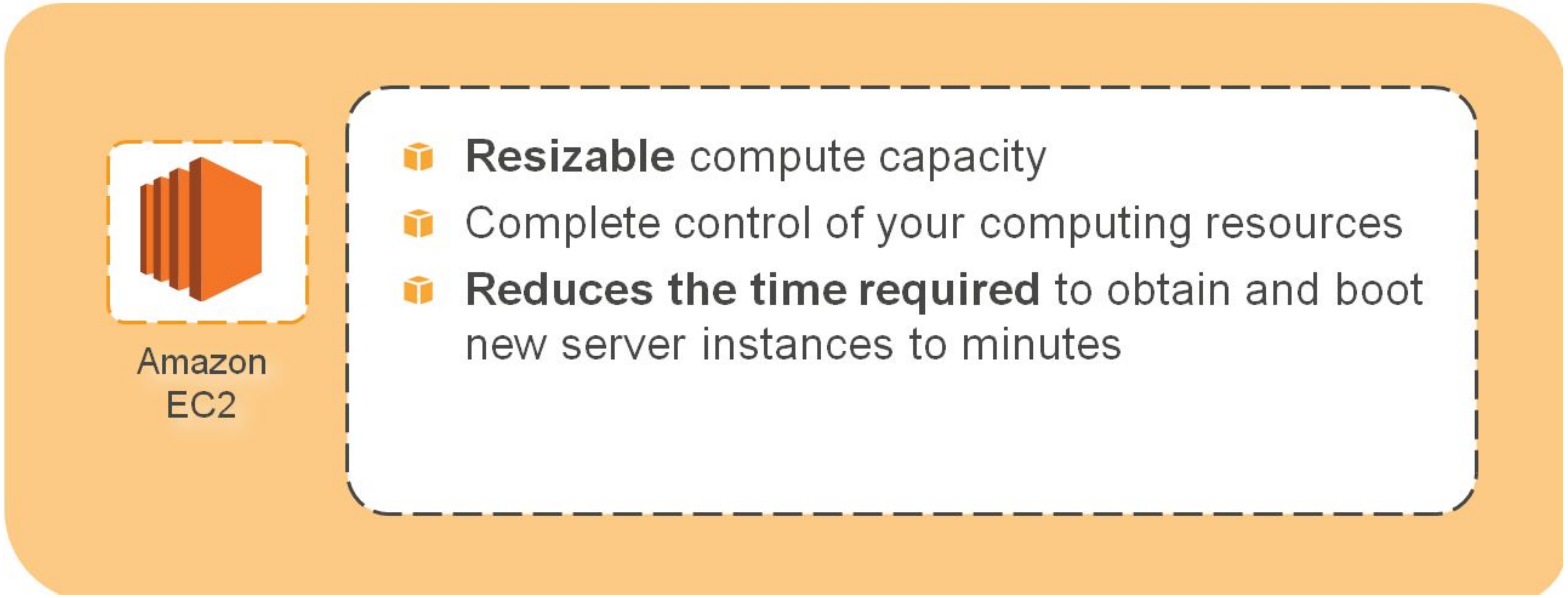
Amazon Elastic Compute Cloud (EC2)

March 2018

What is EC2 ?

- **Virtual Machines / Servers** in Cloud



- **Resizable** compute capacity
- Complete control of your computing resources
- **Reduces the time required** to obtain and boot new server instances to minutes

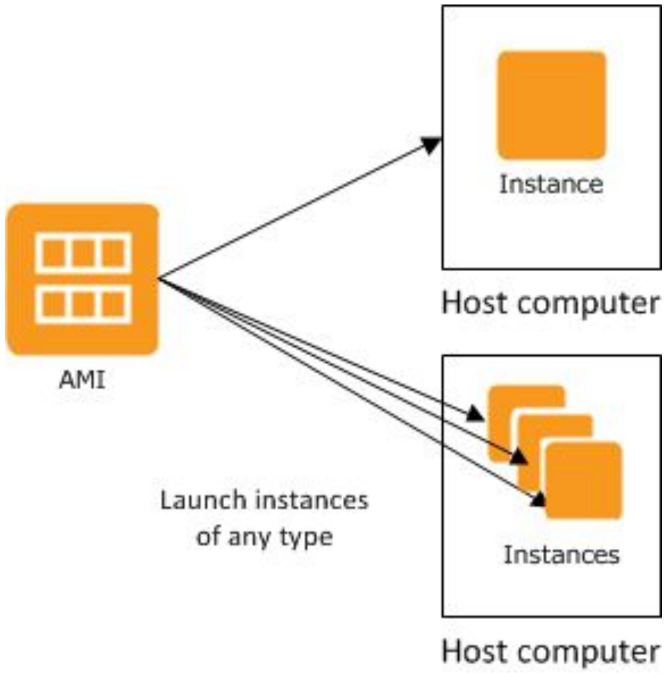
EC2 Key Concepts…

- **Amazon Machine Image – AMI –**
  - Template containing software configuration (Operating System, Application Server, Applications)

- **Instance** – Launched from AMI

- **Instance Type** – CPU /Memory/Network Performance

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



EC2 Key Concepts

- **Security Group**
  - Virtual Firewall that controls the traffic for one or more instances
  - Inbound Rules
  - Outbound Rules

- **Key Pair**
  - Public key cryptography to encrypt and decrypt login information
  - Public key to encrypt password
  - Private key to decrypt password
  - Public and Private keys are known as key-pair
  - AWS Instance stores Public key
  - If you lose your private key, there is no way to recover it

- **Elastic IP**
  - Static PUBLIC IP

Launching an EC2 Instance

1. Determine the AWS Region in which you want to launch the Amazon EC2 instance.

2. Launch an Amazon EC2 instance from a preconfigured Amazon Machine Image (AMI).

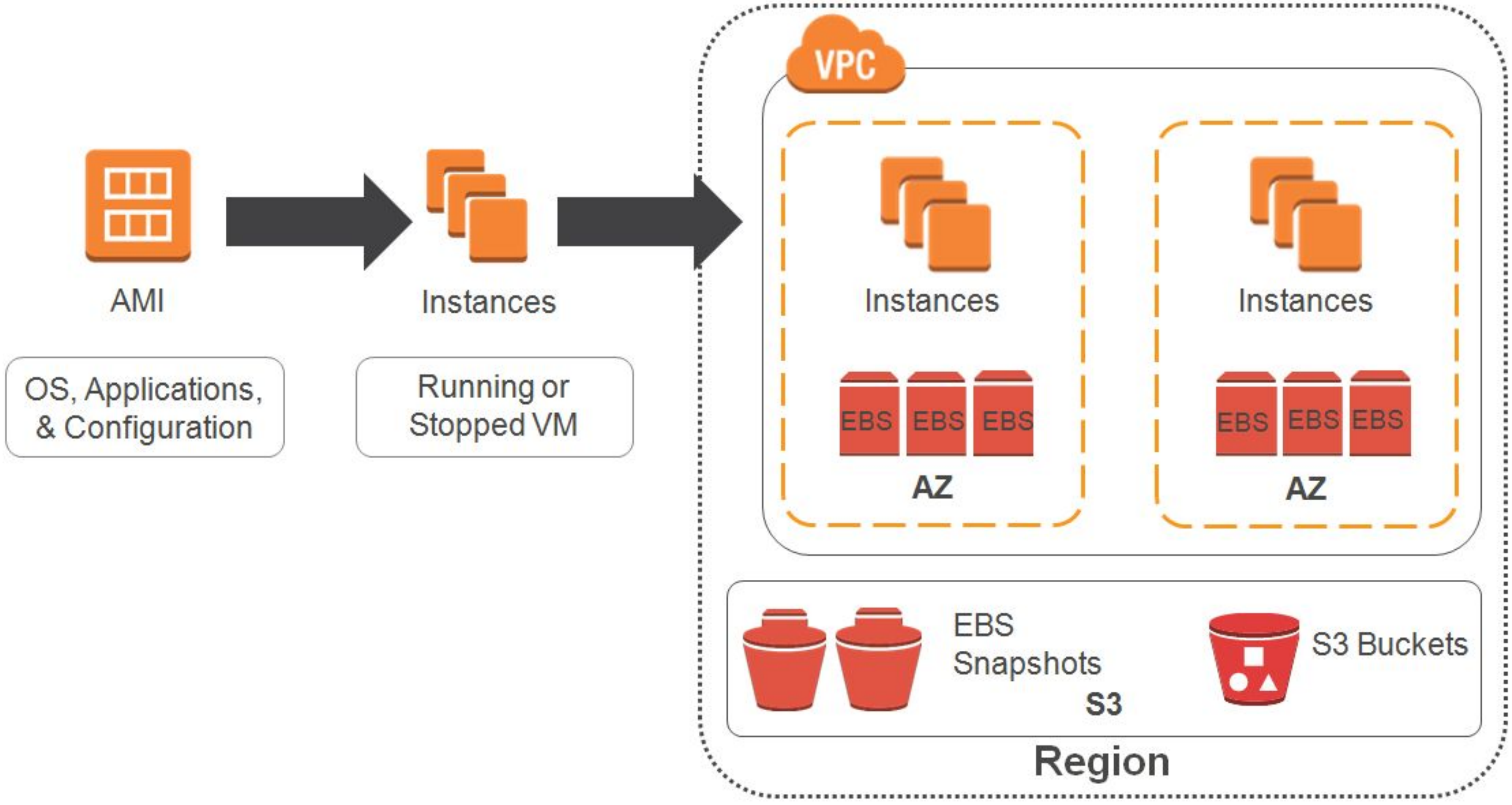
3. Choose an instance type based on CPU, memory, storage, and network requirements.

