instance
- Start new ones from the AMI

Configure Dynamically

- Launch an instance
- Use metadata service & cloud-init to perform actions on the instance when it launches

EC2 Instance Metadata

```
$ curl http://169.254.169.254/latest/meta-data/

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/
public-hostname
public-ipv4
public-keys/
reservation-id
security-groups
services/

$ curl http://169.254.169.254/latest/user-data

…
```

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

EC2 User-data

- AWS provided AMIs include services that access and execute the contents of the User data attribute in EC2 Instance Metadata at instance creation time
- This can be used for automated instance bootstrapping at instance creation time

Windows

```
<script>…</script>
```

or

```
<powershell>…</powershell>
```

Linux

```
#!/bin/bash
yum update -y
```

Create an AMI

- Start an instance
- Configure the instance
- Create an AMI from your instance
- Start new ones from the AMI

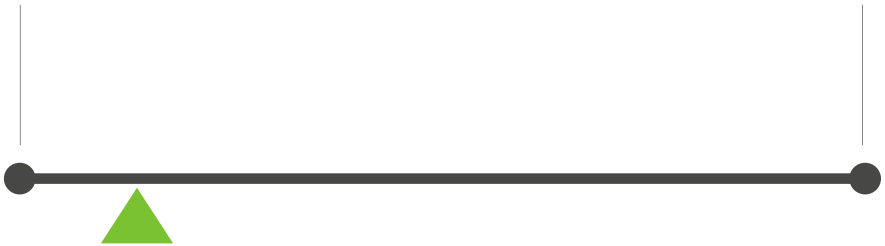
Configure Dynamically

- Launch an instance
- Use metadata service & cloud-init to perform actions on the instance when it launches

These two approaches are not mutually exclusive

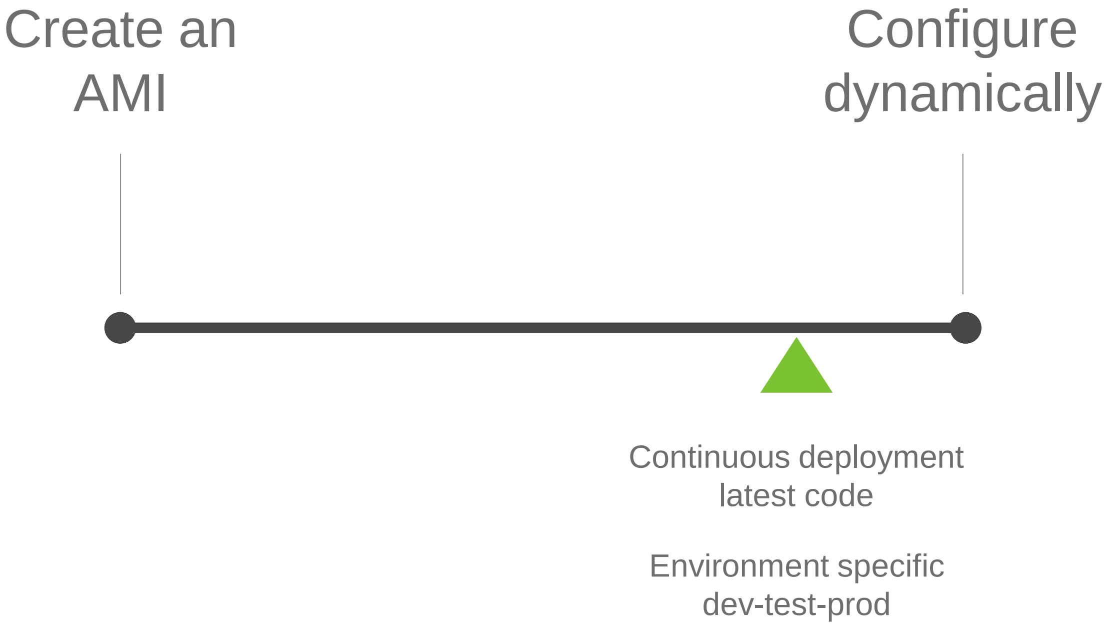
Create an AMI

Configure dynamically



Time consuming configuration
startup time

Static configurations
less change management



Create an AMI

Configure dynamically

Continuous deployment
latest code

Environment specific
dev-test-prod

Auto Scaling

Maintain EC2 instance availability

- Detects impaired EC2 instances
- Replaces the instances automatically

Automatically Scale Your Amazon EC2 Fleet

- Follow the demand curve for your applications
- Reduce the need to manually provision Amazon EC2 capacity
- Run at optimal utilization

