



# Masterclass

....

## *Elastic Compute Cloud*

Ian Massingham – Technical Evangelist

 @ianMmmm

# Masterclass

A technical deep dive beyond the basics

Help educate you on how to get the best from AWS technologies

Show you how things work and how to get things done

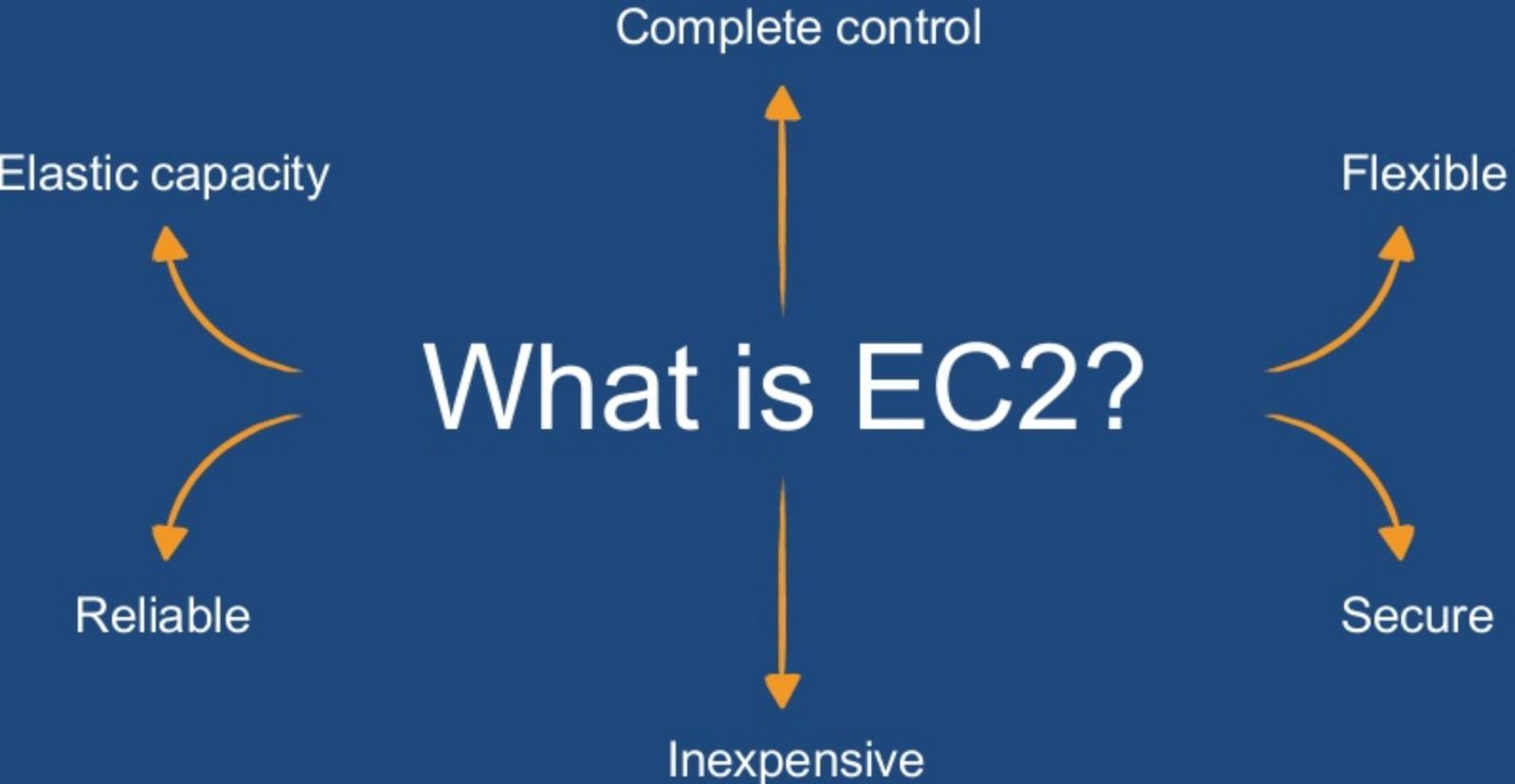
Broaden your knowledge in ~45 mins

# Amazon EC2

On-demand compute to run application workloads

Easy come easy go – disposable resource

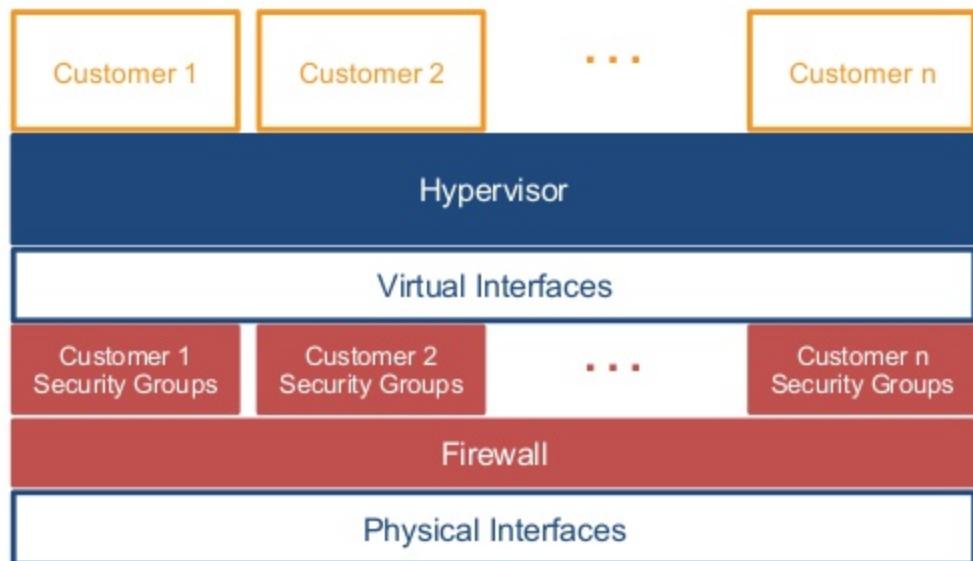
We provide the infrastructure, you decide what you run