4. Configure network, IP address, security groups, storage volume, tags, and key pair.

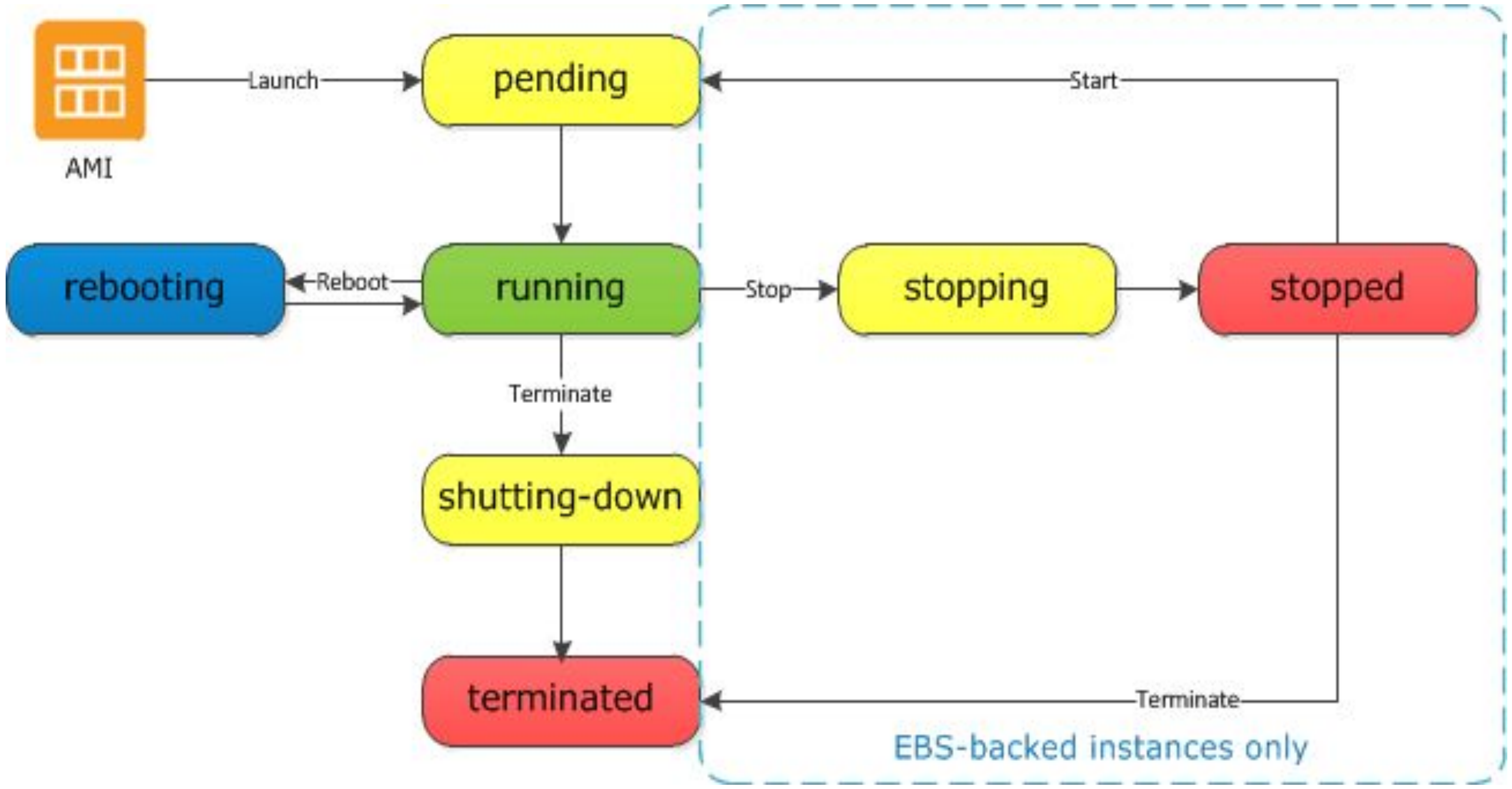
LAB – Launch EC2 Instance

- Login to console
- Create Linux/Windows Instance from AMI
- Finding your instance
- Connect to Instance
- Stop the Instance

Instances in AWS



Instance Lifecycle



Instance Lifecycle States

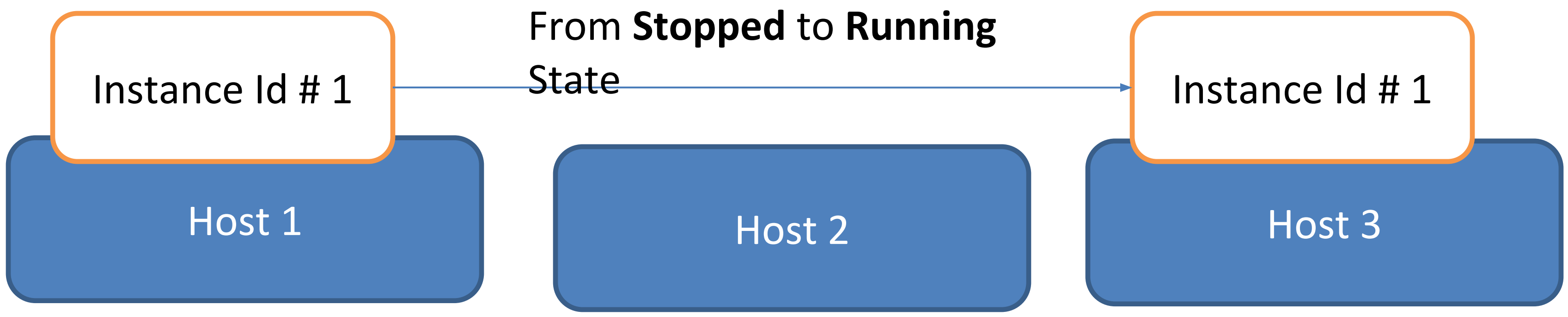
Instance Retirement

- Scheduled to be retired when AWS detects irreparable failure of the underlying hardware hosting the instance
- Either stopped or terminated by AWS

Instance Termination

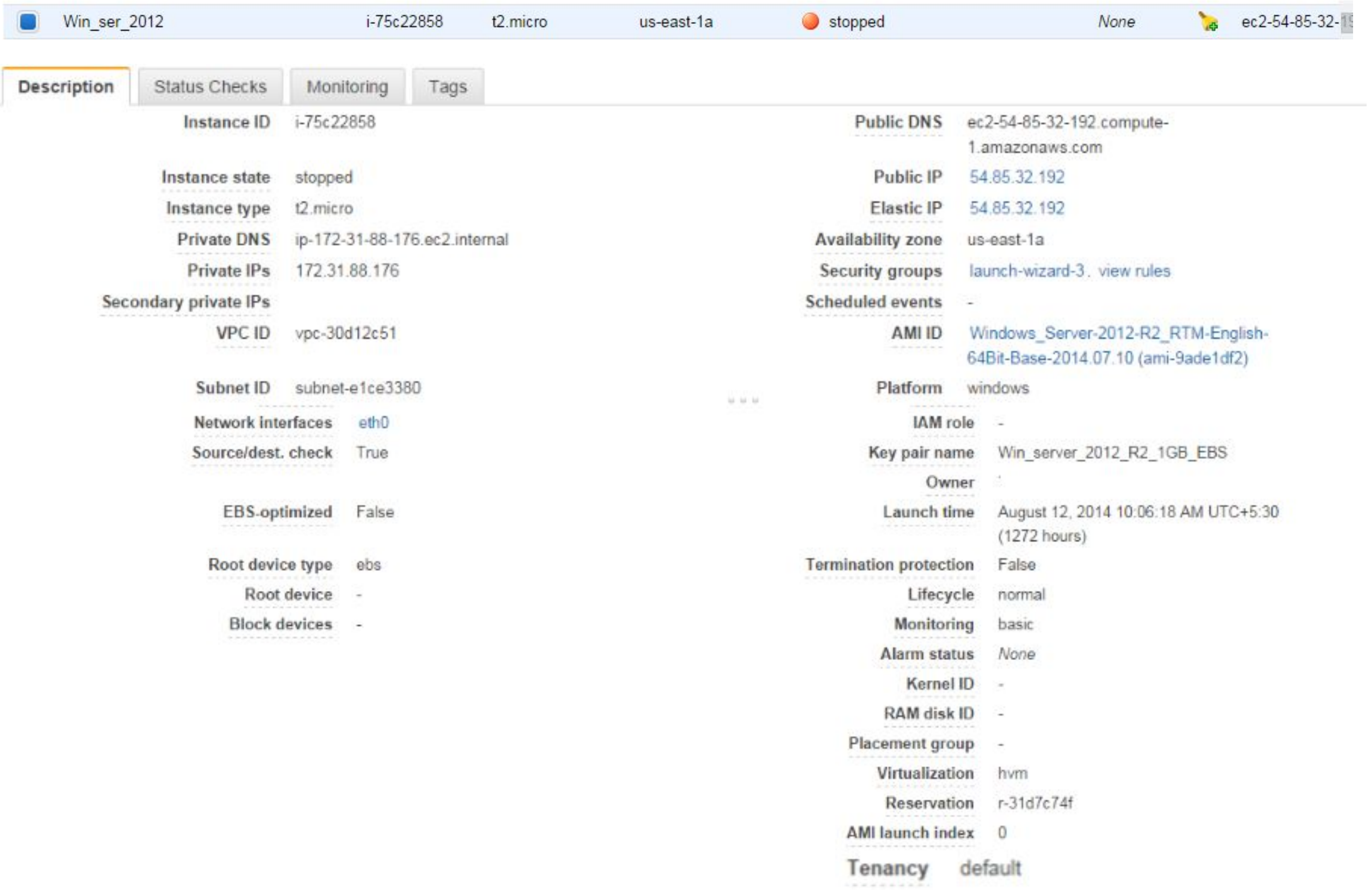
- Cannot connect to terminated instance
- Cannot recover from terminated instance
- Instance Termination Protection

Instance Changing the host computer



- Loose data on ephemeral storage /instance store volume
- Different PUBLIC IP
- Different private IP in ec2-classic
- Retains same private IP in VPC