Launch Configuration

Describes what Auto Scaling will create when adding Instances

Only one active launch configuration at a time

```
aws autoscaling create-launch-configuration
   --launch-configuration-name launch-config
   --image-id ami-54cf5c3d
   --instance-type m3.medium
   --key-name mykey
   --security-groups webservers
```

Auto Scaling Group

Scaling managed grouping of EC2 instances

Automatically scale the number of instances by policy

```
aws autoscaling create-auto-scaling-group
   --auto-scaling-group-name autoscaling-group
   --availability-zones eu-west-1a eu-west-1b
   --launch-configuration launch-config
   --load-balancer-names myELB
   --min-size 1
   --max-size 5
```

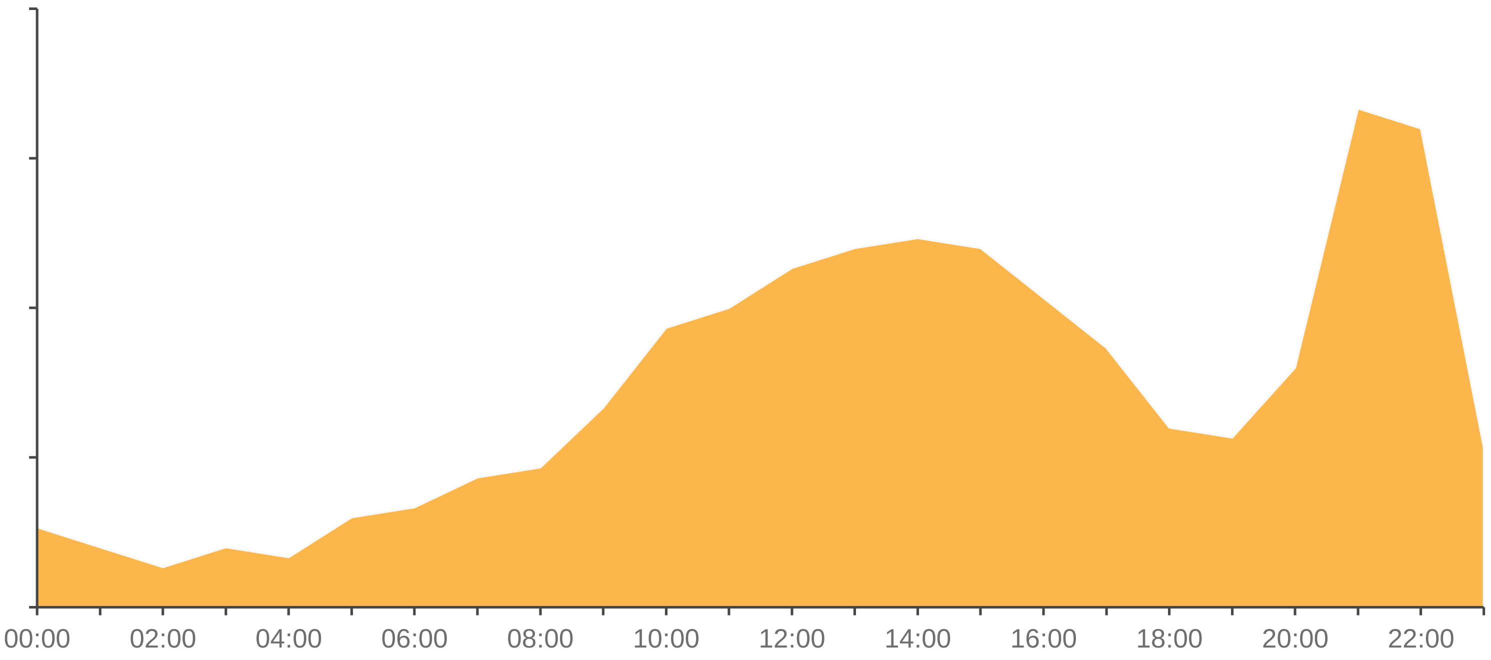
Auto Scaling Policy

Parameters for performing an Auto Scaling action

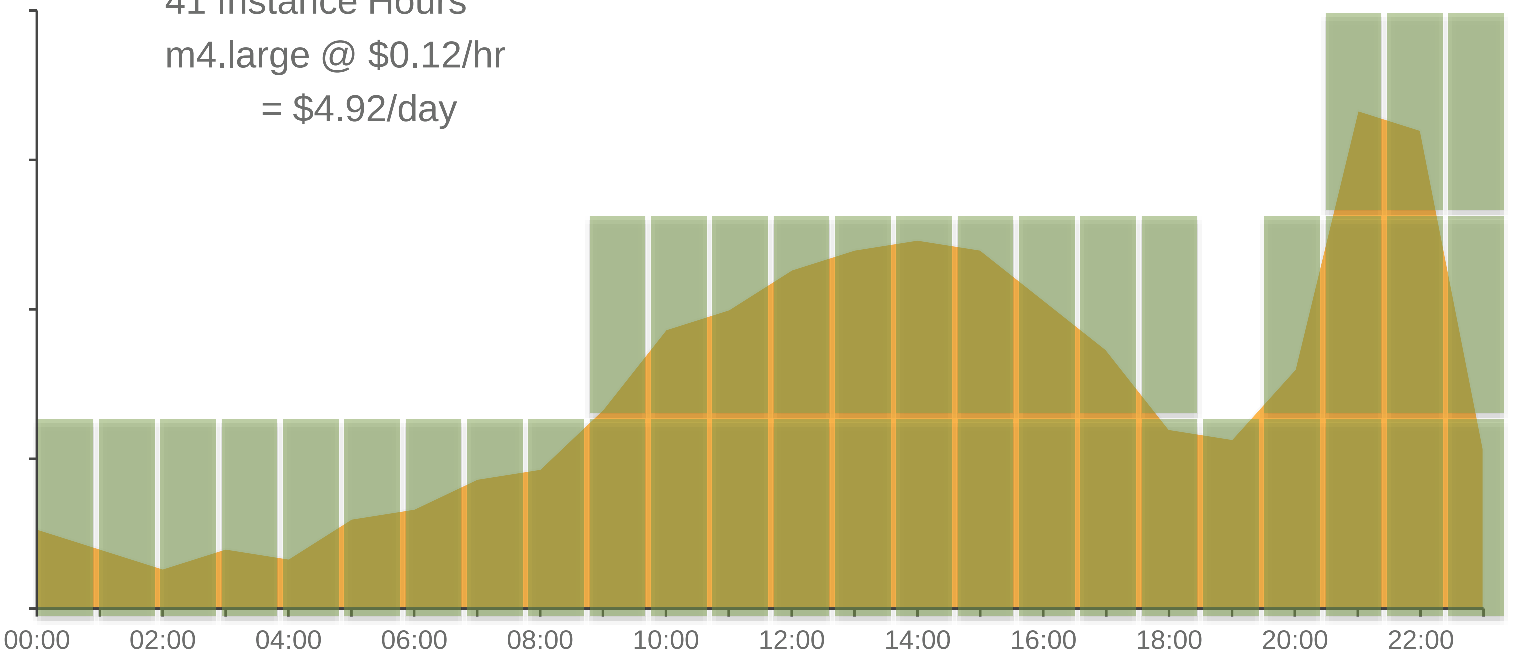
Scale Up/Down and by how much

```
aws autoscaling put-scaling-policy
   --auto-scaling-group-name autoscaling-group
   --policy-name autoscaling-policy
   --min-adjustment-magnitude=2
   --adjustment-type ChangeInCapacity
   --cooldown 300
```