Elastic capacity

Shared environment

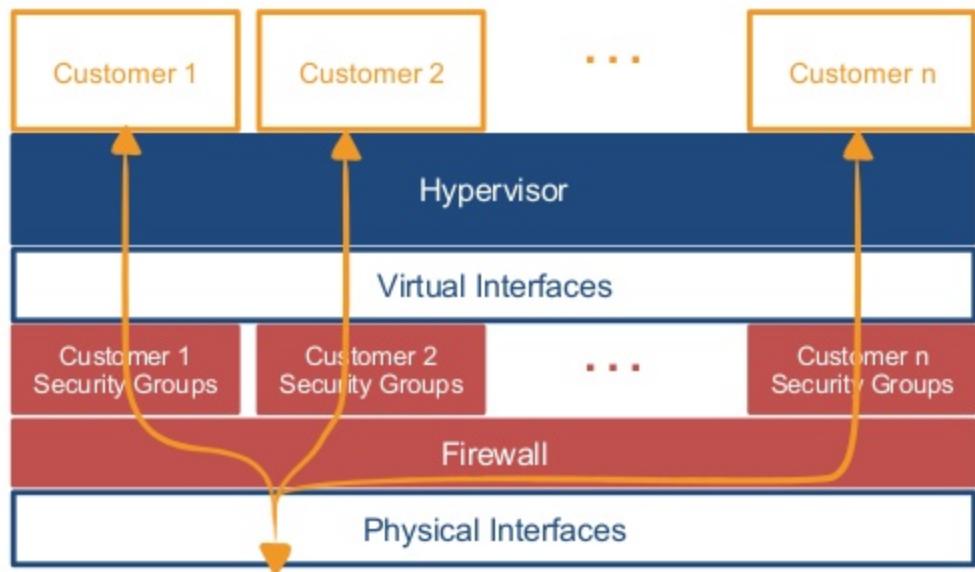
Securely segregated



Elastic capacity



Securely  
segregated



# Terminology



AMI

Amazon Machine  
Image

# Terminology



Amazon Machine  
Image

Running or  
Stopped machine

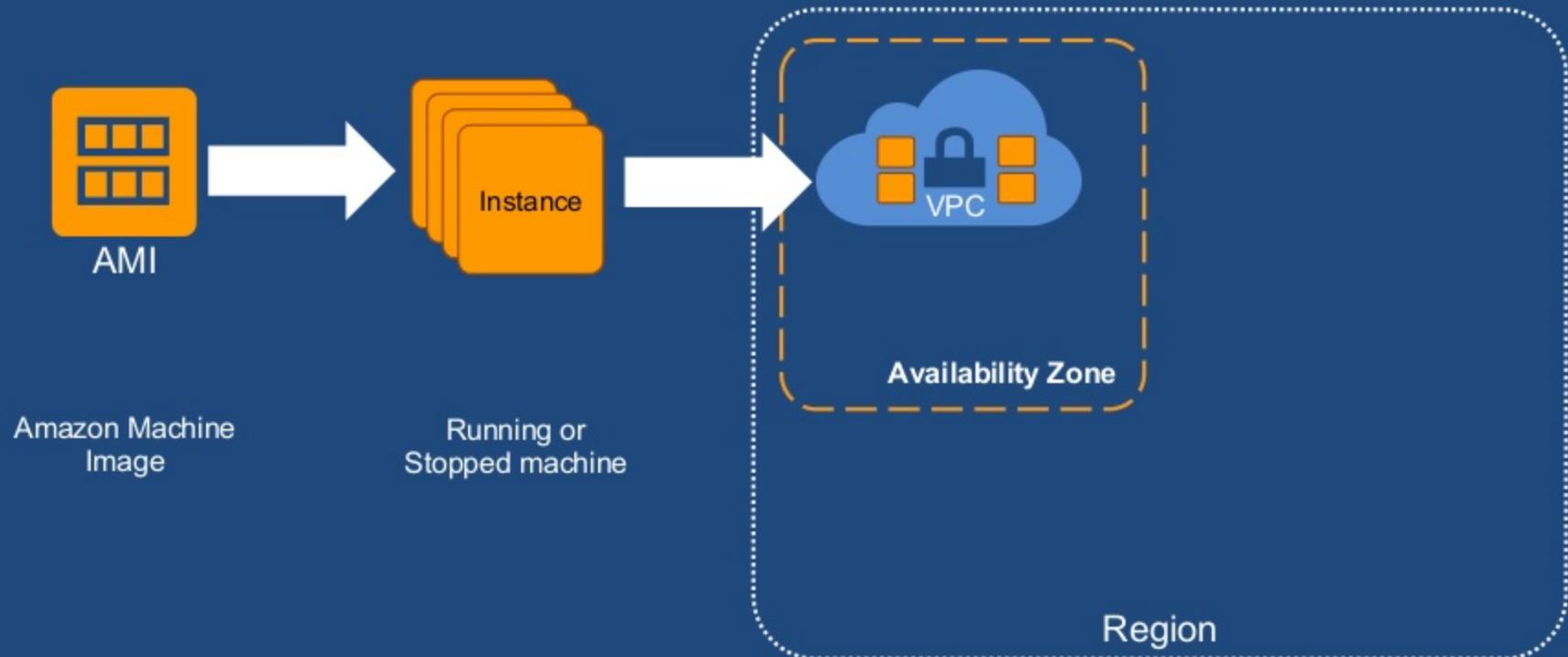
# Terminology



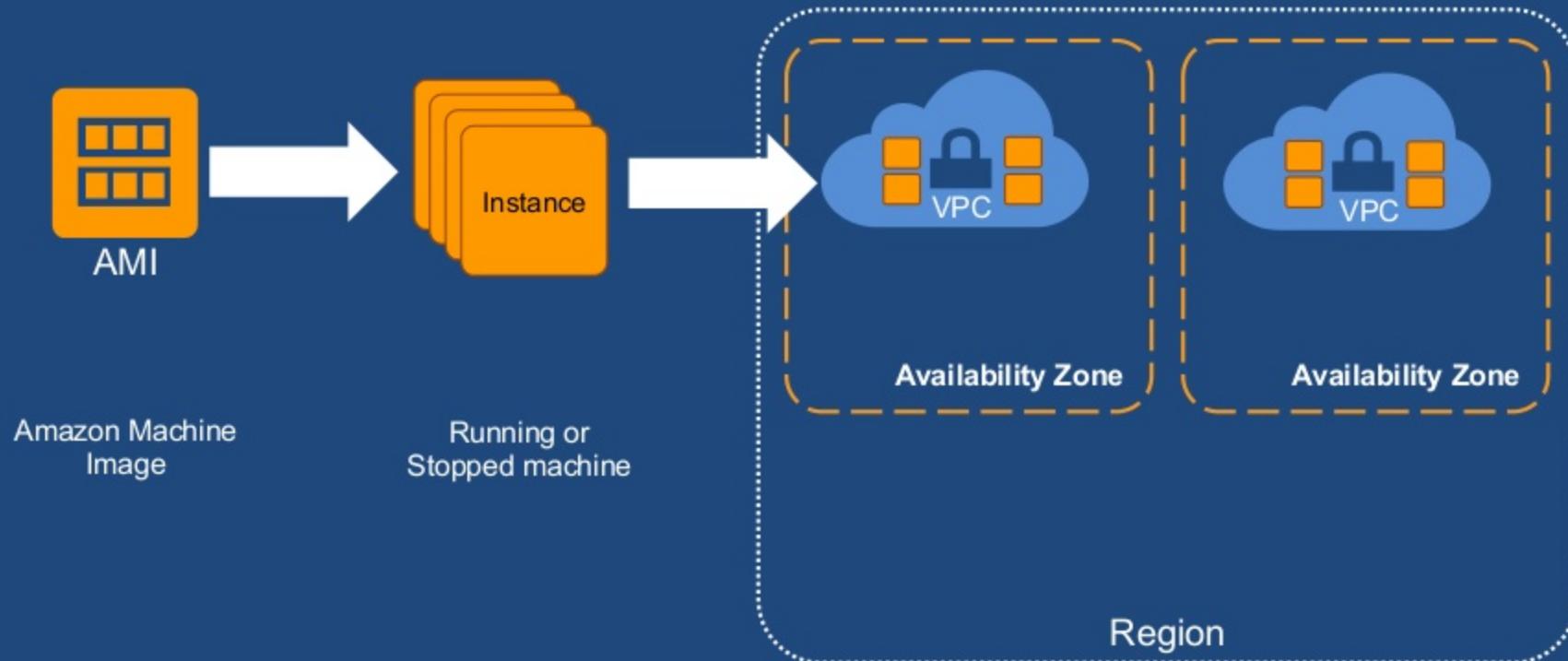
Amazon Machine  
Image

Running or  
Stopped machine

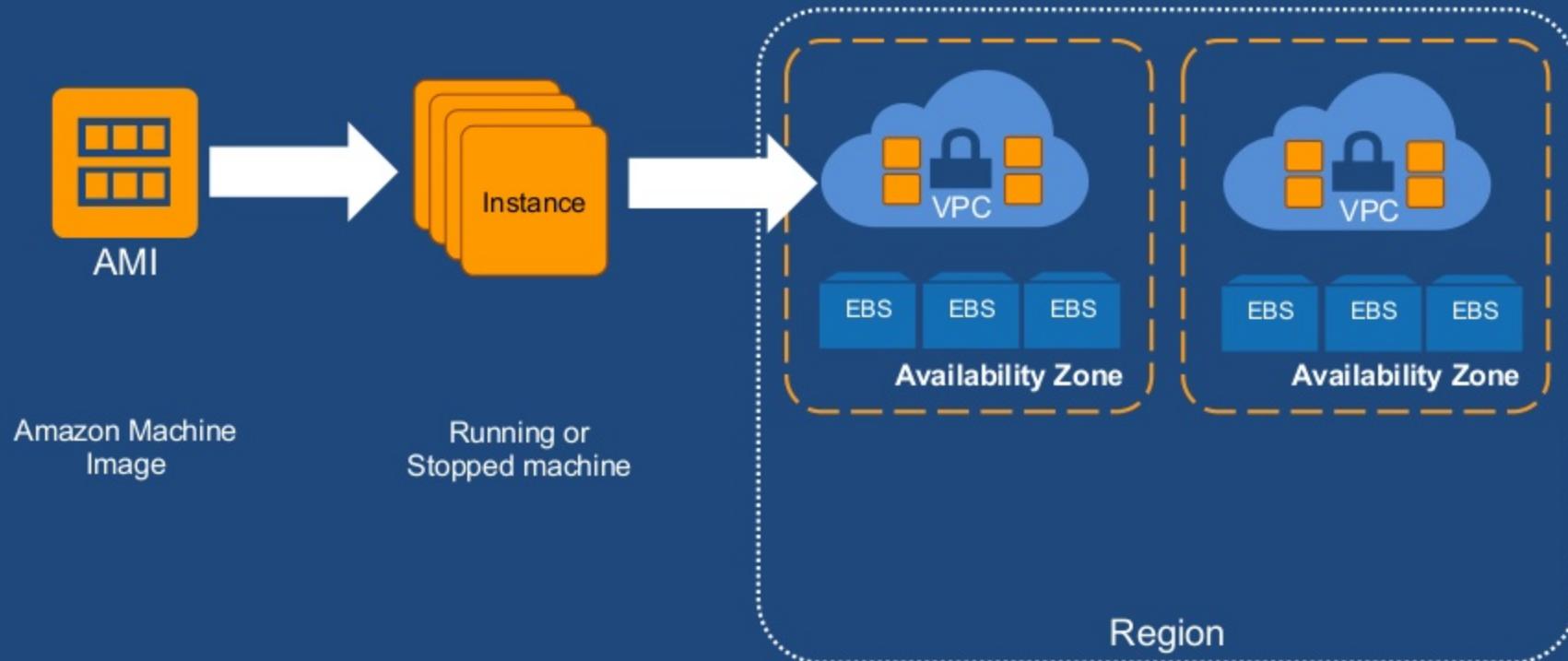
# Terminology



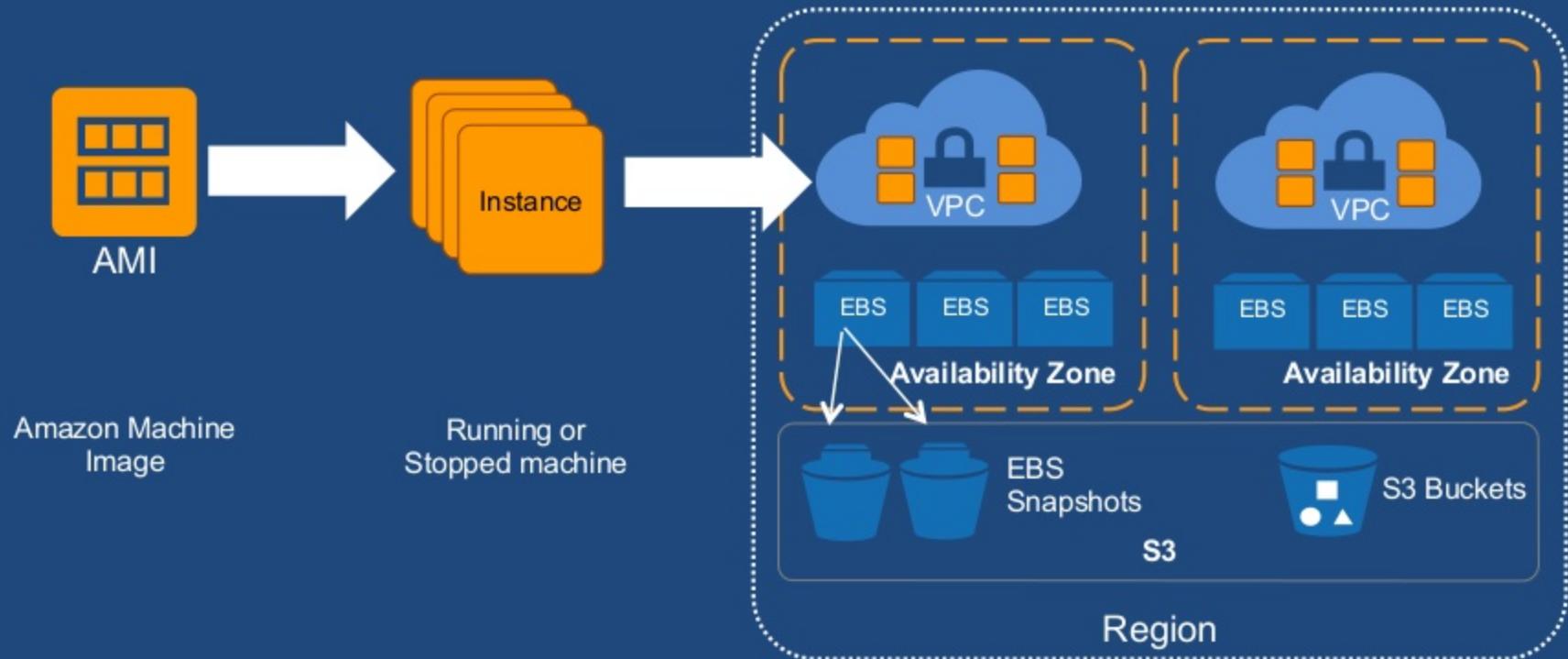
# Terminology



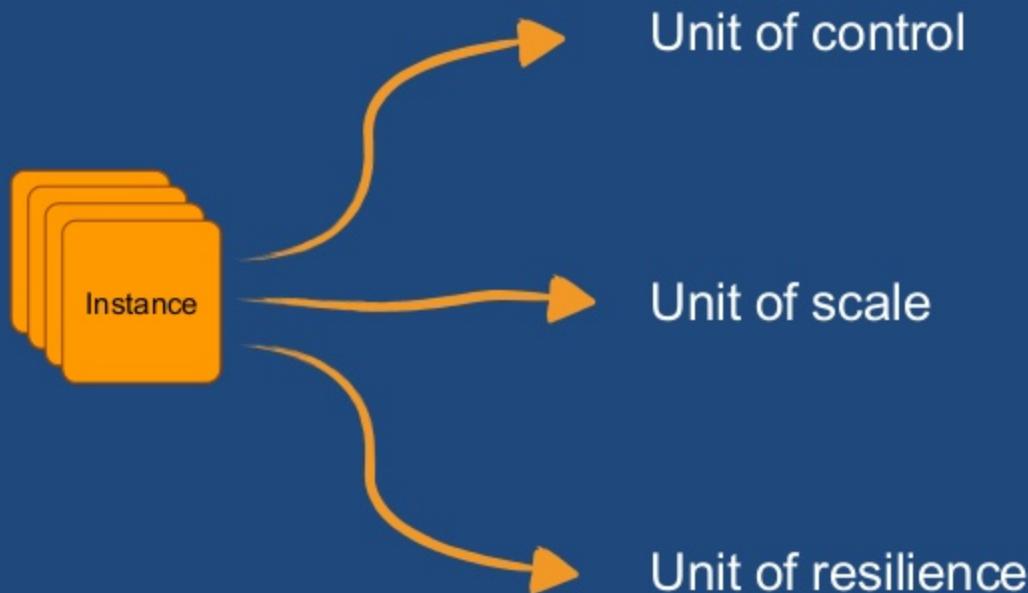
# Terminology

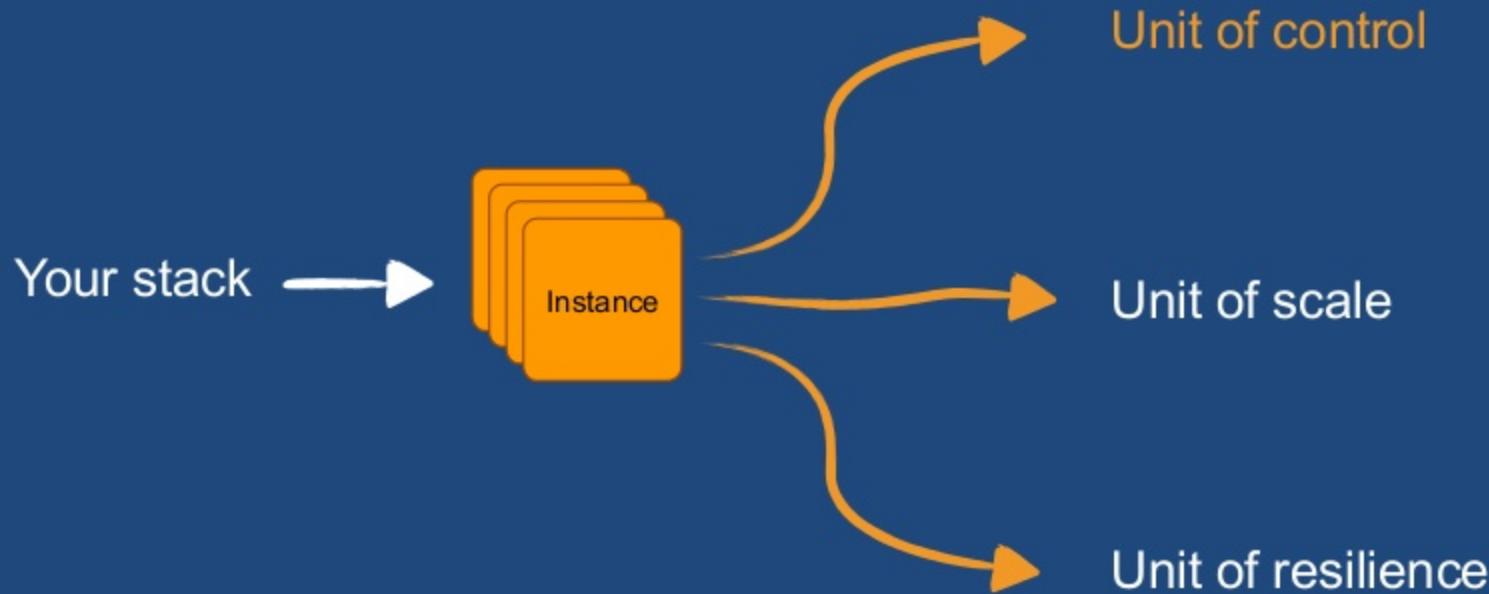


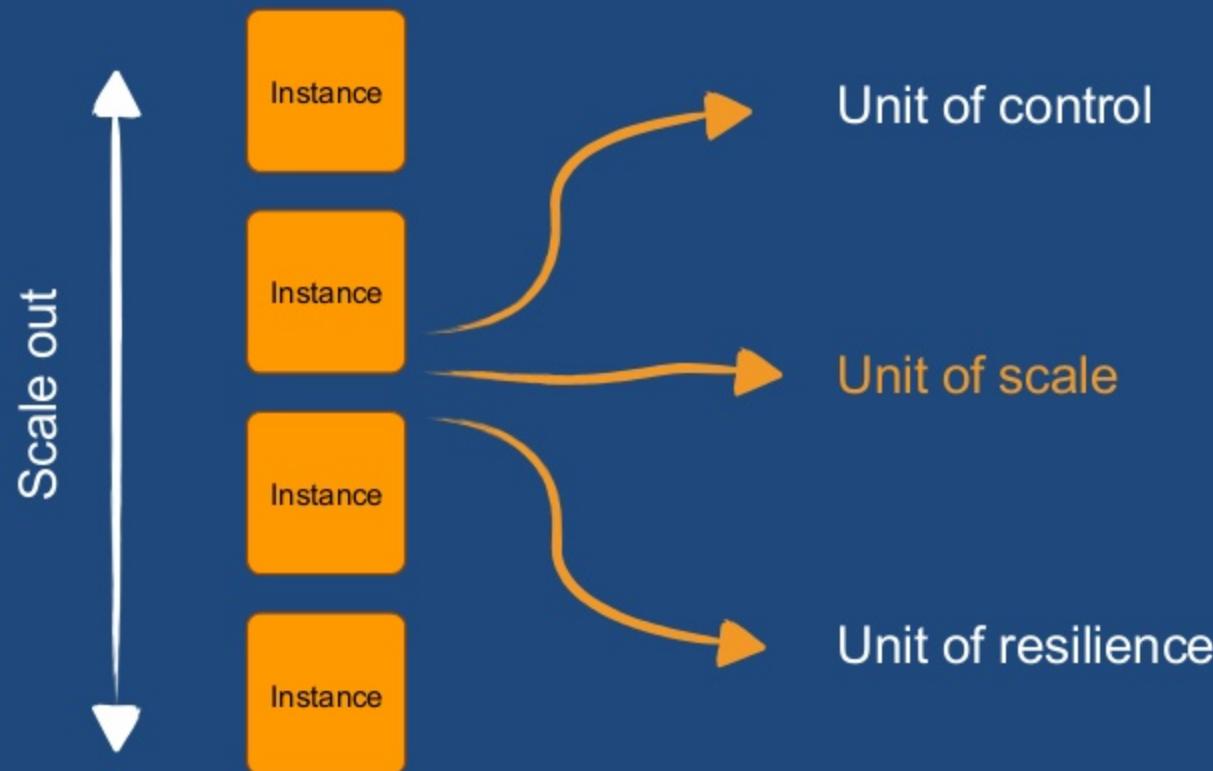
# Terminology

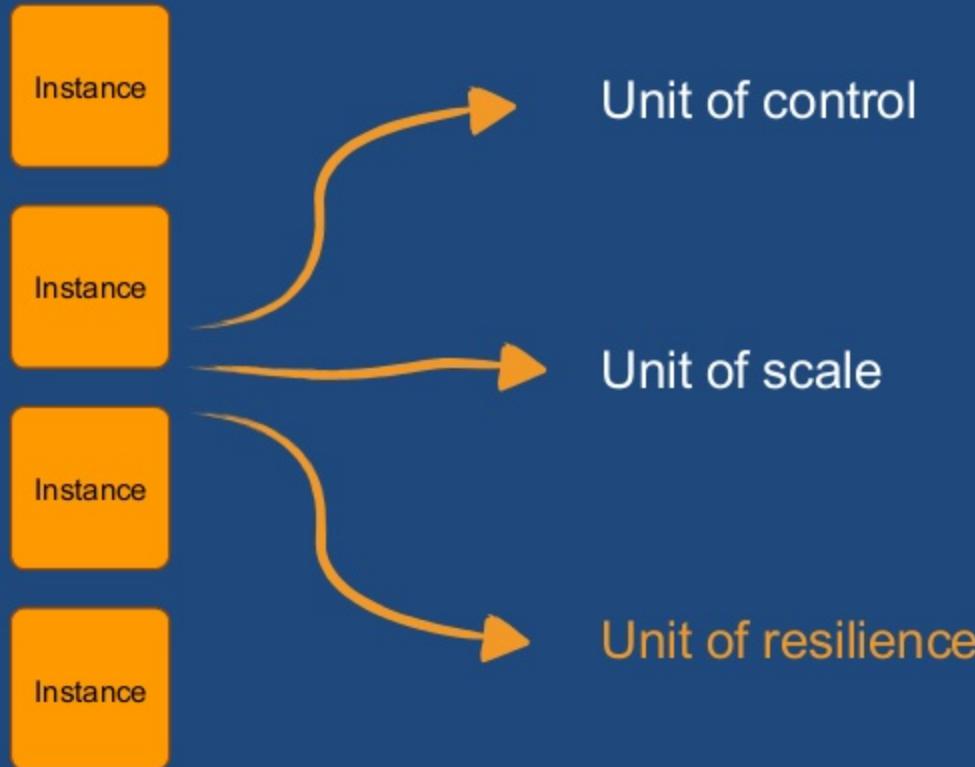


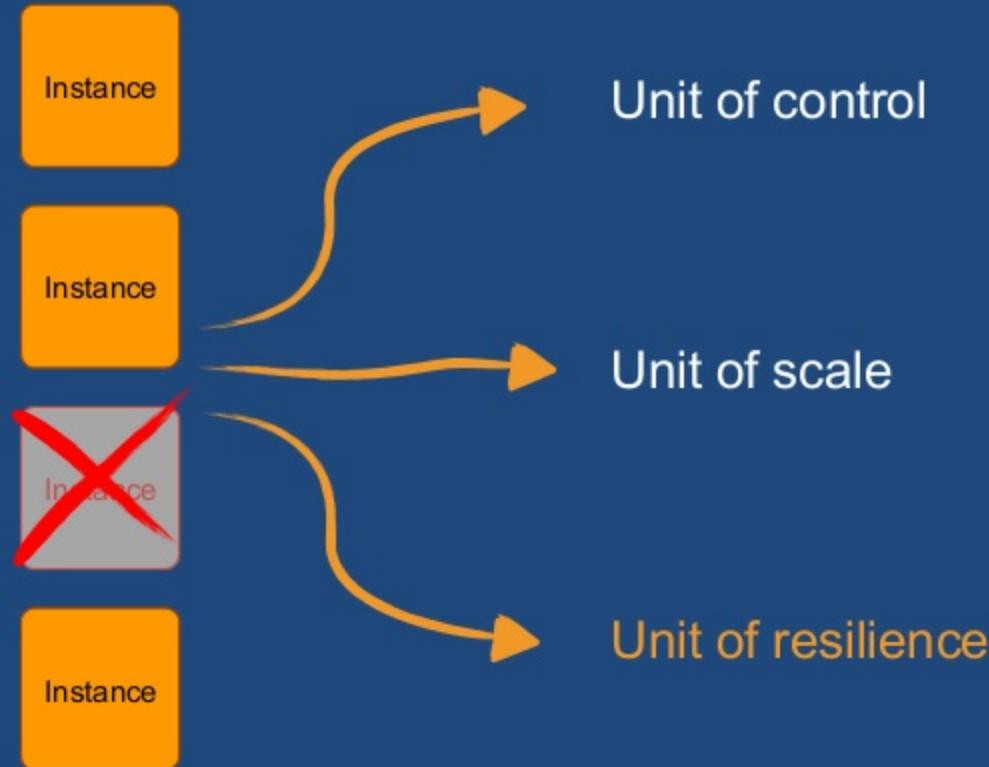


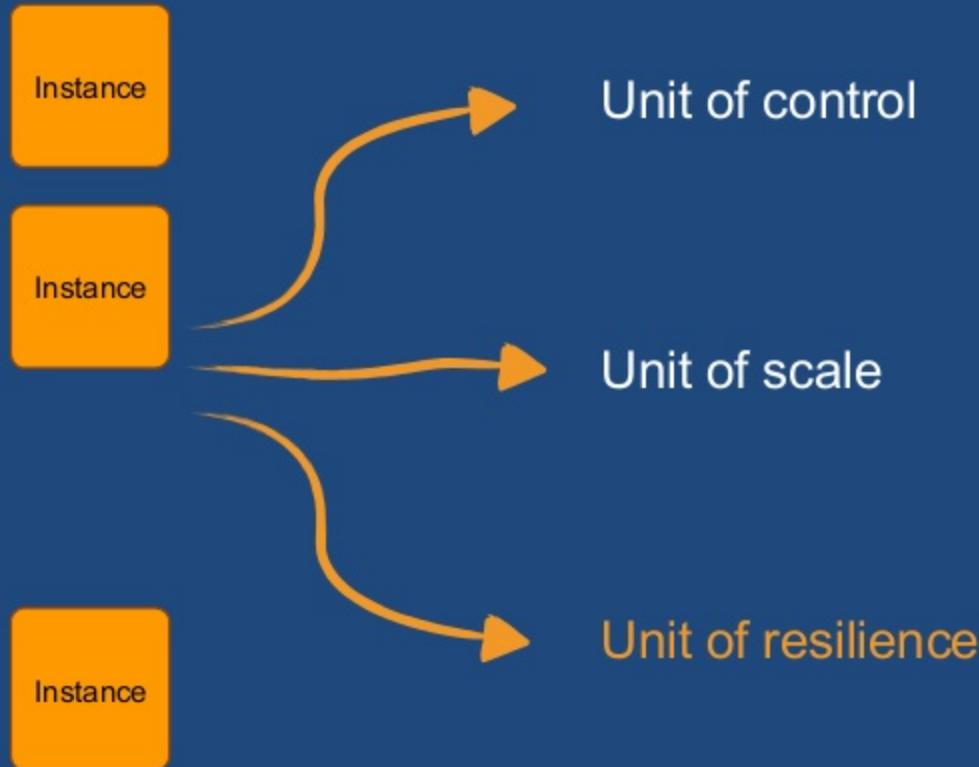


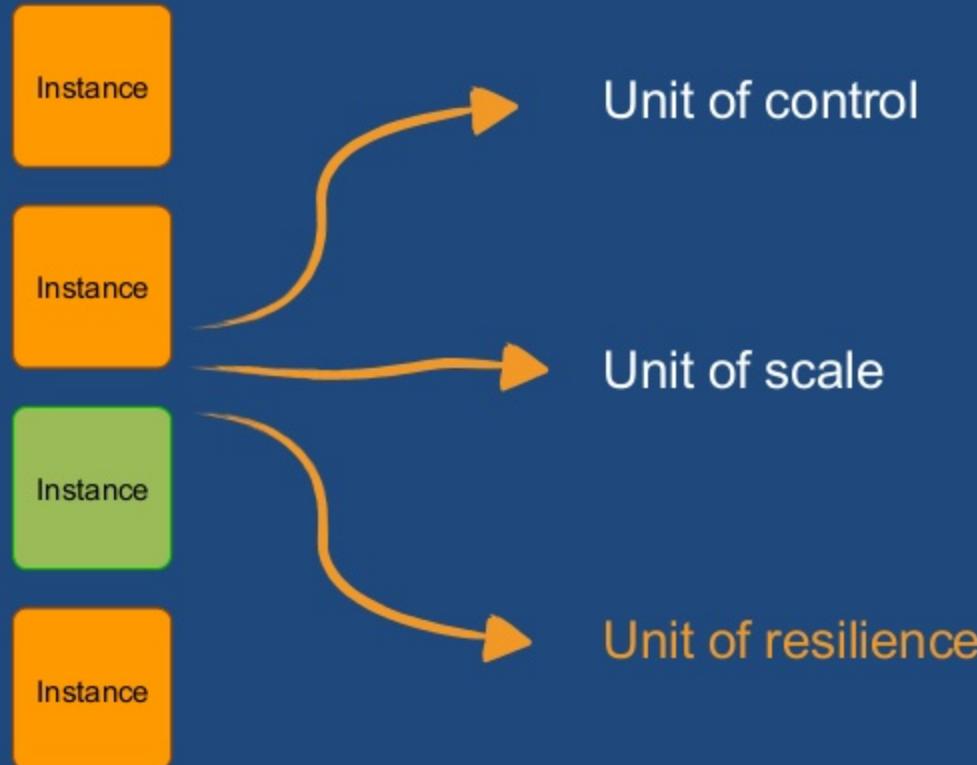






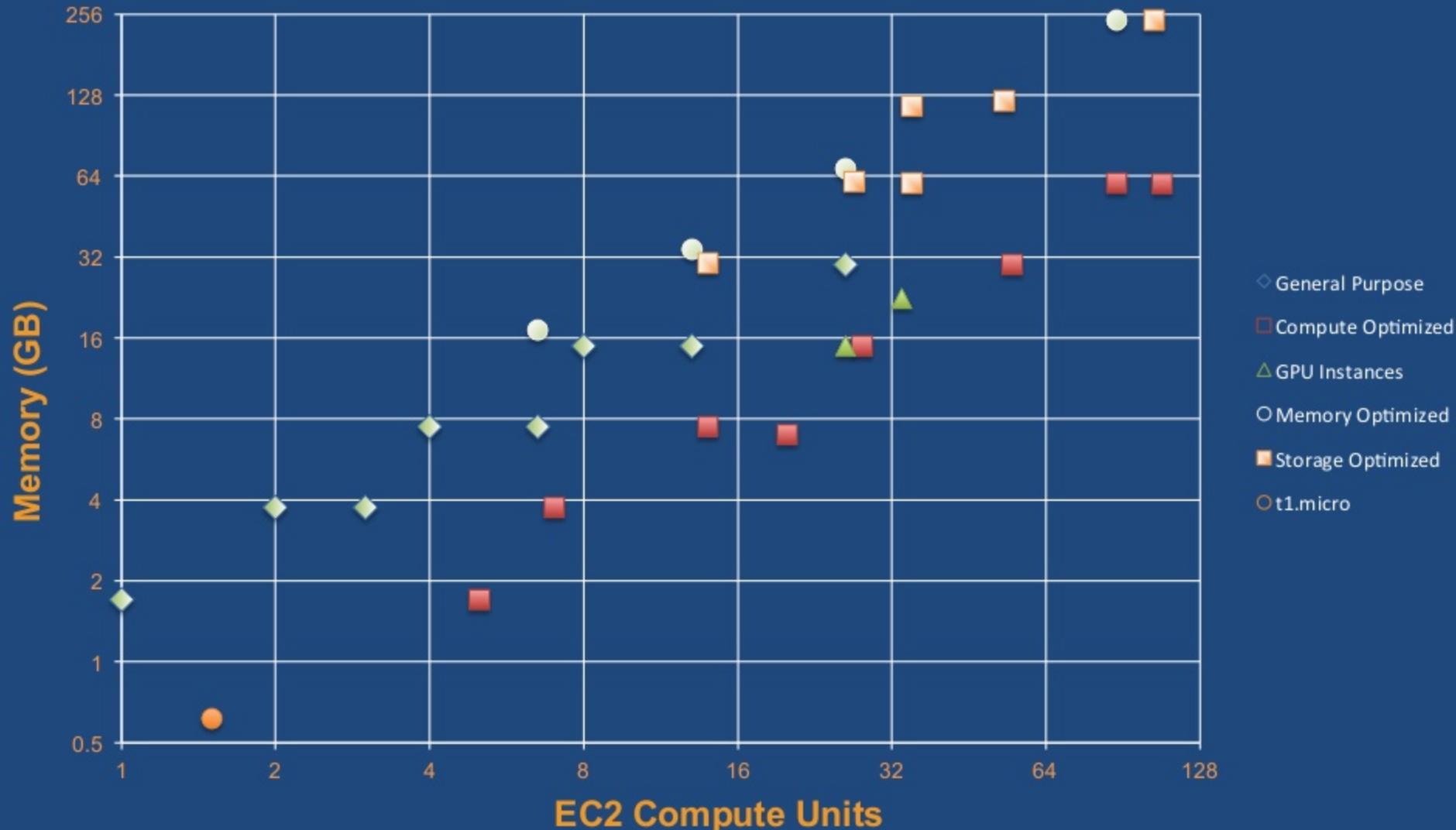


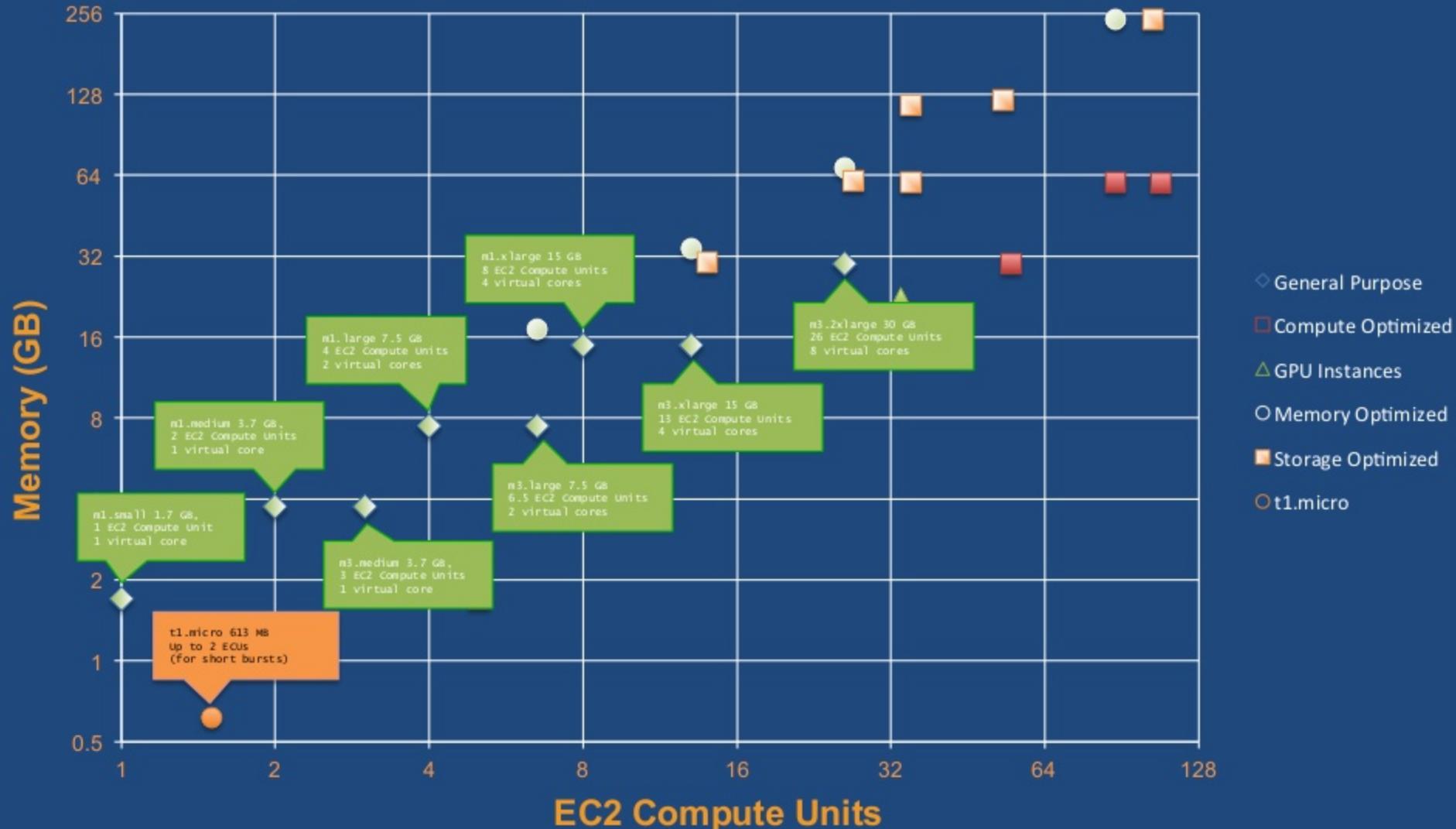


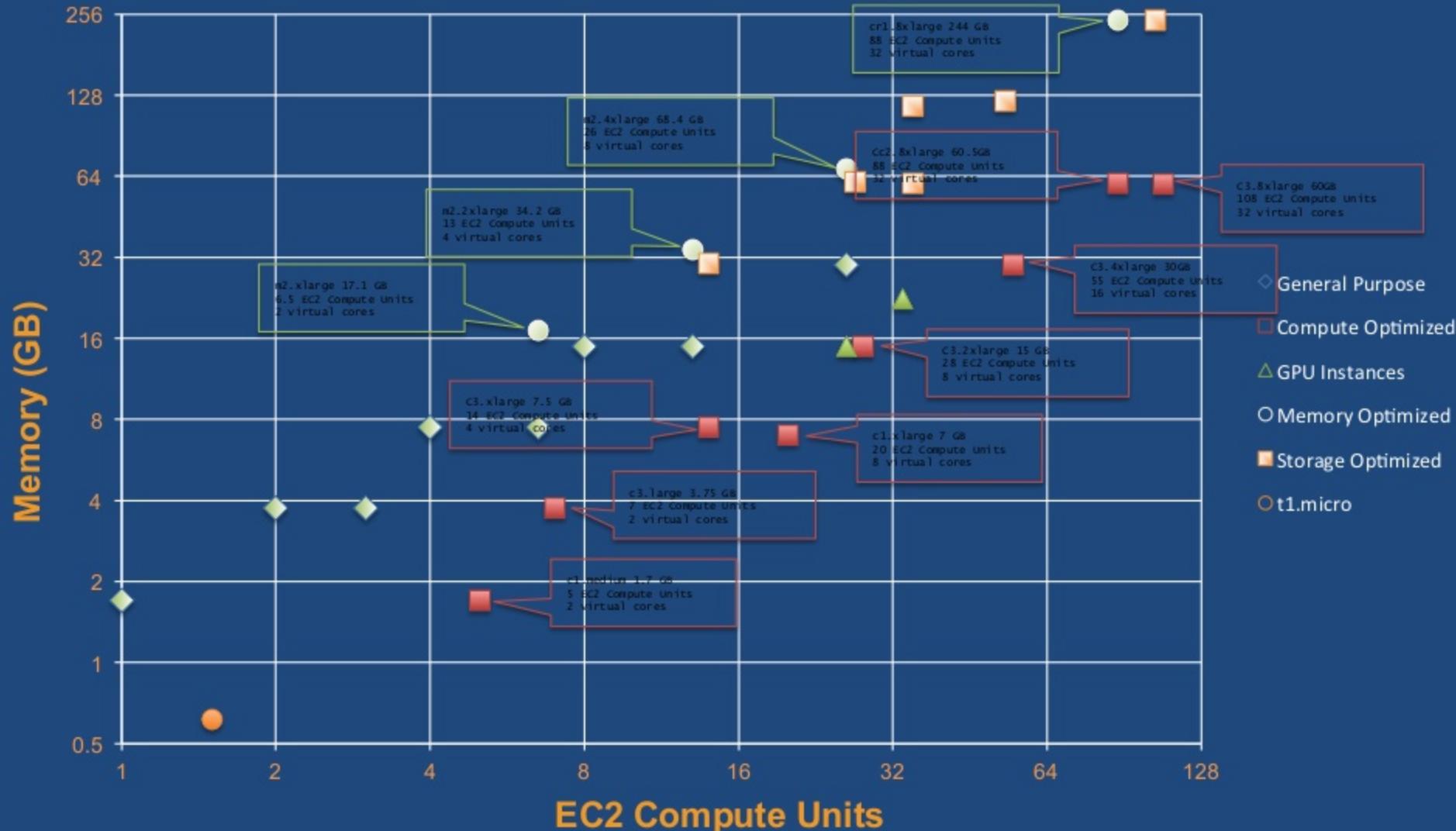


# Instance types

Choose the right unit for your workload







- + Storage Optimised
- + IO Optimised
- + GPU Backed