Understanding EC2 Instance Details



■	Win_ser_2012	i-75c22858	t2.micro	us-east-1a	● stopped	None	ec2-54-85-32-

Instance ID	i-75c22858
Instance state	stopped
Instance type	t2.micro
Private DNS	ip-172-31-88-176.ec2.internal
Private IPs	172.31.88.176
Secondary private IPs	
VPC ID	vpc-30d12c51
Subnet ID	subnet-e1ce3380
Network interfaces	eth0
Source/dest. check	True
EBS-optimized	False
Root device type	ebs
Root device	-
Block devices	-

Public DNS	ec2-54-85-32-192.compute-1.amazonaws.com
Public IP	54.85.32.192
Elastic IP	54.85.32.192
Availability zone	us-east-1a
Security groups	launch-wizard-3. view rules
Scheduled events	-
AMI ID	Windows_Server-2012-R2_RTM-English-64Bit-Base-2014.07.10 (ami-9ade1df2)
Platform	windows
IAM role	-
Key pair name	Win_server_2012_R2_1GB_EBS
Owner	
Launch time	August 12, 2014 10:06:18 AM UTC+5:30 (1272 hours)
Termination protection	False
Lifecycle	normal
Monitoring	basic
Alarm status	None
Kernel ID	-
RAM disk ID	-
Placement group	-
Virtualization	hvm
Reservation	r-31d7c74f
AMI launch index	0
Tenancy	default

IP Addresses

- Public IP - 54.85.32.192
- Public DNS - *ec2-54-85-32-192.compute-1.amazonaws.com*
- Elastic IP - 54.85.32.192 – Account Specific
- Private IP - *172.31.88.176*
- Private DNS - *ip-172-31-88-176.ec2.internal*
- Secondary Private IP – Instance can have multiple IP addresses
    - Host multiple websites on a single server
    - Operate network appliance
    - Redirect Internal Traffic to standby instance
    - Management Network

Source/Destination Check

- Indicates whether source/destination checks are performed
- Instance must be the source or destination of any traffic it sends or receives
- Used for **Network Address Translation (NAT)** instance
- Private Subnets communicate to NAT providing its IP
- NAT Instances uses translation and sends the request on behalf of original request

Other Attributes

Owner – 232079927121 – Account Id

Launch Time – The time when instance got STARTED **September 12, 2014 6:28:25 PM UTC+5:30 (529 hours)**

Placement Group - logical grouping of instances within a Single Availability Zone

Monitoring

Alarm Status

Lifecycle - Normal or Spot

Scheduled Events

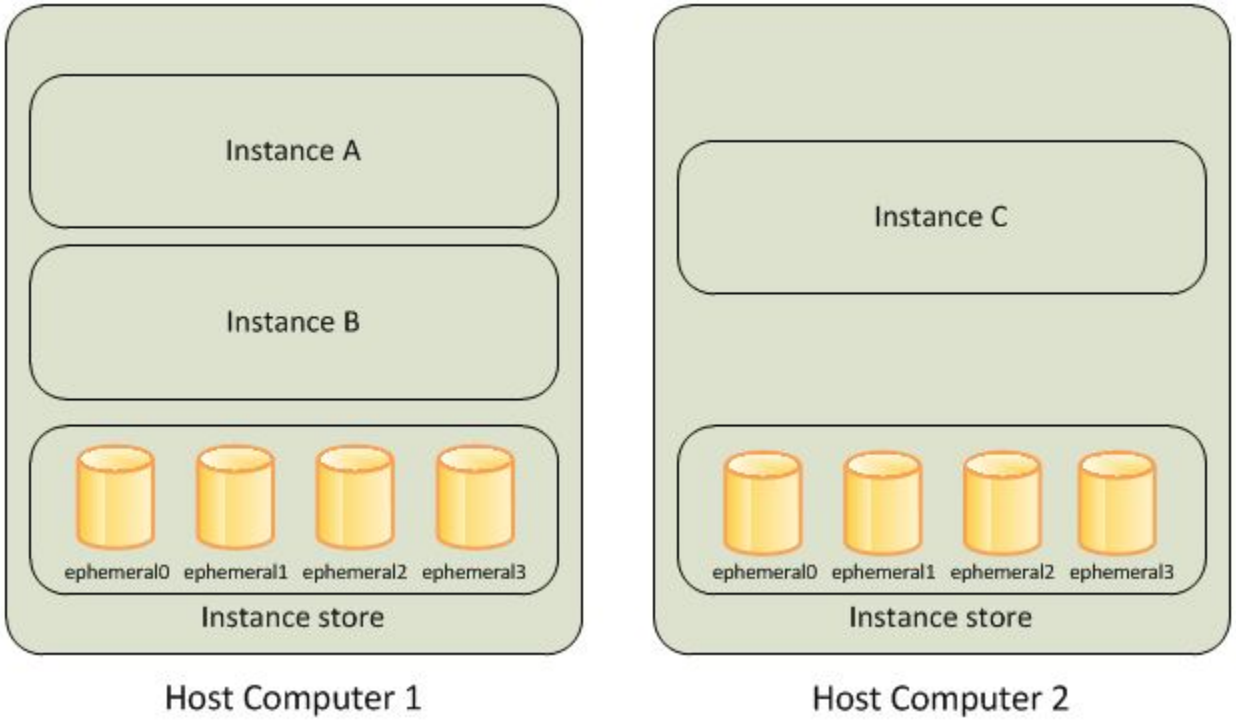
- The number of scheduled events associated with this instance, if applicable
- Types of Scheduled Events
  - Reboot
  - System Maintenance
    - Network
    - Power
  - Instance Retirement
  - Instance Stop

Storage with EC2

Instance Store

- An *instance store* provides temporary block-level storage for your instance
- Storage is located on disks which is physically attached to underlying host
- Ideal for temporary storage of information that changes frequently
  - 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.

Instance Store on Host Computer



Amazon EBS vs. Amazon EC2 Instance Store

Amazon EBS

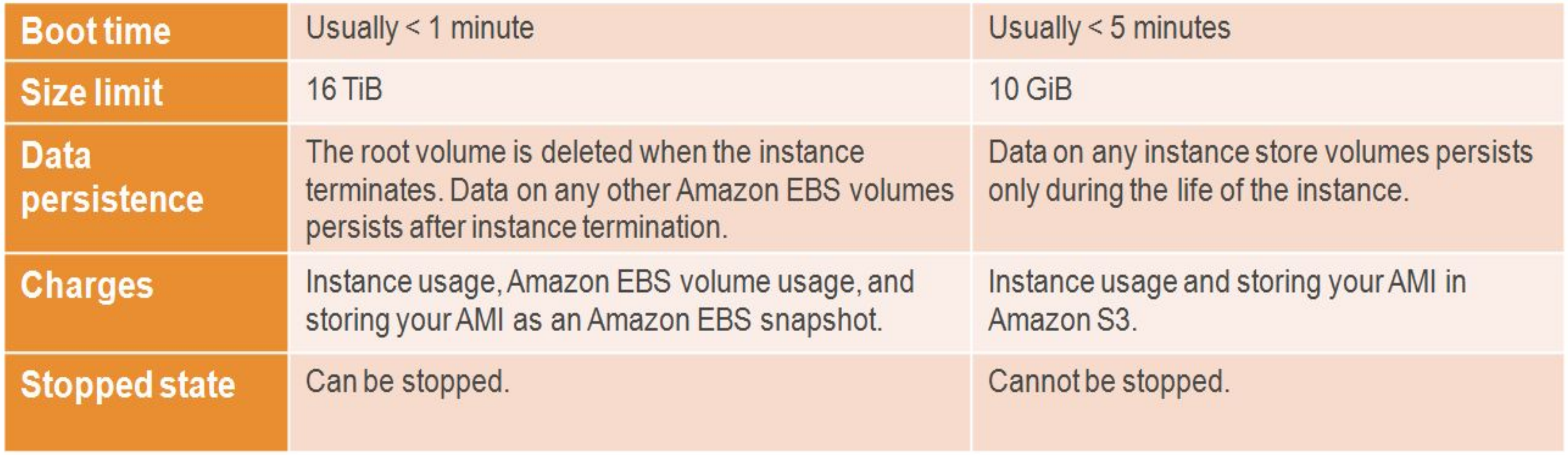
- Data stored on an Amazon EBS volume can persist independently of the life of the instance.
- Storage is persistent.

Amazon EC2 Instance Store

- Data stored on a local instance store persists only as long as the instance is alive.

- Storage is ephemeral

AMI Types - Storage for the Root Device

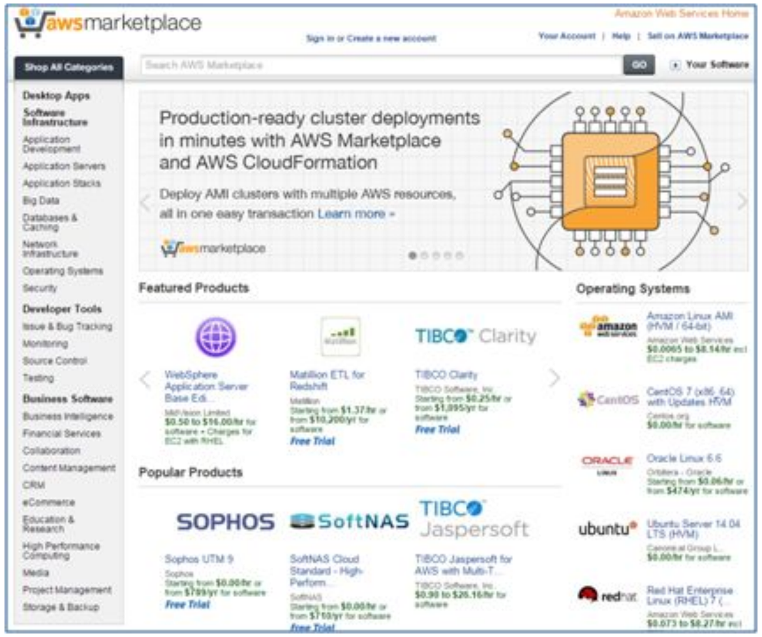


Boot time	Usually < 1 minute	Usually < 5 minutes
Size limit	16 TiB	10 GiB
Data persistence	The root volume is deleted when the instance terminates. Data on any other Amazon EBS volumes persists after instance termination.	Data on any instance store volumes persists only during the life of the instance.
Charges	Instance usage, Amazon EBS volume usage, and storing your AMI as an Amazon EBS snapshot.	Instance usage and storing your AMI in Amazon S3.
Stopped state	Can be stopped.	Cannot be stopped.