Utilization & Auto Scaling Granularity



Utilization & Auto Scaling Granularity



41 Instance Hours

m4.large @ $0.12/hr

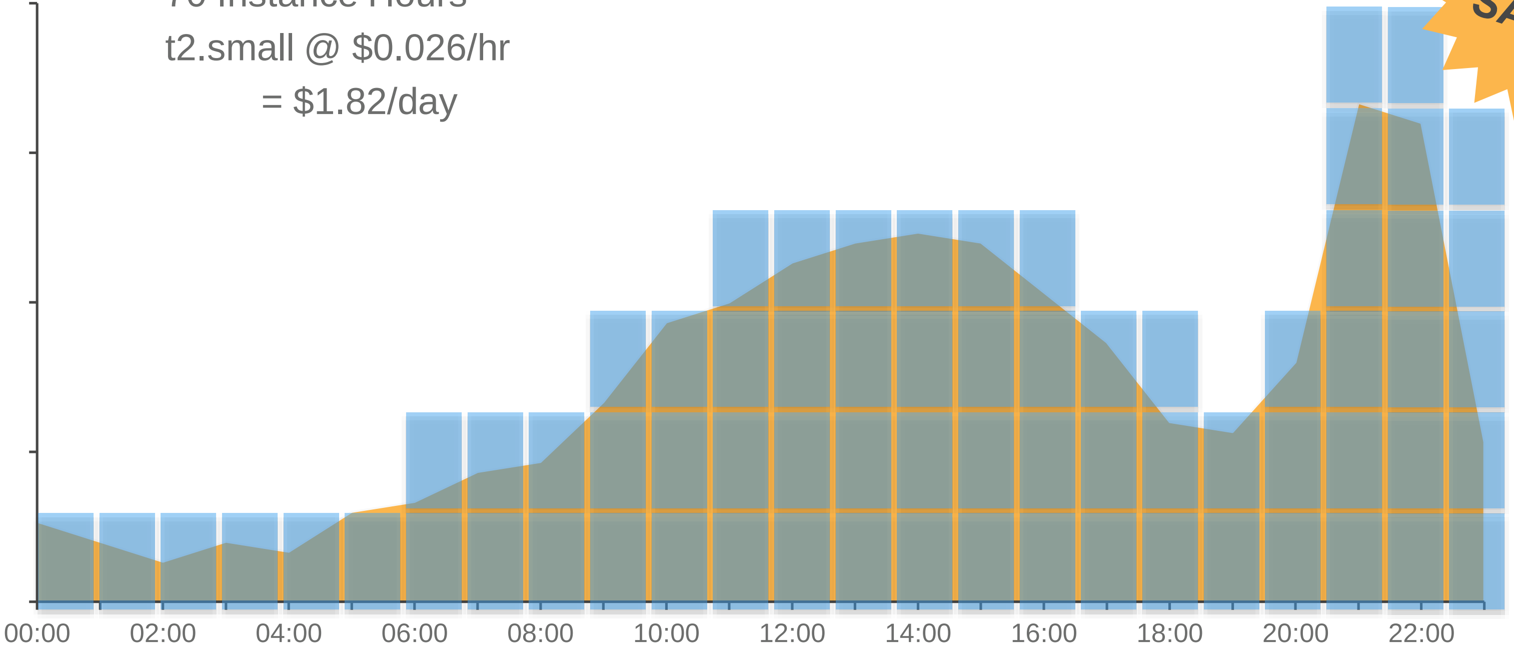
= $4.92/day

Utilization & Auto Scaling Granularity

70 Instance Hours

t2.small @ $0.026/hr

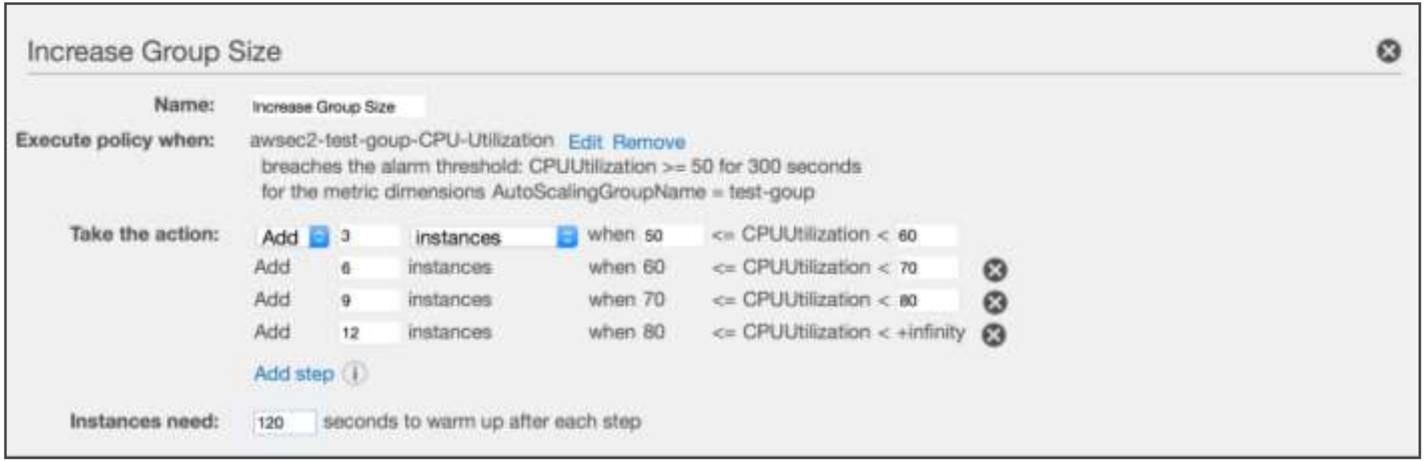
= $1.82/day



63%

SAVING

New Scaling Policies for More Responsive Scaling



Increase Group Size

Name: Increase Group Size

Execute policy when: awsec2-test-goup-CPU-Utilization
breaches the alarm threshold: CPUUtilization >= 50 for 300 seconds
for the metric dimensions AutoScalingGroupName = test-goup