Start small  
Easy to up-size

# AMIs

## 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

# AMIs

## Linux



ubuntu®

Small instance from  
\$0.060 per hour

## Enterprise Linux



Small instance from  
\$0.120 per hour



Small instance from  
\$0.090 per hour

## Windows



Small instance from  
\$0.091 per hour

# Instance types

## On-demand instances

Unix/Linux instances start at \$0.02/  
hour

Pay as you go for compute power

Low cost and flexibility

Pay only for what you use, no up-front  
commitments or long-term contracts

### Use Cases:

*Applications with short term, spiky, or  
unpredictable workloads;*

*Application development or testing*

# Instance types

## On-demand instances

Unix/Linux instances start at \$0.02/  
hour

Pay as you go for compute power

Low cost and flexibility

Pay only for what you use, no up-front  
commitments or long-term contracts

### Use Cases:

*Applications with short term, spiky, or  
unpredictable workloads;*

*Application development or testing*

## Reserved instances

### 1- or 3-year terms

Pay low up-front fee, receive significant hourly  
discount

Low Cost / Predictability

Helps ensure compute capacity is available  
when needed

### Use Cases:

*Applications with steady state or predictable  
usage*

*Applications that require reserved capacity,  
including disaster recovery*

# Instance types

On-demand instances		Heavy utilization RI
Unix/Linux instances start at \$0.02/ hour		> 80% utilization Lower costs up to 58%
Pay as you go for compute power		<i>Use Cases:</i> Databases, Large Scale HPC, Always-on infrastructure, Baseline
Low cost and flexibility		
Pay only for what you use, no up-front commitments or long-term contracts		
Reserved instances		
1- or 3-year terms		
Pay low up-front fee, receive significant hourly discount		
Low Cost / Predictability		
Helps ensure compute capacity is available when needed		
<u>Use Cases:</u> <i>Applications with short term, spiky, or unpredictable workloads;</i>		
<i>Application development or testing</i>		
<u>Use Cases:</u> <i>Applications with steady state or predictable usage</i>		
<i>Applications that require reserved capacity, including disaster recovery</i>		