AWS MarketPlace - IT Software Optimized for Cloud

Online store to discover, purchase, and deploy IT software on top of the AWS infrastructure.

- Catalog of 2300+ IT software solutions
- Including Paid, BYOL, Open Source, SaaS, & free to try options
- Pre-configured to operate on AWS
- Software checked by AWS for security and operability
- Deploys to AWS environment in minutes
- Flexible, usage-based billing models
- Software charges billed to AWS account
- Includes AWS Test Drive.
- https://aws.amazon.com/marketplace



Production-ready cluster deployments in minutes with AWS Marketplace and AWS CloudFormation

Deploy AMI clusters with multiple AWS resources, all in one easy transaction Learn more »

Featured Products

WebSphere Application Server Base Edi...
MidVision Limited
$0.50 to $16.00/hr for software + Charges for EC2 with RHEL

Matillion ETL for Redshift
Matillion
Starting from $1.37/hr or from $10,200/yr for software
Free Trial

TIBCO Clarity
TIBCO Software, Inc.
Starting from $0.25/hr or from $1,095/yr for software
Free Trial

Popular Products

Sophos UTM 9
Sophos
Starting from $0.00/hr or from $799/yr for software
Free Trial

SoftNAS Cloud Standard - High-Perform...
SoftNAS
Starting from $0.00/hr or from $710/yr for software
Free Trial

TIBCO Jaspersoft for AWS with Multi-T...
TIBCO Software, Inc.
$0.90 to $26.16/hr for software

Operating Systems

Amazon Linux AMI (HVM / 64-bit)
Amazon Web Services
$0.0065 to $8.14/hr excl EC2 charges

CentOS 7 (x86_64) with Updates HVM
Centos.org
$0.00/hr for software

Oracle Linux 6.6
Orbitera - Oracle
Starting from $0.06/hr or from $474/yr for software

Ubuntu Server 14.04 LTS (HVM)
Canonical Group L...
$0.00/hr for software

Red Hat Enterprise Linux (RHEL) 7 (...
Amazon Web Services
$0.073 to $8.27/hr excl

Choosing Right type of Instance

- EC2 instance types are optimized for different use cases and come in multiple sizes. This allows you to optimally scale resources to your workload requirements.
- AWS uses Intel® Xeon® processors for EC2 instances, providing customers with high performance and value.
- Consider the following when choosing your instances:
  - Core count, memory size, storage size and type, network performance, and CPU technologies.
- Hurry Up and Go Idle - A larger compute instance can save you time and money, therefore paying more per hour for a shorter amount of time can be less expensive.

Choosing Instance Family

Instance Family	Some Use Cases
General purpose (t2, m4, m3)	• Low-traffic websites and web applications • Small databases and mid-size databases
Compute optimized (c4, c3)	• High performance front-end fleets • Video-encoding
Memory optimized (r3)	• High performance databases • Distributed memory caches
Storage optimized (i2, d2)	• Data warehousing • Log or data-processing applications
GPU instances (g2)	• 3D application streaming • Machine learning

Instance Metadata & User Data

Instance Metadata:

- Is data about your instance.
- Can be used to configure or manage a running instance.

Instance User Data:

- Can be passed to the instance at launch.
- Can be used to perform common automated configuration tasks.
- Runs scripts after the instance starts.

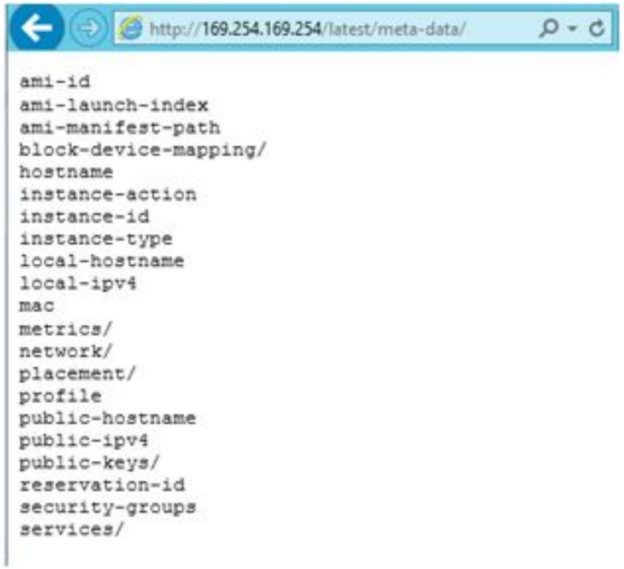
Retrieving Instance Metadata

- To view all categories of instance metadata from within a running instance, use the following URI:
  `http://169.254.169.254/latest/meta-data/`

On a Linux instance, you can use:

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

- All metadata is returned as text (content type text/plain



```
ami-id
ami-launch-index
ami-manifest-path
block-device-mapping/
hostname
instance-action
instance-id
instance-type
local-hostname
local-ipv4
mac
metrics/
network/
placement/
profile
public-hostname
public-ipv4
public-keys/
reservation-id
security-groups
services/
```

Adding User Data

- You can specify user data when launching an instance.

- User data can be:
  - Linux script – executed by **cloud-init**
  - Windows batch or PowerShell scripts – executed by **EC2Config** service

- User data scripts run once per **instance-id** by default

User Data Example Linux

- You can specify user data when launching an instance.

- User data can be:
  - Linux script – executed by **cloud-init**
  - Windows batch or PowerShell scripts – executed by **EC2Config** service

- User data scripts run once per **instance-id** by default

User Data Example - Linux

```
#!/bin/sh
```

User data shell scripts must start with the #! characters and the path to the interpreter you want to read the script.

```
yum -y install httpd
chkconfig httpd on
/etc/init.d/httpd
start
```

Install Apache web server
Enable the web server
Start the web server

User Data Example - Windows

```
<powershell>
Import-Module ServerManager

Install-WindowsFeature web-server, web-webserver
Install-WindowsFeature web-mgmt-tools
</powershell>
```

Import the Server Manager module for Windows PowerShell.

Install IIS
Install Web Management Tools

Retrieving User Data

To retrieve user data, use the following URI:

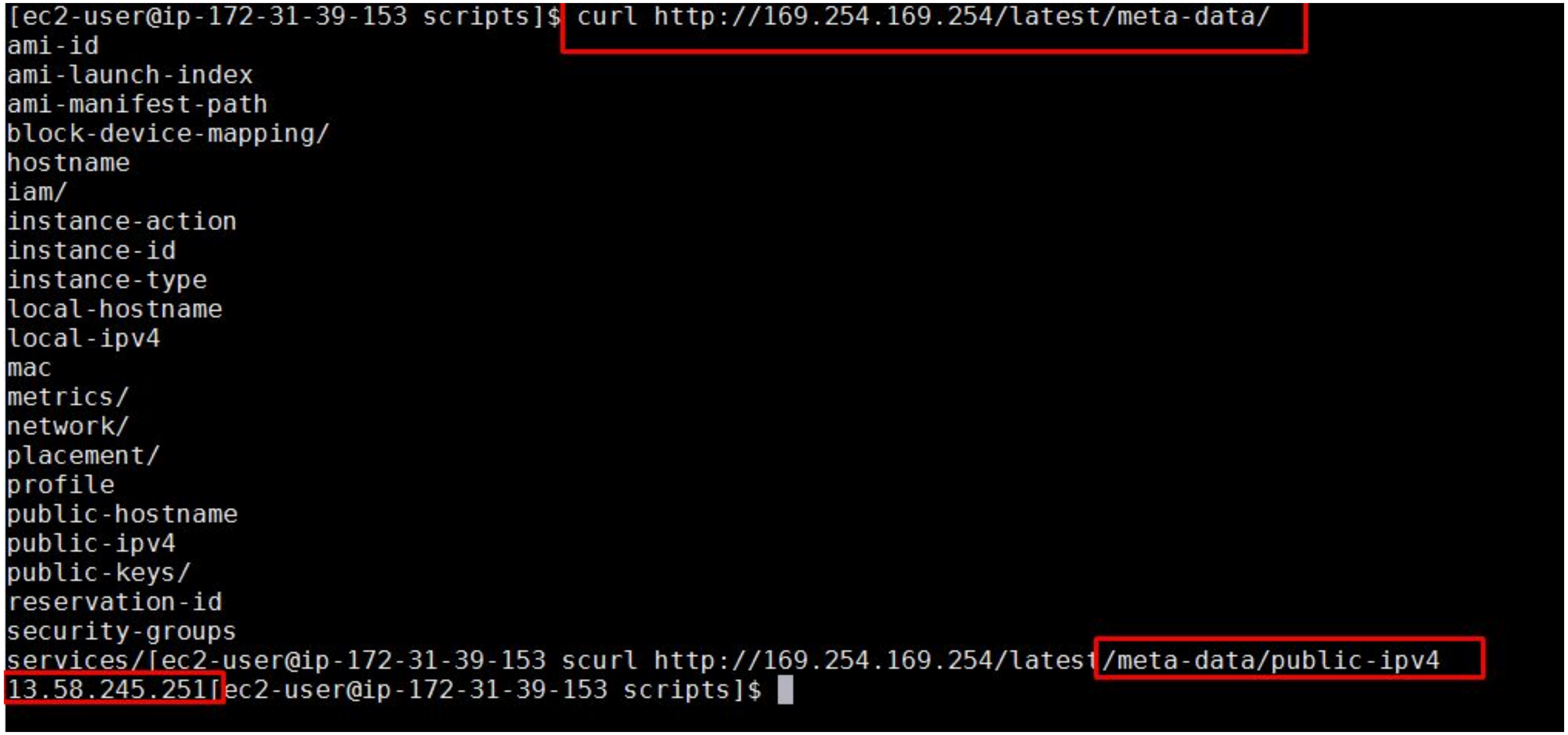
```
http://169.254.169.254/latest/user-data
```