Take the action:

Add	3	instances	when 50	<= CPUUtilization <	60	
Add	6	instances	when 60	<= CPUUtilization <	70	⊗
Add	9	instances	when 70	<= CPUUtilization <	80	⊗
Add	12	instances	when 80	<= CPUUtilization <	+infinity	⊗

Instances need: 120 seconds to warm up after each step

aws.amazon.com/blogs/aws/auto-scaling-update-new-scaling-policies-for-more-responsive-scaling

Cost Optimization

On-Demand Instances

- Pay for compute capacity by the hour with no long-term commitments or upfront payments

Reserved Instances

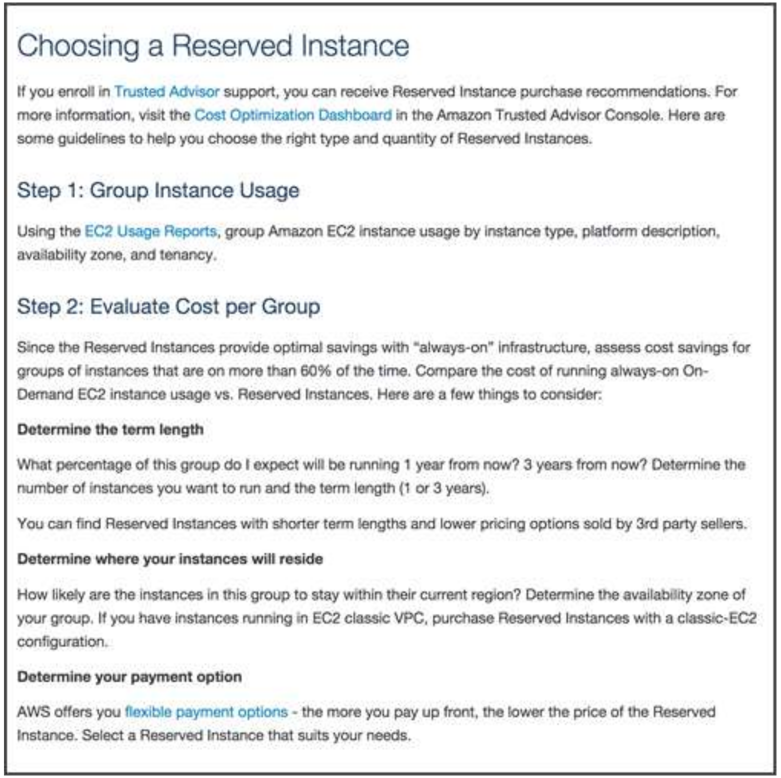
- Provide you with a significant discount (up to 75%) compared to On-Demand Instance pricing

Spot Instances

- Purchase compute capacity with no upfront commitment and at hourly rates usually lower than the On-Demand rate

aws.amazon.com/ec2/purchasing-options/

Getting Started with Reserved Instances



Choosing a Reserved Instance

If you enroll in Trusted Advisor support, you can receive Reserved Instance purchase recommendations. For more information, visit the Cost Optimization Dashboard in the Amazon Trusted Advisor Console. Here are some guidelines to help you choose the right type and quantity of Reserved Instances.

Step 1: Group Instance Usage

Using the EC2 Usage Reports, group Amazon EC2 instance usage by instance type, platform description, availability zone, and tenancy.

Step 2: Evaluate Cost per Group

Since the Reserved Instances provide optimal savings with "always-on" infrastructure, assess cost savings for groups of instances that are on more than 60% of the time. Compare the cost of running always-on On-Demand EC2 instance usage vs. Reserved Instances. Here are a few things to consider:

Determine the term length

What percentage of this group do I expect will be running 1 year from now? 3 years from now? Determine the number of instances you want to run and the term length (1 or 3 years).