# Instance types

## On-demand instances

Unix/Linux instances start at \$0.02/  
hour

Pay as you go for compute power

Low cost and flexibility

Pay only for what you use, no up-front  
commitments or long-term contracts

### Use Cases:

*Applications with short term, spiky, or  
unpredictable workloads;*

*Application development or testing*

## Reserved instances

### 1- or 3-year terms

Pay low up-front fee, receive significant hourly  
discount

Low Cost / Predictability

Helps ensure compute capacity is available  
when needed

### Use Cases:

*Applications with steady state or predictable  
usage*

*Applications that require reserved capacity,  
including disaster recovery*

## Heavy utilization RI

> 80% utilization  
Lower costs up to 58%

*Use Cases: Databases, Large Scale HPC,  
Always-on infrastructure, Baseline*

## Medium utilization RI

41-79% utilization  
Lower costs up to 49%

*Use Cases: Web applications, many heavy  
processing tasks, running much of the time*

# Instance types

## On-demand instances

Unix/Linux instances start at \$0.02/  
hour

Pay as you go for compute power

Low cost and flexibility

Pay only for what you use, no up-front  
commitments or long-term contracts

### Use Cases:

*Applications with short term, spiky, or  
unpredictable workloads;*

*Application development or testing*

## Reserved instances

### 1- or 3-year terms

Pay low up-front fee, receive significant hourly  
discount

Low Cost / Predictability

Helps ensure compute capacity is available  
when needed

### Use Cases:

*Applications with steady state or predictable  
usage*

*Applications that require reserved capacity,  
including disaster recovery*

## Heavy utilization RI

> 80% utilization  
Lower costs up to 58%

*Use Cases: Databases, Large Scale HPC,  
Always-on infrastructure, Baseline*

## Medium utilization RI

41-79% utilization  
Lower costs up to 49%

*Use Cases: Web applications, many heavy  
processing tasks, running much of the time*

## Light utilization RI

15-40% utilization  
Lower costs up to 34%

*Use Cases: Disaster Recovery, Weekly /  
Monthly reporting, Elastic Map Reduce*

# Instance types

## On-demand instances

Unix/Linux instances start at \$0.02/  
hour

Pay as you go for compute power

Low cost and flexibility

Pay only for what you use, no up-front  
commitments or long-term contracts

### Use Cases:

*Applications with short term, spiky, or  
unpredictable workloads;*

*Application development or testing*

## Reserved instances

1- or 3-year terms

Pay low up-front fee, receive significant hourly  
discount

Low Cost / Predictability

Helps ensure compute capacity is available  
when needed

### Use Cases:

*Applications with steady state or predictable  
usage*

*Applications that require reserved capacity,  
including disaster recovery*

## Spot instances

**Bid on unused EC2 capacity**

Spot Price based on supply/demand,  
determined automatically

Cost / Large Scale, dynamic workload handling

### Use Cases:

*Applications with flexible start and end times*

*Applications only feasible at very low compute  
prices*

# Launch an instance

Commands, keypairs & security groups

# Console

EC2 Management Console

Services EC2 RDS CloudFront Route 53 Edit

Ianmas@ianmas-aws Ireland Help

1. Choose AMI 2. Choose Instance Type 3. Configure Instance 4. Add Storage 5. Tag Instance 6. Configure Security Group 7. Review Cancel and Exit

### Step 1: Choose an Amazon Machine Image (AMI)

An AMI is a template that contains the software configuration (operating system, application server, and applications) required to launch your instance. You can select an AMI provided by AWS, our user community, or the AWS Marketplace, or you can select one of your own AMIs.

**Quick Start**

**My AMIs**

**AWS Marketplace**

**Community AMIs**

Free tier only ⓘ

**Amazon Linux**

 **Amazon Linux AMI 2013.09.2 - ami-5256b825 (64-bit) / ami-8a66b81d (32-bit)**  
The Amazon Linux AMI is an EBS-backed, PV-GRUB image. It includes Linux 3.4, AWS tools, and repository access to multiple versions of MySQL, PostgreSQL, Python, Ruby, and Tomcat.  
 64-bit  32-bit  
**Select**

**Red Hat**

 **Red Hat Enterprise Linux 6.4 (PV) - ami-75342c01 (64-bit) / ami-8b332bff (32-bit)**  
Red Hat Enterprise Linux version 6.4 (PV), EBS-backed.  
 64-bit  32-bit  
**Select**

**SUSE Linux**

 **SUSE Linux Enterprise Server 11 sp3 (PV) - ami-8d1109f9 (64-bit) / ami-fd1109ff (32-bit)**  
SUSE Linux Enterprise Server 11 Service Pack 3 (PV), EBS-backed with Amazon EC2 AMI Tools preinstalled; Apache 2.2, MySQL 5.5, PHP 5.3, and Ruby 1.8.7 available.  
 64-bit  32-bit  
**Select**

**Ubuntu**

 **Ubuntu Server 12.04 LTS (PV) - ami-808675f7 (64-bit) / ami-828675f5 (32-bit)**  
Ubuntu Server 12.04 LTS (PV), EBS-backed with support available from Canonical (<http://www.ubuntu.com/cloud/services>).  
 64-bit  32-bit  
**Select**

 **Ubuntu Server 13.10 (PV) - ami-e050a19b (64-bit) / ami-e050a197 (32-bit)**  
Ubuntu Server 13.10 (PV), EBS-backed with support available from Canonical (<http://www.ubuntu.com/cloud/services>).  
 64-bit  32-bit  
**Select**

**Amazon Linux**

 **Amazon Linux AMI (HVM) 2013.09.2 - ami-7c56b80b**  
The Amazon Linux AMI is an EBS-backed, HVM image. It includes Linux 3.4, AWS tools, and repository access to multiple versions of MySQL, PostgreSQL, Python, Ruby, and Tomcat.  
 64-bit  
**Select**

**Red Hat**

 **Red Hat Enterprise Linux 6.5 (HVM) - ami-2ce30f5b**  
Red Hat Enterprise Linux version 6.5 (HVM), EBS-backed.  
 64-bit  
**Select**

**SUSE Linux**

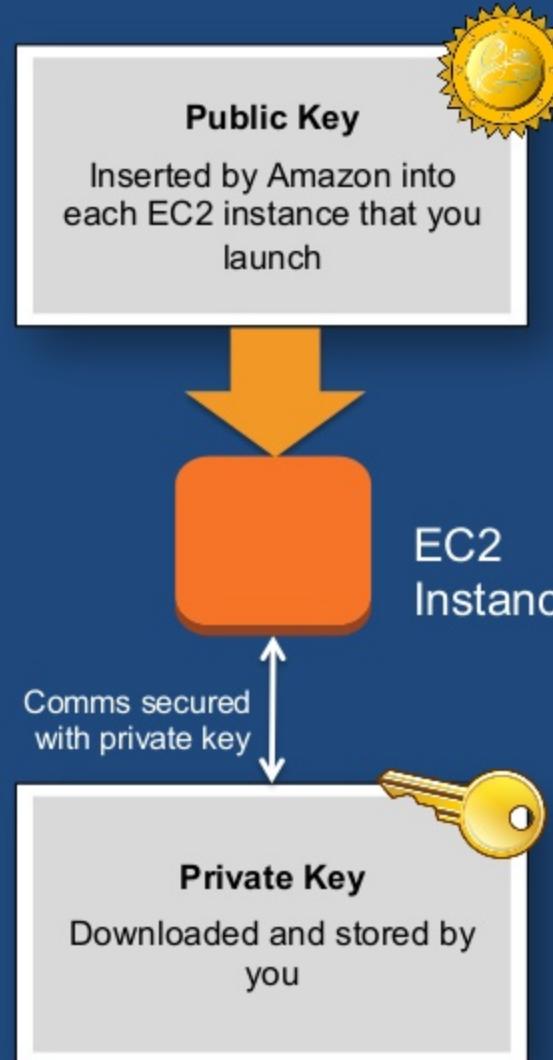
 **SUSE Linux Enterprise Server 11 sp3 (HVM) - ami-4a5fb63d**  
SUSE Linux Enterprise Server 11 Service Pack 3, EBS-backed. Nvidia driver installs automatically during startup.  
 64-bit  
**Select**

◀ ▶ 1 to 19 of 19 AMIs ▶ ▷

© 2008 - 2014, Amazon Web Services, Inc. or its affiliates. All rights reserved. Privacy Policy Terms of Use Feedback

Region  
Instance size  
AMI  
Key pair  
Security group

key pairs  
secure access



# Keypairs & Secrets

## Keypairs

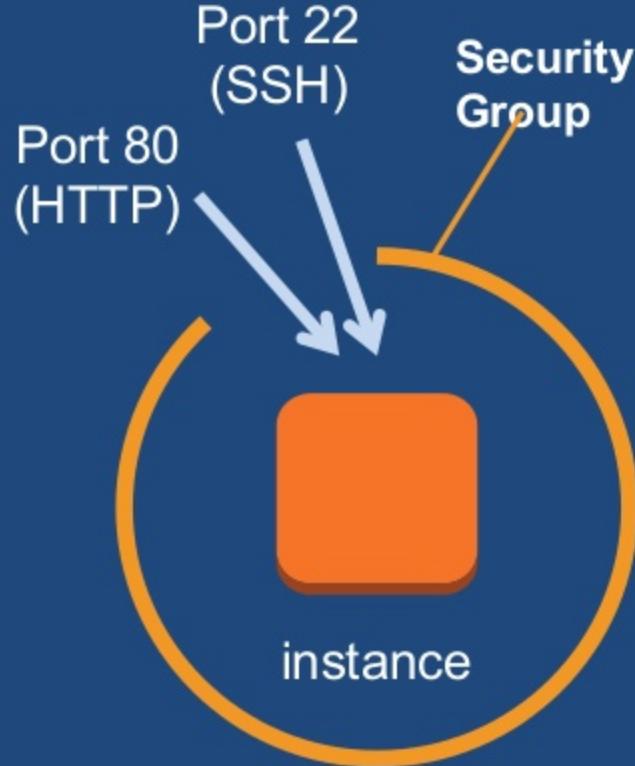
Used to authenticate  
when accessing and  
instance

## Credentials

Access key and secret  
key used to authenticate  
against APIs

# security groups

## instance firewalling



- Name  
Description  
Protocol  
Port range  
IP Address, range, or another security group

```
PS C:> New-EC2Instances
```

```
    -ImageId ami-269dbb63  
    -KeyName mykey  
    -SecurityGroupId sg-9cf9e5d9  
    -InstanceType t1.micro
```

```
$> aws ec2 run-instances  
--image-id ami-54cf5c3d  
--count 2  
--security-groups webservers  
--key-name mykey  
--instance-type m1.small
```

```
>>> import boto.ec2  
>>> conn = boto.ec2.connect_to_region("us-east-1")  
>>> conn.run_instances(  
    'ami-54cf5c3d',  
    key_name='mykey',  
    instance_type='m1.small',  
    security_groups=['webservers'])
```

Wait a minute  
I want to use those tools too...

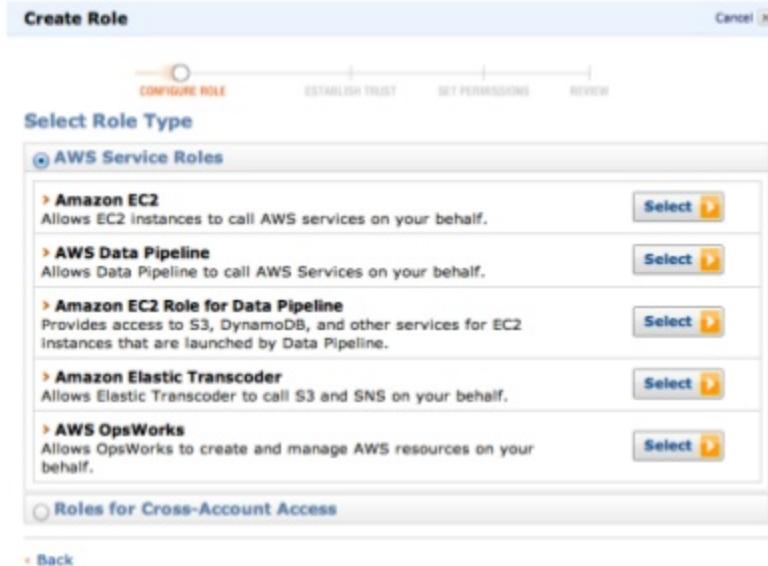
## Tip

### IAM Roles and EC2 tools

1. Start an EC2 Linux instance
2. Assign an IAM role at launch time:

IAM Role:  aws-tools

3. Sets up all the tools you need & manages API access credentials
4. Up and running with CLI tools in a couple of minutes – just SSH on and use
5. Terminate/stop instance when you are done



```
{  
  "Statement": [  
    {  
      "Effect": "Allow",  
      "NotAction": "iam:*",  
      "Resource": "*"  
    }  
  ]  
}
```

# Now you have tools

Try this...

```
$> aws ec2 run-instances  
  --image-id ami-54cf5c3d  
  --count 2
```

```
$> aws ec2 run-instances  
  --image-id ami-54cf5c3d  
  --count 2  
  
  --security-groups webservers  
  --key-name mykey  
  --instance-type m1.small
```

What about all this?



```
$> aws ec2 run-instances  
  --image-id ami-54cf5c3d  
  --count 2  
  
  --security-groups Default  
  --key-name NONE  
  --instance-type default (m1.small)
```

Defaults



```
$> aws ec2 run-instances  
  --image-id ami-54cf5c3d  
  --count 2  
  
?  --security-groups Default  
   --key-name NONE  
   --instance-type default (m1.small)
```

Instances don't need  
keypairs

But how do you configure it if you can't  
log onto it?