On Linux instance, you can use:

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


$ GET http://169.254.169.254/latest/user-data/
```

Retrieving metadata values



```
[ec2-user@ip-172-31-39-153 scripts]$ curl http://169.254.169.254/latest/meta-data/
ami-id
ami-launch-index
ami-manifest-path
block-device-mapping/
hostname
iam/
instance-action
instance-id
instance-type
local-hostname
local-ipv4
mac
metrics/
network/
placement/
profile
public-hostname
public-ipv4
public-keys/
reservation-id
security-groups
services/[ec2-user@ip-172-31-39-153 scurl http://169.254.169.254/latest/meta-data/public-ipv4
13.58.245.251[ec2-user@ip-172-31-39-153 scripts]$ █
```

Purchasing Options

On-Demand Instances

Pay by the hour.

Reserved Instances

Purchase at significant discount. Instances are always available.

1-year to 3-year terms.

Scheduled Instances

Purchase a 1-year RI for a recurring period of time.

Spot Instances

Highest bidder uses instance at a significant discount.

Spot blocks supported.

Dedicated Hosts

Physical host is fully dedicated to run your instances. Bring your per-socket, per-core, or per-VM software licenses to reduce cost.

Spot Instances

What it is ?

- Bids on spare Amazon EC2 instances
- Price always less than On-Demand EC2 instance
- Prices varies based on demand
- Will not be charged for the interrupted hour
- Types of pot Instance Request
    - One time request
    - Persistent Request
        - request is opened again after your Spot instance is terminated.

How it works ?

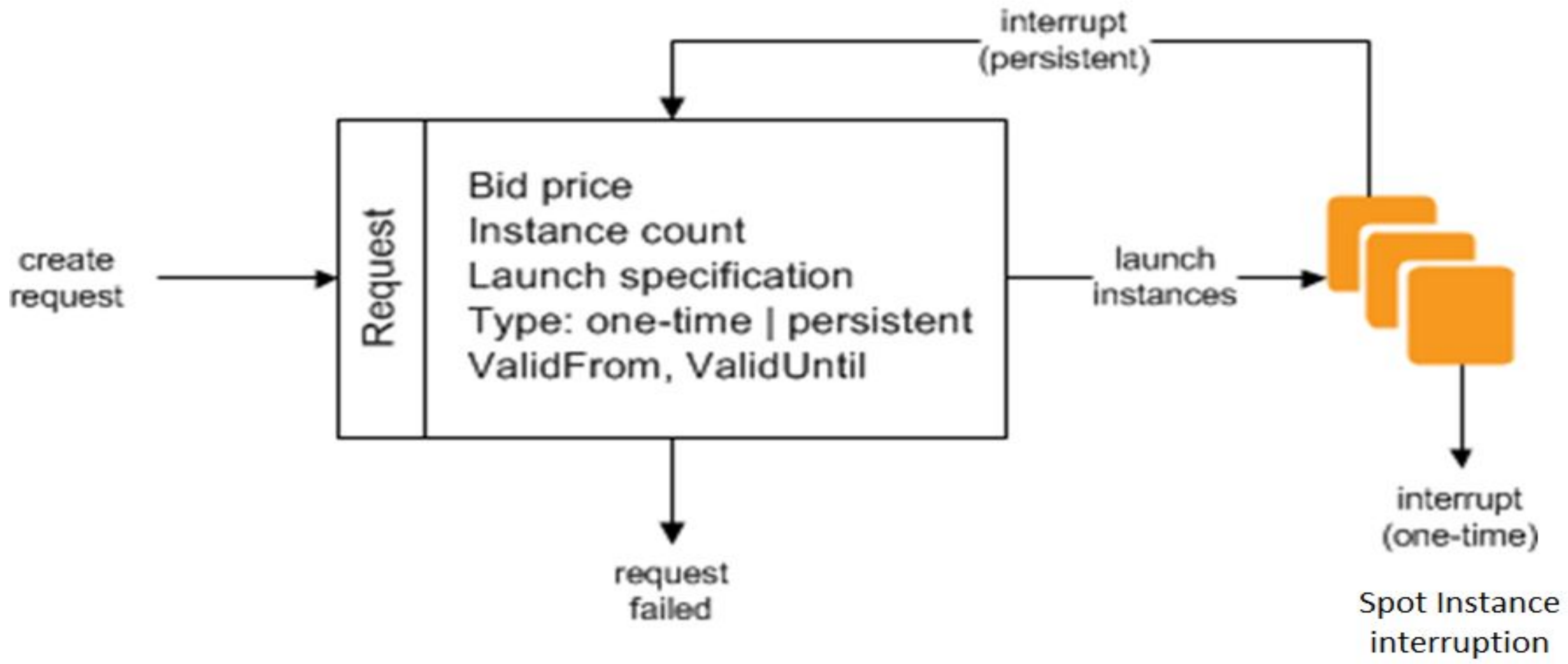
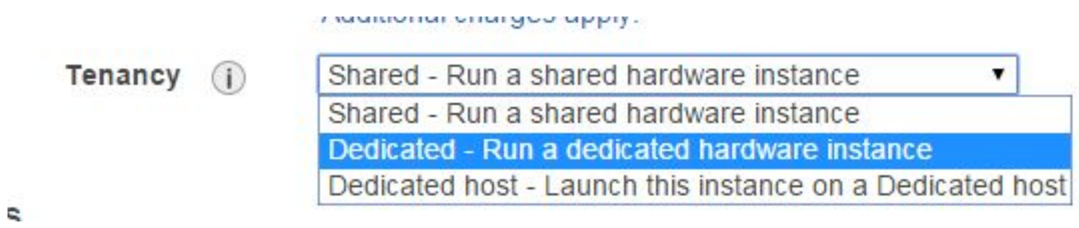


Image Source: http://docs.aws.amazon.com/AWSEC2/latest/UserGuide/how-spot-instances-work.html

When to Use ?

- Only for time-flexible, interruption-tolerant tasks
- Architect for the potential of interruption
- Large Scale processing
- Payroll processing
- PDF conversion
- Audio - Video Encoding

Dedicated Hosts - New One - Announced on Nov 23 - 2015



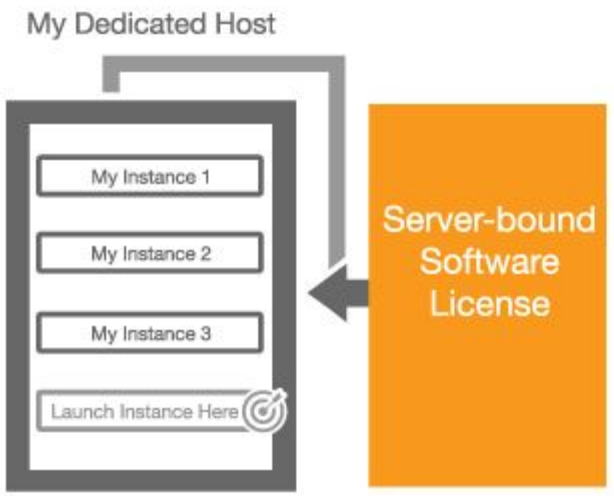
Additional charges apply.

Tenancy

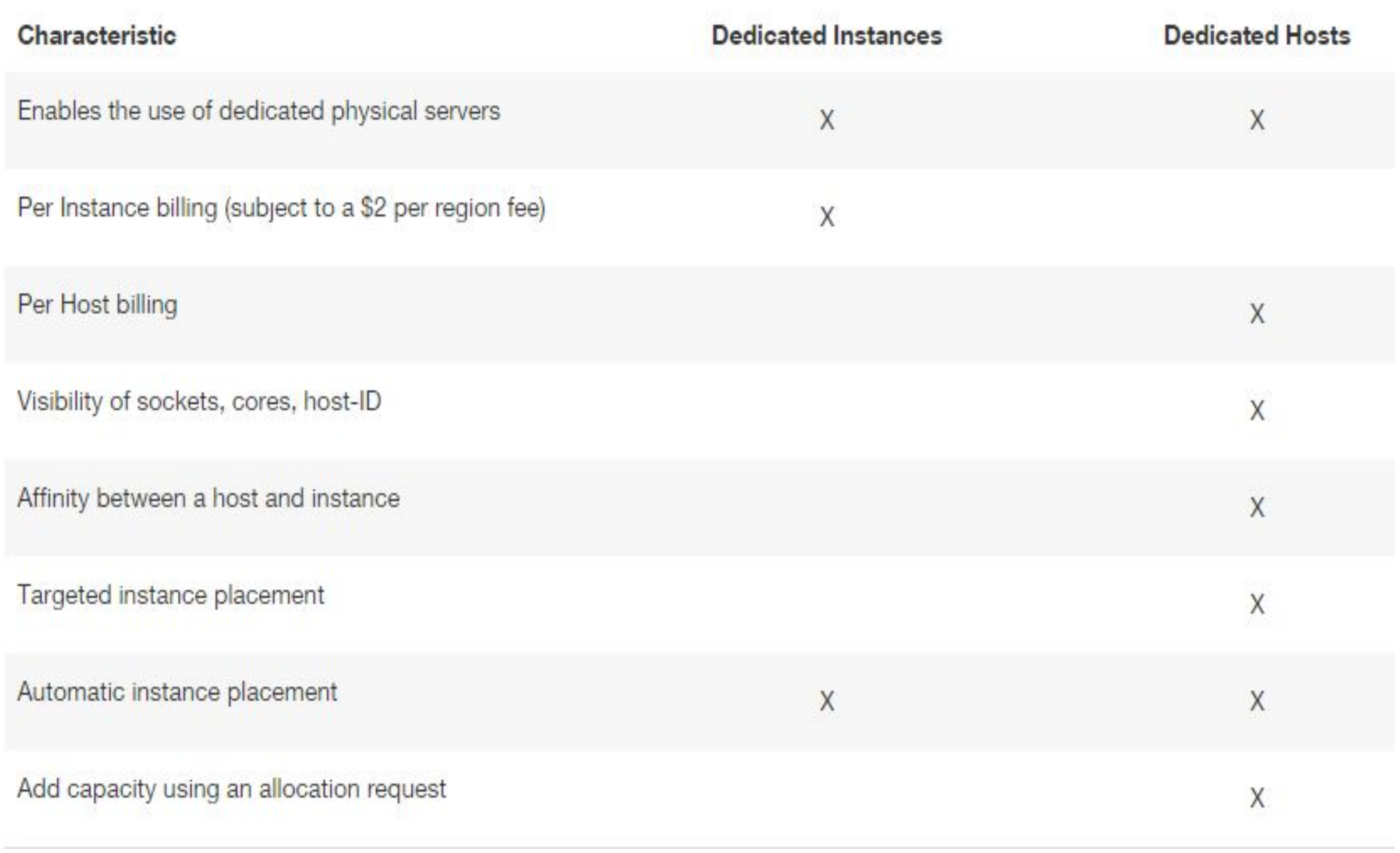
Shared - Run a shared hardware instance

- Shared - Run a shared hardware instance
- Dedicated - Run a dedicated hardware instance