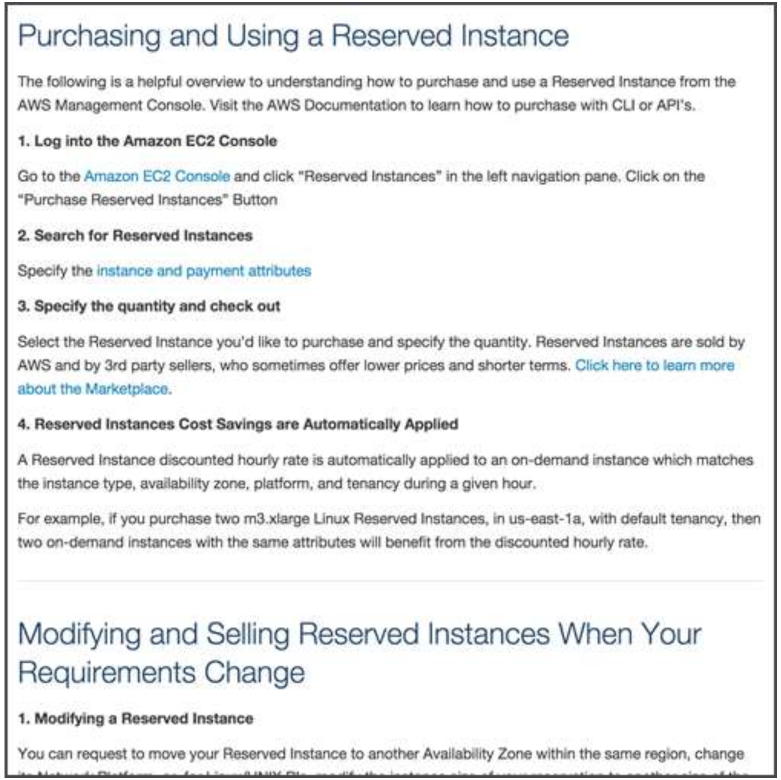
You can find Reserved Instances with shorter term lengths and lower pricing options sold by 3rd party sellers.

Determine where your instances will reside

How likely are the instances in this group to stay within their current region? Determine the availability zone of your group. If you have instances running in EC2 classic VPC, purchase Reserved Instances with a classic-EC2 configuration.

Determine your payment option

AWS offers you flexible payment options - the more you pay up front, the lower the price of the Reserved Instance. Select a Reserved Instance that suits your needs.



Purchasing and Using a Reserved Instance

The following is a helpful overview to understanding how to purchase and use a Reserved Instance from the AWS Management Console. Visit the AWS Documentation to learn how to purchase with CLI or API's.

1. Log into the Amazon EC2 Console

Go to the Amazon EC2 Console and click "Reserved Instances" in the left navigation pane. Click on the "Purchase Reserved Instances" Button

2. Search for Reserved Instances

Specify the instance and payment attributes

3. Specify the quantity and check out

Select the Reserved Instance you'd like to purchase and specify the quantity. Reserved Instances are sold by AWS and by 3rd party sellers, who sometimes offer lower prices and shorter terms. Click here to learn more about the Marketplace.

4. Reserved Instances Cost Savings are Automatically Applied

A Reserved Instance discounted hourly rate is automatically applied to an on-demand instance which matches the instance type, availability zone, platform, and tenancy during a given hour.

For example, if you purchase two m3.xlarge Linux Reserved Instances, in us-east-1a, with default tenancy, then two on-demand instances with the same attributes will benefit from the discounted hourly rate.

Modifying and Selling Reserved Instances When Your Requirements Change

1. Modifying a Reserved Instance

You can request to move your Reserved Instance to another Availability Zone within the same region, change

aws.amazon.com/ec2/purchasing-options/reserved-instances/getting-started/

Spot Instances

Spot Instances are spare Amazon EC2 instances that you can bid on.

The Spot price fluctuates in real-time based on supply and demand.

When your bid exceeds the Spot Price and spot capacity is available, your Spot instance is launched and will run until the Spot market price exceeds your bid (a Spot interruption).

aws.amazon.com/ec2/purchasing-options/spot-instances/

Useful Resources

aws.amazon.com/ec2/

Thank you!

Ian Massingham

Chief Evangelist (EMEA)

Amazon Web Services

@IanMmmm

Adam Boeglin

Solutions Architect

Amazon Web Services

Questions

Adam Boeglin,
Solutions Architect,
Amazon Web Services