# Bootstrapping

## Bake an AMI

Start an instance

Configure the instance

Create an AMI from your  
instance

Start new ones from the  
AMI

# Bootstrapping

Bake an AMI

vs

Configure dynamically

Start an instance

Launch an instance

Configure the instance

Use metadata service  
and cloud-init to perform  
actions on instance  
when it launches

Create an AMI from your  
instance

Start new ones from the  
AMI

# Bootstrapping

Bake an AMI



Configure dynamically

Build your base images  
and setup custom  
initialisation scripts

Maintain your ‘golden’  
base

Use bootstrapping to  
pass custom information  
in and perform post  
launch tasks like pulling  
code from SVN

# Bootstrapping

Bake an AMI



Configure dynamically

Time consuming  
configuration (*startup time*)

Static configurations (*less  
change management*)

# Bootstrapping

Bake an AMI



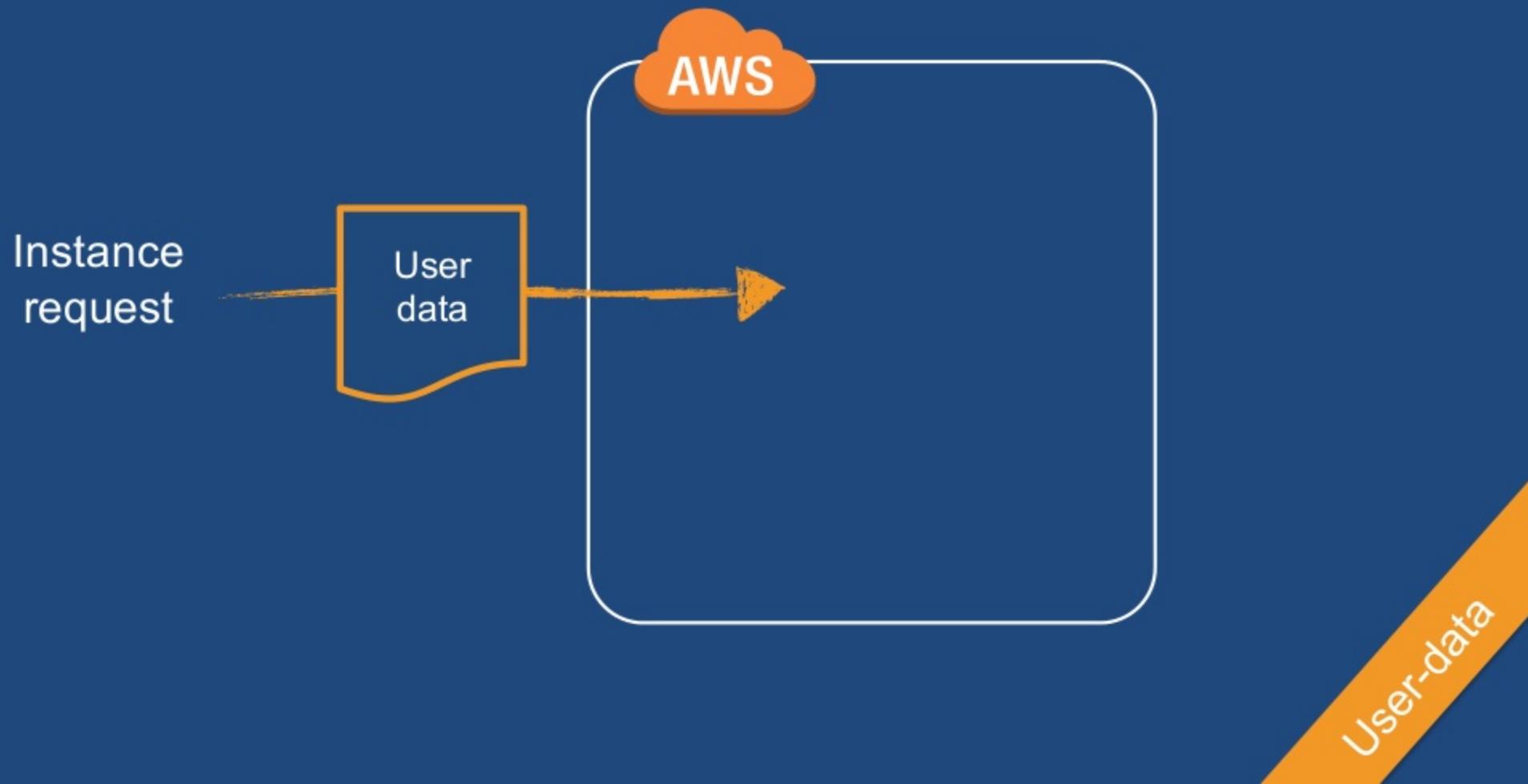
Configure dynamically

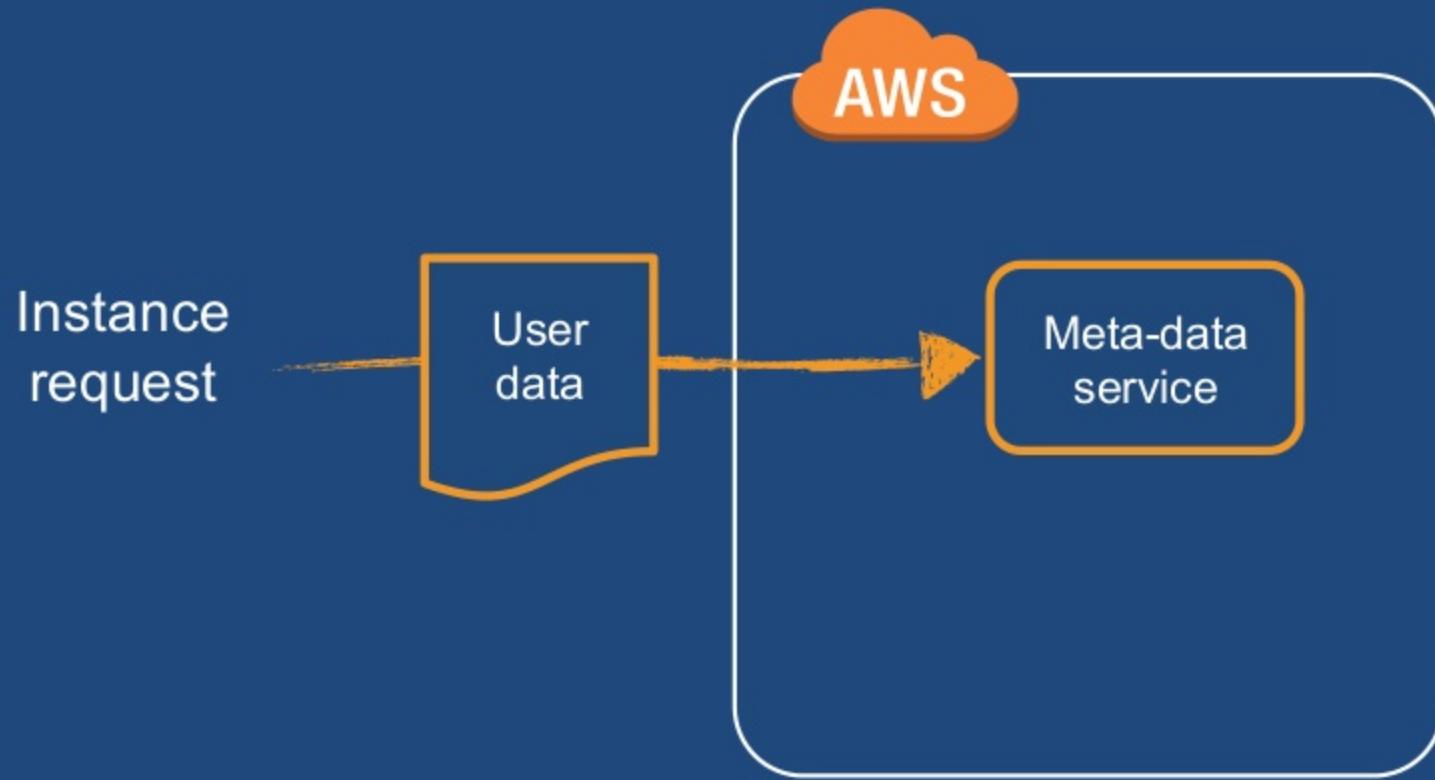
Continuous deployment  
(*latest code*)

Environment specific (*dev-test-prod*)

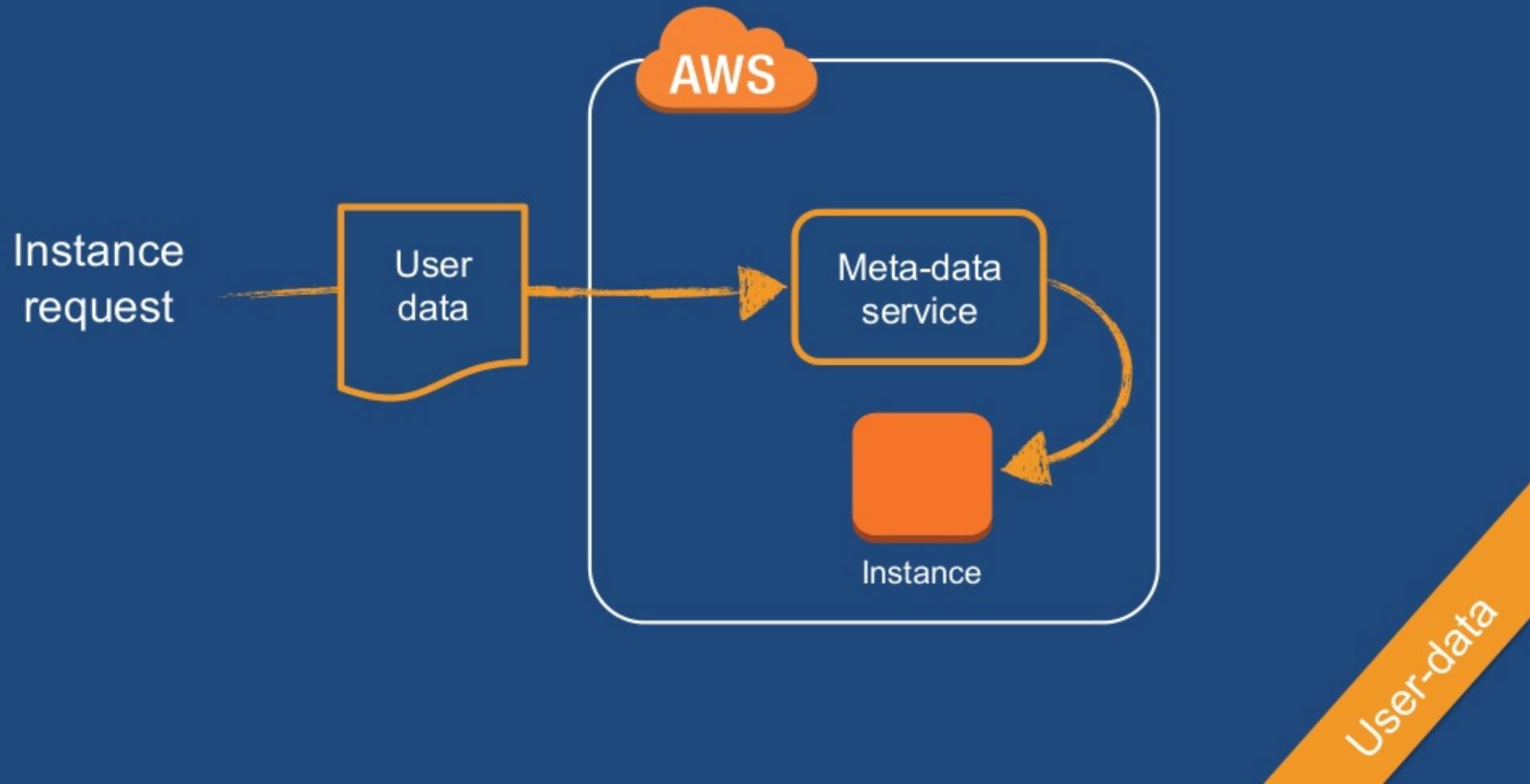
Goal is bring an instance up in a useful state

The balance will vary depending upon your application





User-data



## Tip

Shell script in user-data will be executed on launch:

```
#!/bin/sh
yum -y install httpd php mysql php-mysql
chkconfig httpd on
/etc/init.d/httpd start
```

Amazon Windows EC2Config Service executes user-data on launch:

```
<script>dir > c:\test.log</script>
```

```
<powershell>any command that you can run</powershell>
```

AWS Powershell Tools (use IAM roles as before...)

```
<powershell>
    Read-S3Object -BucketName myS3Bucket
        -Key myFolder/myFile.zip
        -File c:\destinationFile.zip
</powershell>
```