- Dedicated host - Launch this instance on a Dedicated host

- Physical Server with full EC2 instance capacity dedicated to you
- Address compliance requirements
- Allows to reduce costs by allowing to use existing server bound software licences
  - per socket licence
  - per-core licence
  - per-vm licence

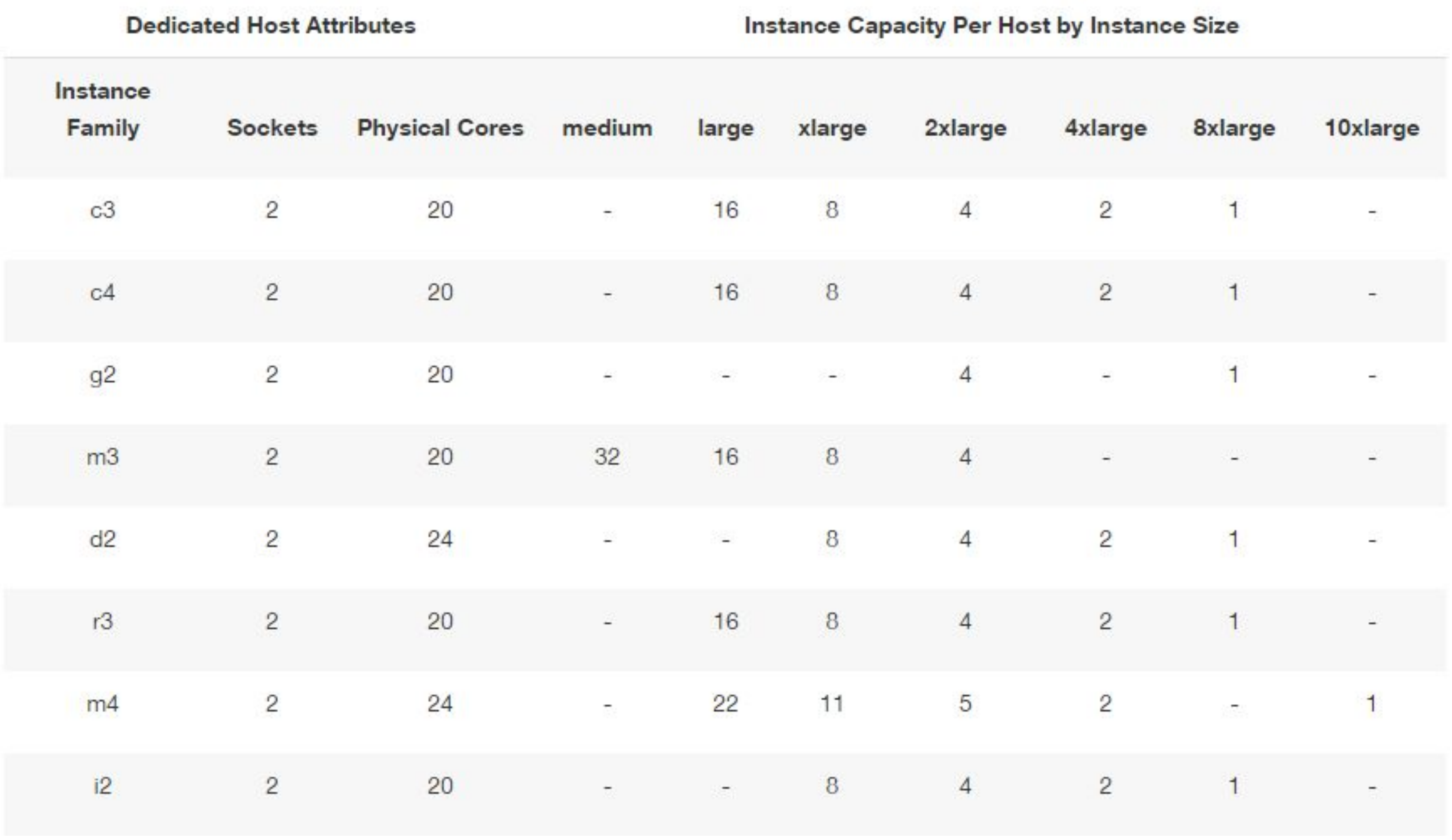


Dedicated Hosts vs. Dedicated Instances



Characteristic	Dedicated Instances	Dedicated Hosts
Enables the use of dedicated physical servers	X	X
Per Instance billing (subject to a $2 per region fee)	X	
Per Host billing		X
Visibility of sockets, cores, host-ID		X
Affinity between a host and instance		X
Targeted instance placement		X
Automatic instance placement	X	X
Add capacity using an allocation request		X

Dedicated Hosts to Instance Mapping

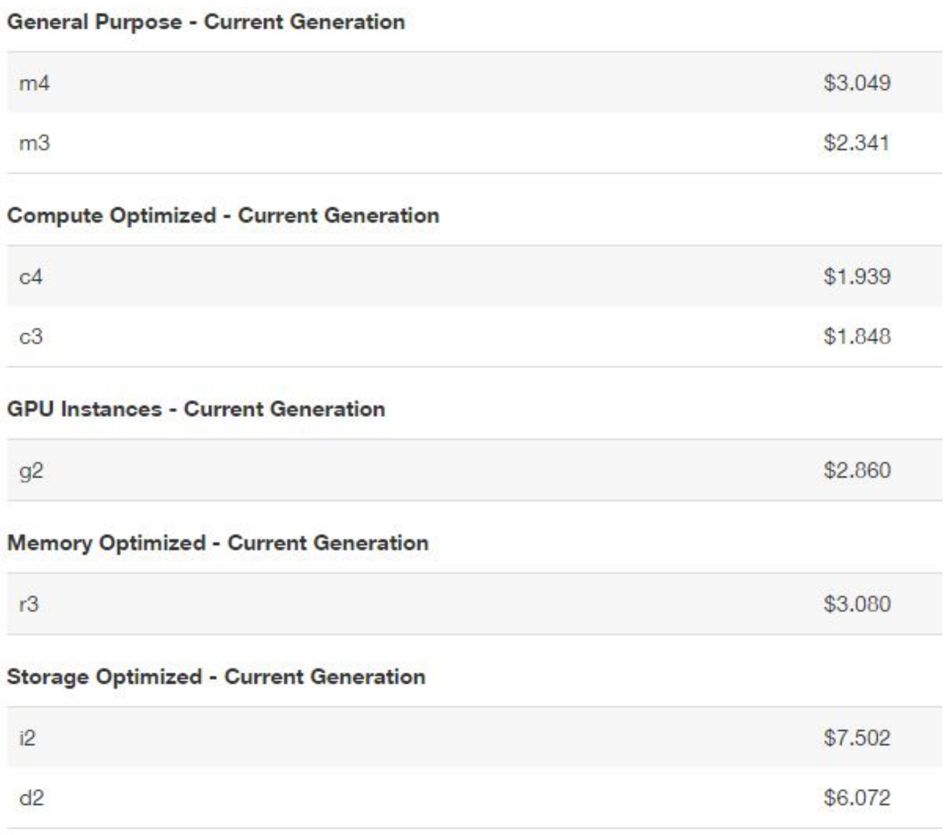


Dedicated Host Attributes			Instance Capacity Per Host by Instance Size						
Instance Family	Sockets	Physical Cores	medium	large	xlarge	2xlarge	4xlarge	8xlarge	10xlarge
c3	2	20	-	16	8	4	2	1	-
c4	2	20	-	16	8	4	2	1	-
g2	2	20	-	-	-	4	-	1	-
m3	2	20	32	16	8	4	-	-	-
d2	2	24	-	-	8	4	2	1	-
r3	2	20	-	16	8	4	2	1	-
m4	2	24	-	22	11	5	2	-	1
i2	2	20	-	-	8	4	2	1	-

- A Dedicated Host is configured to support one instance type at a time.
- e.g. if you allocate a **c3.xlarge** Dedicated Host, you use a Dedicated Host with two sockets and 20 physical cores configured to support up to **8 c3.xlarge i**nstances.

Pricing

- Price varies by
  - instance family , region and payment option
- Pay hourly
  - for each active Dedicated Host
  - No matter how many instances are launched



General Purpose - Current Generation

m4	$3.049
m3	$2.341

Compute Optimized - Current Generation

c4	$1.939
c3	$1.848

GPU Instances - Current Generation

g2	$2.860

Memory Optimized - Current Generation

r3	$3.080

Storage Optimized - Current Generation

i2	$7.502
d2	$6.072

https://aws.amazon.com/blogs/aws/now-available-ec2-dedicated-hosts/

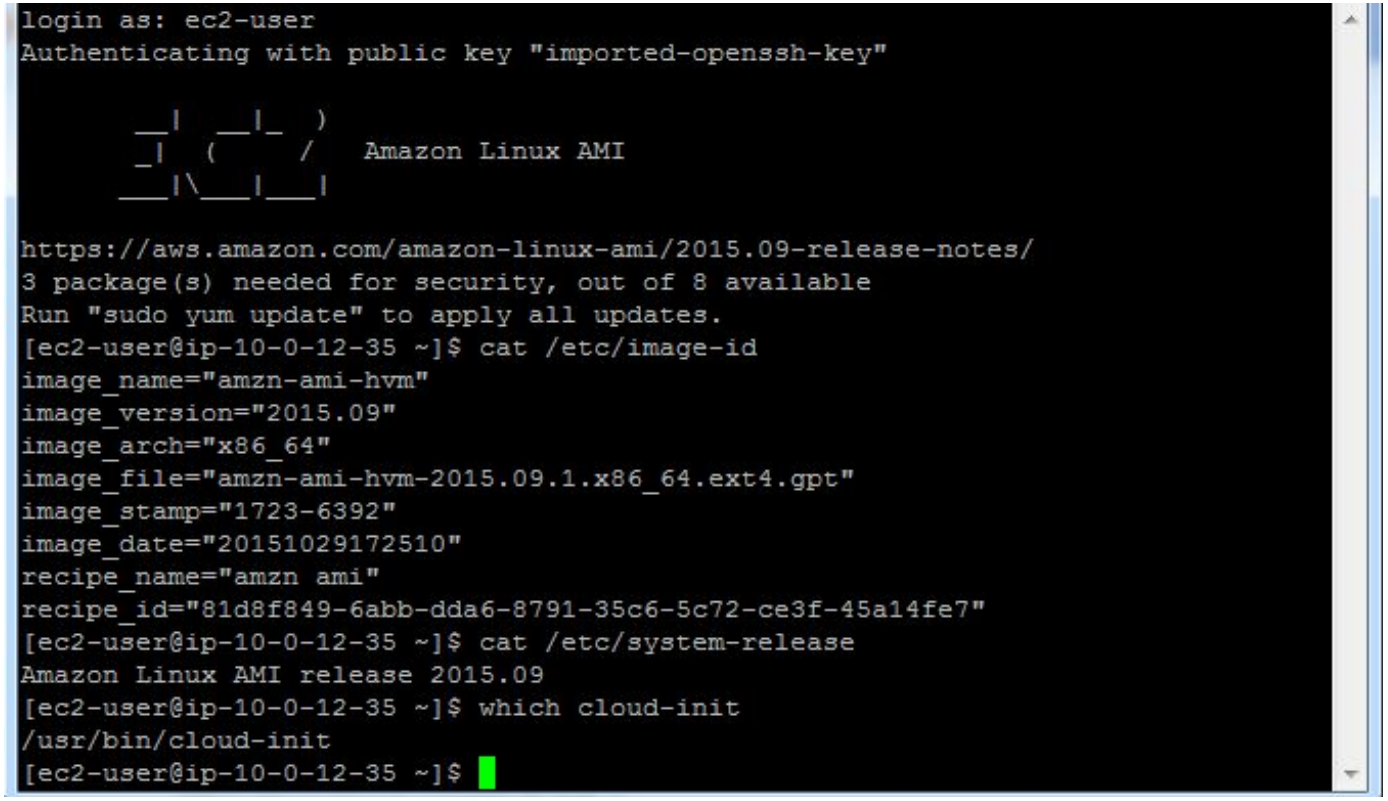
LAB – Termination Protection

- Login to console
- Enable Termination Protection
- Try to terminate

Amazon Machine Image (AMI)

Amazon Machine Image

- PreConfigured Software Template
- AMI is region specific
- Each Images has a Unique Id