**Automation**  
Less fingers, less mistakes

**Security**  
Instances locked down by default

**Availability**  
Drive higher availability with self-healing

**Flexible**  
Shell, Powershell, CloudFormation,Chef, Puppet, OpsWorks

**Scale**  
Manage large scale deployments and drive autoscaling

# Why do this?

## Some does and don'ts

### Do

Use IAM roles

Go keyless if you can

Strike a balance between  
AMI and dynamic  
bootstrapping

## Some does and don'ts

### Do

- Use IAM roles
- Go keyless if you can
- Strike a balance between  
AMI and dynamic  
bootstrapping

### Don't

- Put your API access keys  
into code (and then publish  
to GIT) or bake into AMIs  
(and share)



# Block storage

Understanding instance storage vs EBS

## Instance Storage

Local ‘on host’ disk  
volumes

Data dependent upon  
instance lifecycle

## Instance Storage

Local ‘on host’ disk  
volumes

Data dependent upon  
instance lifecycle

vs

## Elastic Block Storage

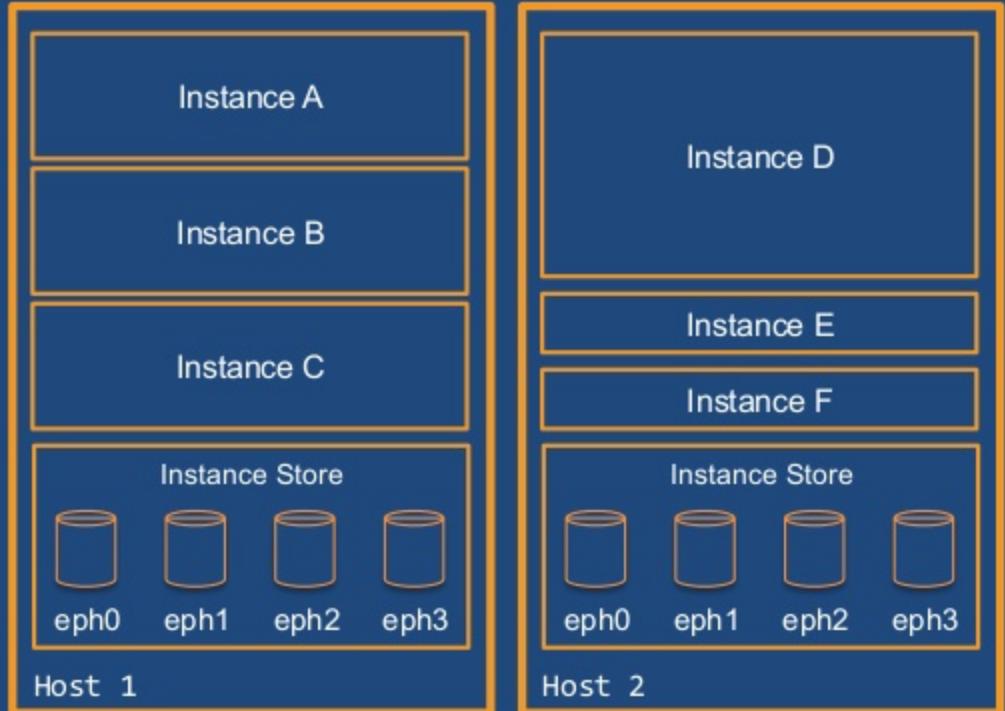
Network attached optimised  
block storage

Data independent of instance  
lifecycle

## Instance Storage

Local 'on host' disk volumes

Data dependent upon instance lifecycle



## Instance Storage

Local ‘on host’ disk volumes

Data dependent upon instance lifecycle

If an instance reboots (intentionally or unintentionally), data in the instance store persists

Data on instance store volumes is **lost** under the following circumstances:

- Failure of an underlying drive
- Stopping an Amazon EBS-backed instance
- Terminating an instance

# Instance Storage



## Options

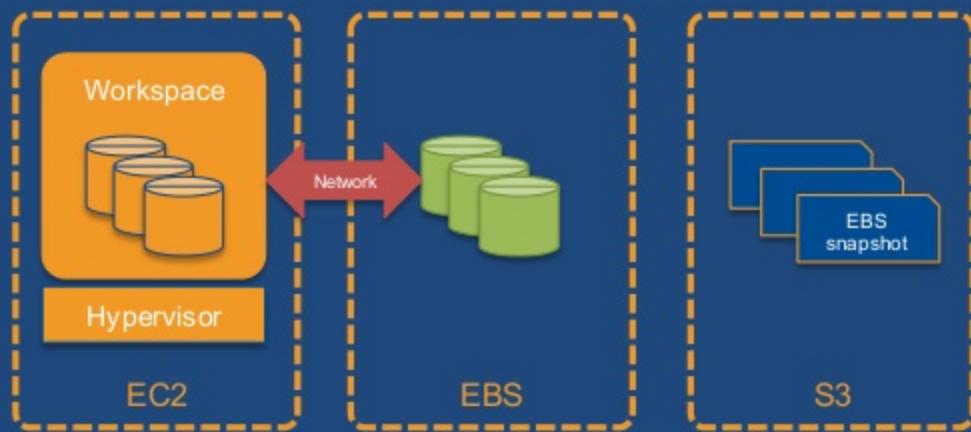
Differing types of instance storage

General purpose	m3.medium	1 x 4 SSD <sup>*6</sup>	Compute optimized	c1.xlarge	4 x 420
General purpose	m3.large	1 x 32 SSD <sup>*6</sup>	Compute optimized	cc2.8xlarge	4 x 840
General purpose	m3.xlarge	2 x 40 SSD <sup>*6</sup>	GPU instances	g2.2xlarge	1 x 60 SSD
General purpose	m3.2xlarge	2 x 80 SSD <sup>*6</sup>	GPU instances	cg1.4xlarge	2 x 840
General purpose	m1.small	1 x 160	Memory optimized	m2.xlarge	1 x 420
General purpose	m1.medium	1 x 410	Memory optimized	m2.2xlarge	1 x 850
General purpose	m1.large	2 x 420	Memory optimized	m2.4xlarge	2 x 840
General purpose	m1.xlarge	4 x 420	Memory optimized	cr1.8xlarge	2 x 120 SSD
Compute optimized	c3.large	2 x 16 SSD	Storage optimized	i2.xlarge	1 x 800 SSD
Compute optimized	c3.xlarge	2 x 40 SSD	Storage optimized	i2.2xlarge	2 x 800 SSD
Compute optimized	c3.2xlarge	2 x 80 SSD	Storage optimized	i2.4xlarge	4 x 800 SSD
Compute optimized	c3.4xlarge	2 x 160 SSD	Storage optimized	i2.8xlarge	8 x 800 SSD
Compute optimized	c3.8xlarge	2 x 320 SSD	Storage optimized	hs1.8xlarge	24 x 2,048 <sup>*3</sup>
Compute optimized	c1.medium	1 x 350	Storage optimized	hi1.4xlarge	2 x 1,024 SSD

One or more ephemeral  
(temporary) drives  
(instance storage)

One or more EBS  
(persistent) drives

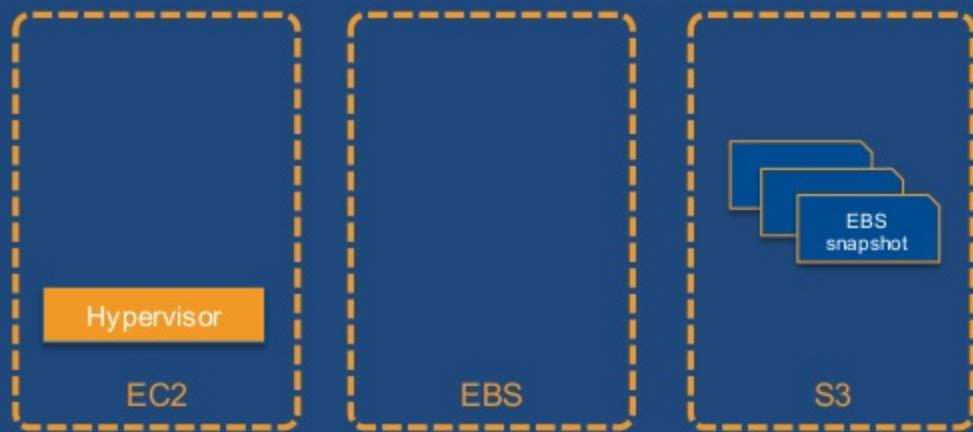
EBS snapshots  
(backup images)



## Elastic Block Storage

Network attached optimised  
block storage  
Data independent of instance  
lifecycle

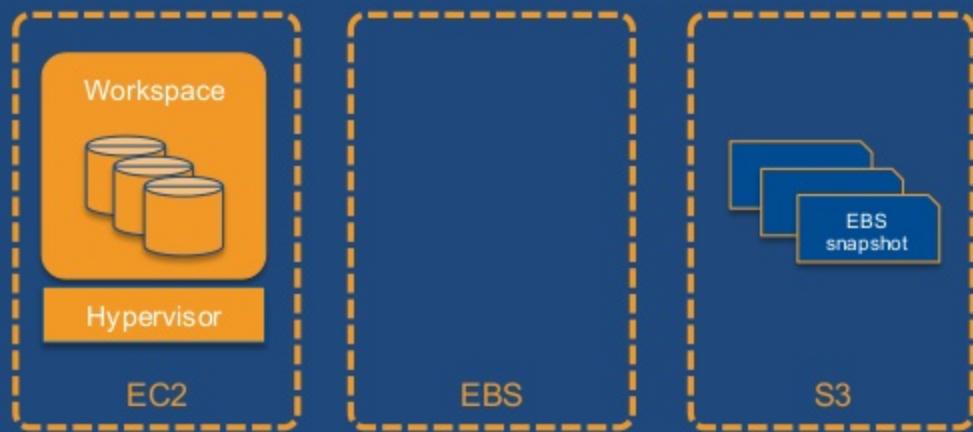
## Boot cycle



## Elastic Block Storage

Network attached optimised  
block storage  
Data independent of instance  
lifecycle

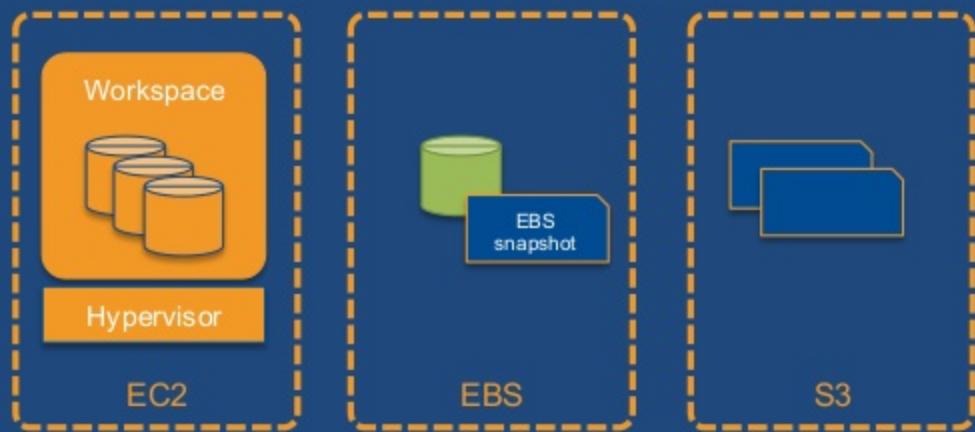
## Boot cycle



## Elastic Block Storage

Network attached optimised  
block storage  
Data independent of instance  
lifecycle

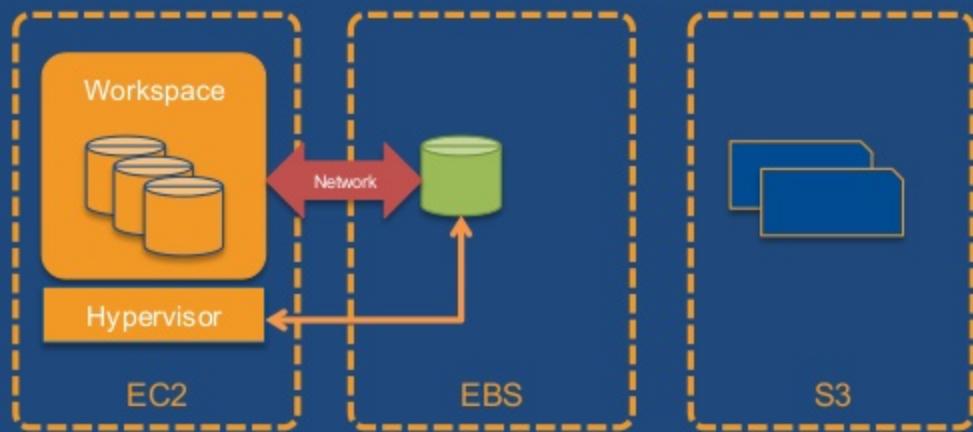
## Boot cycle



## Elastic Block Storage

Network attached optimised  
block storage  
Data independent of instance  
lifecycle

## Boot cycle



## Elastic Block Storage

Network attached optimised  
block storage  
Data independent of instance  
lifecycle