```
login as: ec2-user
Authenticating with public key "imported-openssh-key"

       __|  __|_  )
       _|  (     /   Amazon Linux AMI
      ___|\___|___|

https://aws.amazon.com/amazon-linux-ami/2015.09-release-notes/
3 package(s) needed for security, out of 8 available
Run "sudo yum update" to apply all updates.
[ec2-user@ip-10-0-12-35 ~]$ cat /etc/image-id
image_name="amzn-ami-hvm"
image_version="2015.09"
image_arch="x86_64"
image_file="amzn-ami-hvm-2015.09.1.x86_64.ext4.gpt"
image_stamp="1723-6392"
image_date="20151029172510"
recipe_name="amzn ami"
recipe_id="81d8f849-6abb-dda6-8791-35c6-5c72-ce3f-45a14fe7"
[ec2-user@ip-10-0-12-35 ~]$ cat /etc/system-release
Amazon Linux AMI release 2015.09
[ec2-user@ip-10-0-12-35 ~]$ which cloud-init
/usr/bin/cloud-init
[ec2-user@ip-10-0-12-35 ~]$
```

“cloud-init” package

- Application to bootstrap Linux images in a cloud environment
- Open Source
- Developed by Canonical
- Enables to specify actions that should happen to instance at boot time
- Actions can be passed via `user-data` fields
- Use Base Image and pass dynamic data at launch time

For example :

- `action: CONFIG_SSH`
  - Generates host private SSH keys
  - Adds a user's public SSH keys to `.ssh/authorized_keys` for easy login and administration
- `action: PACKAGE_SETUP`
  - Prepares **`yum`** repo
  - Handles package actions defined in user data

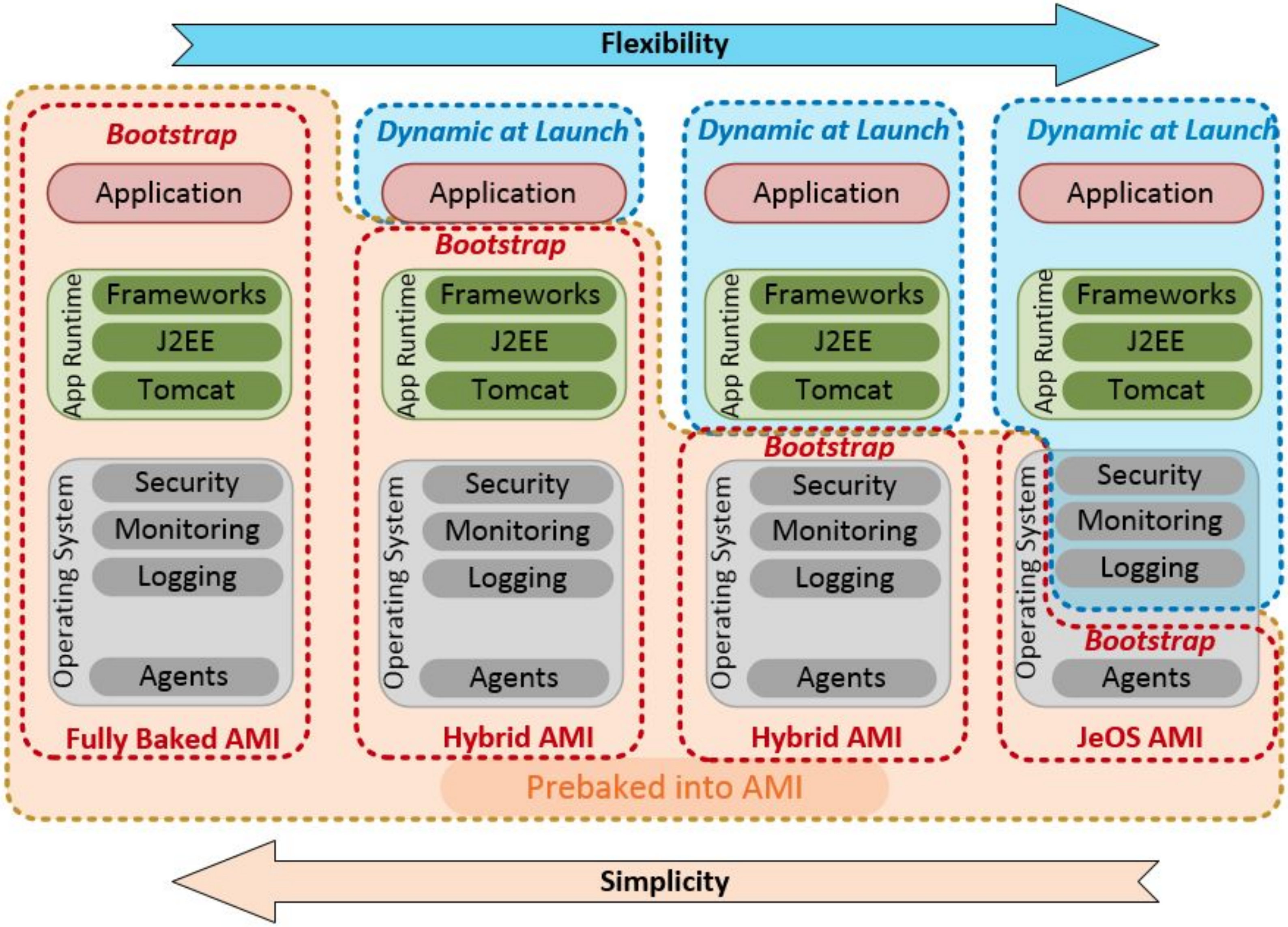
Upgrade the instance

- AMIs are configured to download and install security updates during launch time
- Controlled through ‘cloud-init’ configuration
- Possible values for repo-upgrade
  - security
  - bugfix
  - all
  - none

```
#cloud-config
repo_upgrade: security
```

Strategies for Creating Reusable EC2 Instances ? AMIs?

AMI Design Strategies



Choosing Strategy

- How quickly do you need to be able to recover a failing instance or add additional compute capacity?
- Does the workload baseline stay static for a relatively long period?
- Does your server configuration require manual provisioning or configuration?
- Do you need to minimize the complexity of deploying resources to both AWS and on-premises environments?
- Are there existing server provisioning tools or processes that you are trying to align with AWS?

Best Practices

- Avoid embedding passwords, private keys, or other sensitive information in AMIs
- Leverage AWS CloudFormation or a third-party configuration management tool to document and automate AMI creation and updating
- Create a library of reusable, modular templates that can be programmatically assembled to create different types of AMIs
- Instrument AMIs with a standard bootstrapping capability that allows the instance to reference runtime information at launch.
- Develop a consistent strategy for tagging AMIs to allow for the easy organization and identification of images and their contents

LAB – Scaling up/down

- Login to console
- Launch Instance as t1.micro
- Stop it
- Change Instance type to t2.small and Launch
- Connect to the instance

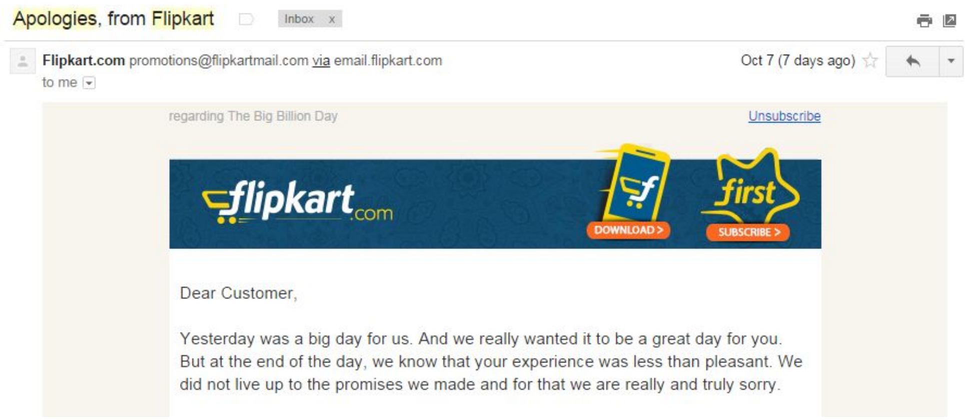
AutoScaling

Auto Scaling – Why ?

- Scale Automatically
- High Availability
- Fault Tolerance

Viral Nature

Auto Scaling – Why ?

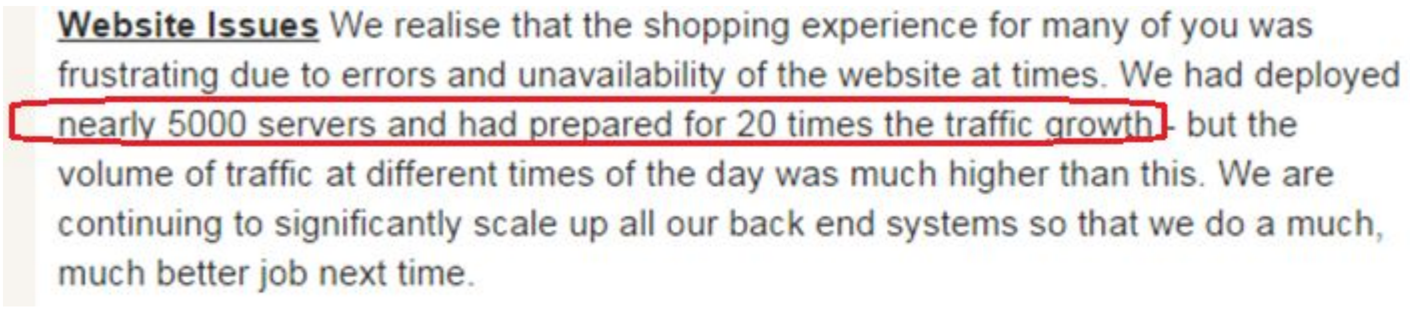

Apologies, from Flipkart Inbox x

Flipkart.com promotions@flipkartmail.com via email.flipkart.com Oct 7 (7 days ago)
to me

regarding The Big Billion Day Unsubscribe

Dear Customer,

Yesterday was a big day for us. And we really wanted it to be a great day for you. But at the end of the day, we know that your experience was less than pleasant. We did not live up to the promises we made and for that we are really and truly sorry.

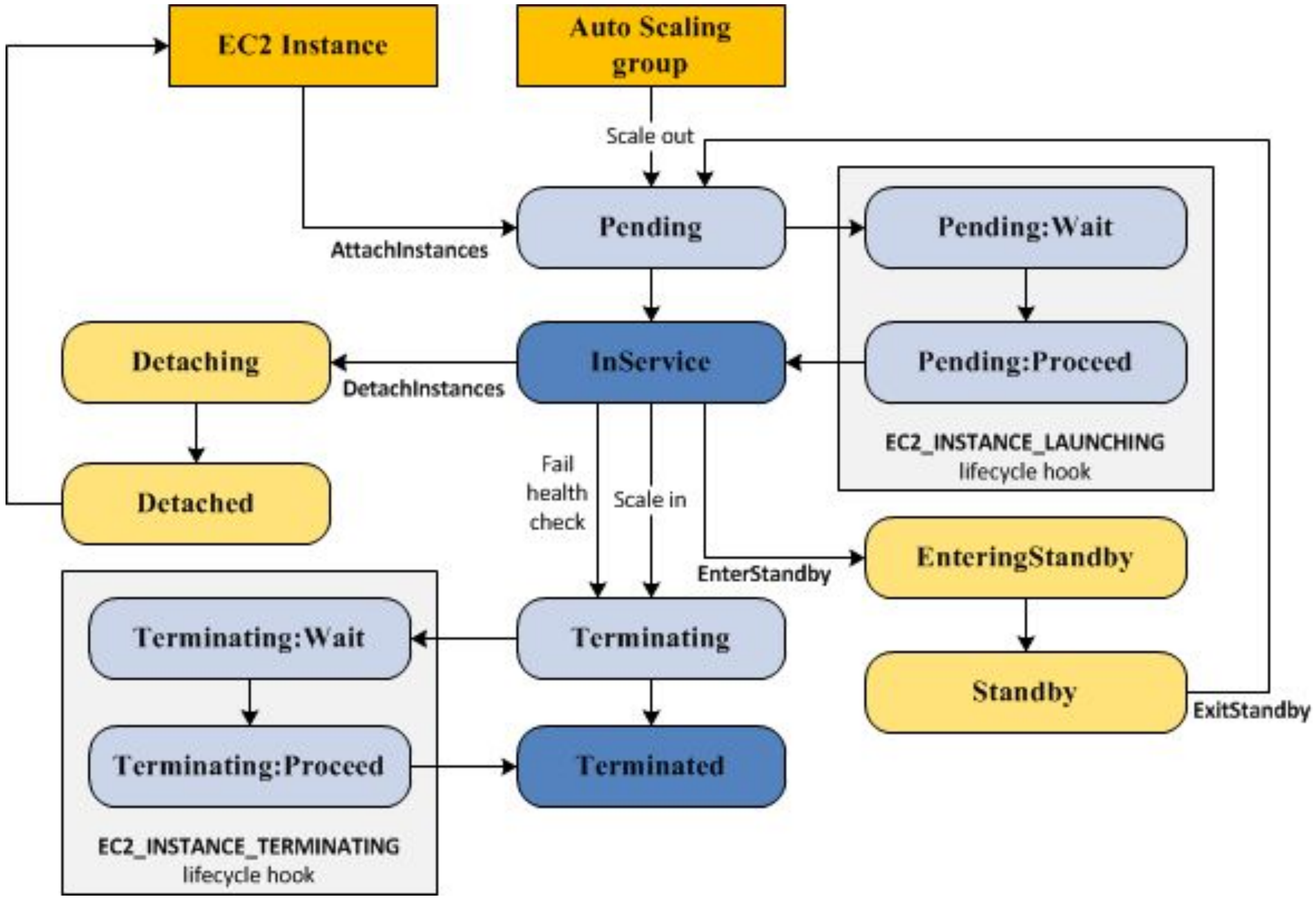

Website Issues We realise that the shopping experience for many of you was frustrating due to errors and unavailability of the website at times. We had deployed nearly 5000 servers and had prepared for 20 times the traffic growth - but the volume of traffic at different times of the day was much higher than this. We are continuing to significantly scale up all our back end systems so that we do a much, much better job next time.

Auto Scaling – Key Concepts

- **Auto Scaling Group**
  - Name
  - Collection of EC2 instances
  - Minimum Instances/ Maximum Instances / Desired Number of Instances
  - Amazon VPC
  - Subnet
  - Metrics and Health Checks

- **Launch Configurations (LC)**
  - Template that will be used by auto scaling group
  - AMI ID, instance type, key pair, security groups, and block device mapping for your instances.
- **Scaling Plan**
  - Tells when and how to scale
    - based on dynamic configuration (cloudwatch alarm)
    - scheduled
- **Integrated with Cloud Watch Alarms**

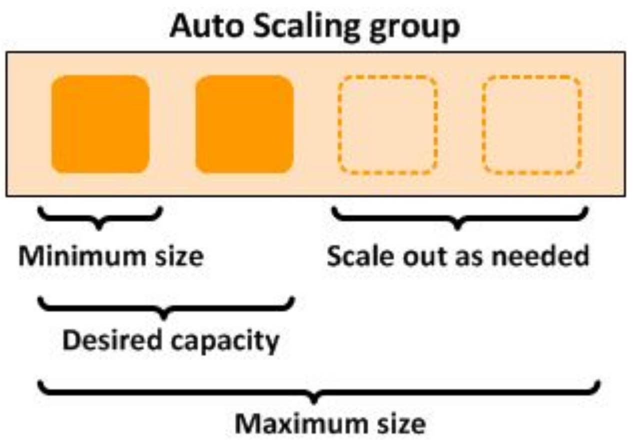
Auto Scaling – EC2 Instance Lifecycle



Auto Scaling – Steps

1) **Create Launch Configuration**
    - What to Launch - AMI
    - Storage
    - Security Group
    - Monitoring

2) **Create Auto Scaling Group**
    - Name
    - Group Size – Desired Capacity
    - Scaling Policies
        - When to Scale up and Scale Down



3) **Attach Launch Configuration to Auto Scaling Group**

End of Module