## EBS Persistence

EBS volume is off-instance storage

You pay for the volume usage as long as the data persists

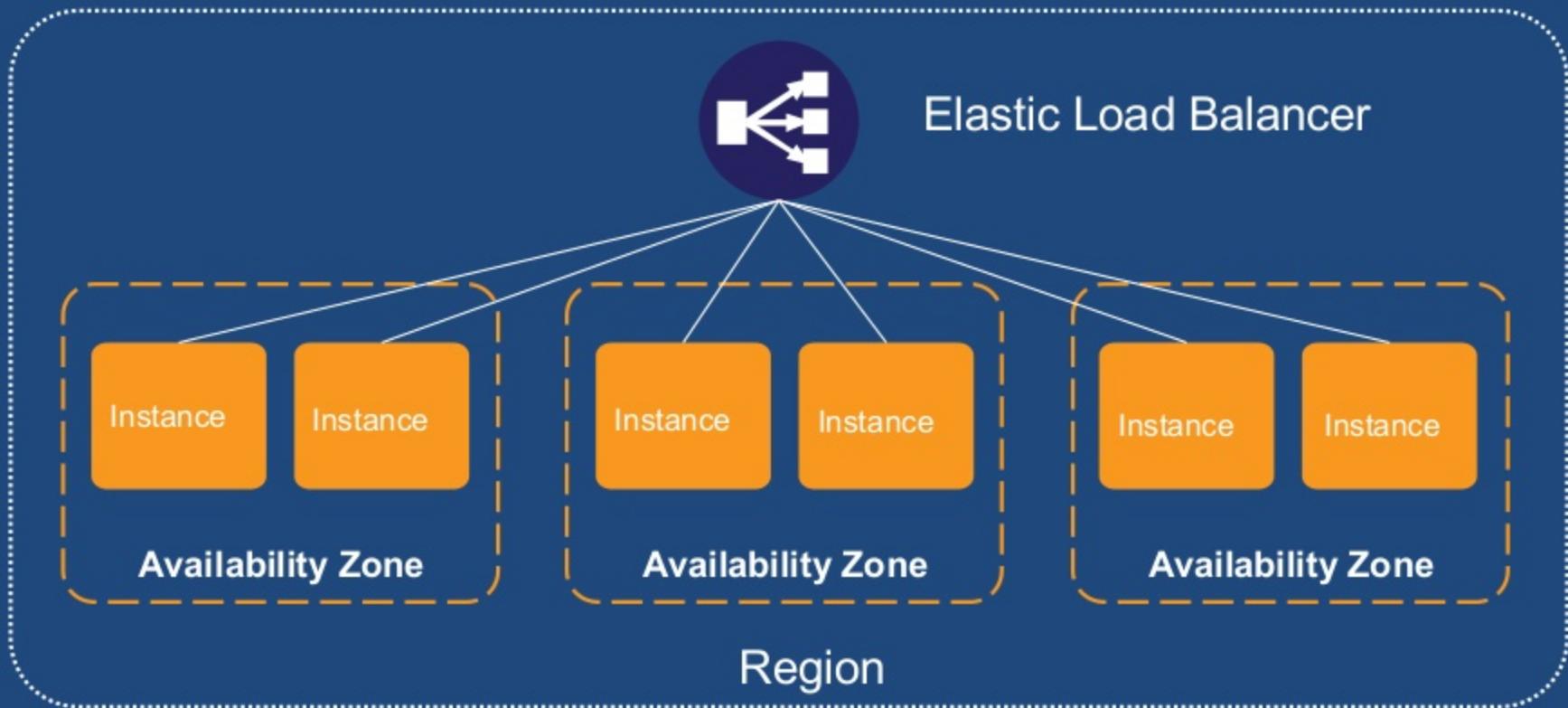
1. By default, EBS volumes that **are attached to a running instance** automatically detach from the instance with their data intact when that instance is terminated
2. By default, EBS volumes that **are created and attached to an instance at launch** are deleted when that instance is terminated. You can modify this behavior by changing the value of the flag **DeleteOnTermination** to false when you launch the instance.

# Elastic Load Balancer

Spreading the load and fronting EC2

# A regional service

Load balance across availability zones



# Elastic Load Balancing

## Spread

Go small and wide

Balance resources across  
AZs

## Offload

SSL processing on ELB

Remove load from EC2  
instances

## Health check

Choose the right healthcheck  
point

Check whole layers

1. Persistent HTTP connections – enable them and ELB to Server will be optimized
2. Never address underlying IP – always DNS name
  - *There's a set behind an ELB and real clients spread across them*
  - *They will change as the ELB scales to keep ahead of demand*
3. If you span ELB across AZs have an instance in all Azs
4. De-register instances from an ELB before terminating

# Trusted Advisor

No issue detected

Investigation Recommended

Action Recommended

Not Available

## Summary

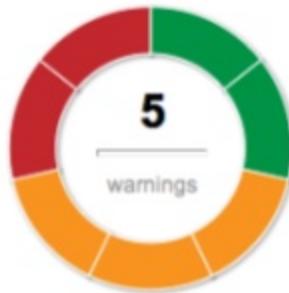
**\$3,861**  
In potential annual savings



**Cost Optimizing**  
Suppressed( 0 )

## Cost Optimizing

**21**  
Opportunities to enhance security



**Security**  
Suppressed( 0 )

## Security

Rec

**EC2 Availability Zone Balance**  
Summary:  
2 regions have an imbalanced instance distribution across availability zones



**Fault Tolerance**  
Suppressed( 0 )

## Fault Tolerance

## Performance

**0**  
Opportunities to improve performance



**Performance**  
Suppressed( 0 )

# AutoScaling

Automate EC2 commissioning and  
decommissioning

## Launch Configuration

Describes what Auto Scaling will create when adding Instances

*AMI  
Instance Type  
Security Group  
Instance Key Pair*

Only one active launch configuration at a time

Auto Scaling will terminate instances with old launch configuration first rolling update

## Auto-Scaling Group

Auto Scaling managed grouping of EC2 instances

Automatic health check to maintain pool size

Automatically scale the number of instances by policy – Min, Max, Desired

Automatic Integration with ELB

Automatic distribution & balancing across AZs

## Auto-Scaling Policy

Parameters for performing an Auto Scaling action

Scale Up/Down and by how much

*ChangeInCapacity (+/- #)  
ExactCapacity (#)  
ChangeInPercent (+/- %)*

Cool Down (seconds)

Policy can be triggered by CloudWatch events

Create a launch configuration:

```
aws autoscaling create-launch-configuration  
  --image-id ami-54cf5c3d  
  --instance-type m1.small  
  --key-name mykey  
  --security-groups webservers  
  --launch-configuration-name 101-launch-config
```

Create a launch configuration:

```
aws autoscaling create-launch-configuration  
  --image-id ami-54cf5c3d  
  --instance-type m1.small  
  --key-name mykey  
  --security-groups webservers  
  --launch-configuration-name 101-launch-config
```

The usual  
suspects

CLI Tools

Create an auto scaling group:

```
aws autoscaling create-auto-scaling-group  
  --auto-scaling-group-name 101-as-group  
  --availability-zones us-east-1a us-east-1b us-east-1c  
  --launch-configuration-name 101-launch-config  
  --load-balancer-names myELB  
  --max-size 5  
  --min-size 1
```

Create an auto scaling group:

```
aws autoscaling create-auto-scaling-group  
  --auto-scaling-group-name 101-as-group  
  --availability-zones us-east-1a us-east-1b us-east-1c  
  --launch-configuration-name 101-launch-config  
  --load-balancer-names myELB  
  --max-size 5  
  --min-size 1
```



What's going to launch

Create an auto scaling group:

```
aws autoscaling create-auto-scaling-group  
  --auto-scaling-group-name 101-as-group  
  --availability-zones us-east-1a us-east-1b us-east-1c  
  --launch-configuration-name 101-launch-config  
  --load-balancer-names myELB  
  --max-size 5  
  --min-size 1
```



Integrate with an ELB?

Create an auto-scaling policy (scale up):

```
aws autoscaling put-scaling-policy  
  --policy-name 101ScaleUpPolicy  
  --auto-scaling-group-name 101-as-group  
  --scaling-adjustment 1  
  --adjustment-type ChangeInCapacity  
  --cooldown 300
```

Create an auto-scaling policy (scale up):

```
aws autoscaling put-scaling-policy  
  --policy-name 101ScaleUpPolicy  
  --auto-scaling-group-name 101-as-group  
  --scaling-adjustment 1  
  --adjustment-type ChangeInCapacity  
  --cooldown 300
```



Period before another action will take place  
(Damper)

Create an auto-scaling policy (scale down):

```
aws autoscaling put-scaling-policy  
  --policy-name 101ScaleDownPolicy  
  --auto-scaling-group-name 101-as-group  
  "--scaling-adjustment=-1"  
  --adjustment-type ChangeInCapacity  
  --cooldown 300
```

EC2 Management Console

Services EC2 S3 RDS CloudFront Route 53 Edit ianmas @ ianmas-aws Ireland Help

EC2 Dashboard  
Events  
Tags  
**INSTANCES**  
Instances  
Spot Requests  
Reserved Instances  
**IMAGES**  
AMIs  
Bundle Tasks  
**ELASTIC BLOCK STORE**  
Volumes  
Snapshots  
**NETWORK & SECURITY**  
Security Groups  
Elastic IPs  
Placement Groups  
Load Balancers  
Key Pairs  
Network Interfaces  
**AUTO SCALING**  
Launch Configurations  
**Auto Scaling Groups**

Welcome to Auto Scaling

You can use Auto Scaling to manage Amazon EC2 capacity automatically, maintain the right number of instances for your application, operate a healthy group of instances, and scale it according to your needs. [Learn more](#)

**Create Auto Scaling group**

Note: To create your Auto Scaling groups in a different region, select your region from the navigation bar.

**Benefits of Auto Scaling**

**Reusable Instance Templates**  
  
Provision Instances based on a reusable template you define, called a launch configuration.  
[Learn more](#)

**Automated Provisioning**  
  
Keep your Auto Scaling group healthy and balanced, whether you need one instance or 1,000.  
[Learn more](#)

**Adjustable Capacity**  
  
Maintain a fixed group size or adjust dynamically based on Amazon CloudWatch metrics.  
[Learn more](#)

Additional Information

Getting Started Guide  
Documentation  
All EC2 Resources  
Forums  
Pricing  
Contact Us

© 2008 - 2014, Amazon Web Services, Inc. or its affiliates. All rights reserved. [Privacy Policy](#) [Terms of Use](#) [Feedback](#)

# CloudWatch

Know what is going on

## Create Alarm

X

1. Select Metric    2. Define Alarm

Browse Metrics

Search Metrics

X

### CloudWatch Metrics by Category

Your CloudWatch metric summary has loaded. Total metrics: 76

EC2 Metrics: 49

Per-Instance Metrics: 21  
By Auto Scaling Group: 7  
By Image (AMI) Id: 7  
Aggregated by Instance Type: 7  
Across All Instances: 7

ELB Metrics: 24

Per-LB Metrics: 3  
Per LB, per AZ Metrics: 6  
By Availability Zone: 8  
Across All LBs: 3  
Namespace: 3  
Service: 3

SNS Metrics: 3

Topic Metrics: 3

Update Graph



Cancel

Back

Next

Create Alarm

## Create Alarm

1. Select Metric    2. Define Alarm

Browse Metrics

Search Metrics

EC2 > Across All Instances

< < 1 to 7 of 7 Metrics > >

Showing all results (7) for EC2 > Across All Instances.

Select All | Clear

EC2 > Across All Instances

Metric Name

CPUUtilization

DiskReadBytes

DiskReadOps

DiskWriteBytes

DiskWriteOps

NetworkIn

NetworkOut

CPUUtilization (Percent)

Average

5 Minutes



Update Graph



Time Range

Relative   Absolute   UTC (GMT)

From: 12 hours ago

To: 0 minutes ago

Zoom: 1h | 3h | 6h | 12h | 1d | 3d | 1w | 2w



Cancel

Back

Next

Create Alarm

# Cloud Watch Alarm:

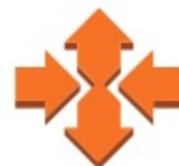


CPU  $\geq$  50% for 5 mins

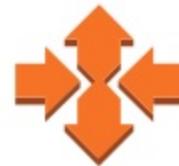


CPU < 30% for 10 mins

# Takes action:



Scale up policy



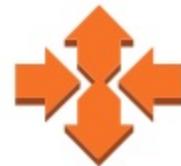
Scale down policy

Cloud Watch Alarm:



CPU  $\geq$  50% for 5 mins

Takes action:



Scale up policy



Services



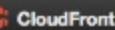
EC2



S3



RDS



CloudFront

Edit

ianmas @ ianmas-aws

In

1. Configure Auto Scaling group details

2. Configure scaling policies

3. Configure Notifications

4. Review

## Create Auto Scaling Group

You can optionally add scaling policies if you want to adjust the size (number of instances) of your group automatically. A scaling policy is a set of instructions for making such adjustments in response to an Amazon CloudWatch alarm that you assign to it. In each policy, you can choose to add or remove a specific number of instances or a percentage of the existing group size, or you can set the group to an exact size. When the alarm triggers, it will execute the policy and adjust the size of your group accordingly. [Learn more](#) about scaling policies.

 Keep this group at its initial size Use scaling policies to adjust the capacity of this groupScale between  and  instances. These will be the minimum and maximum number of instances in the group.

### Increase Group Size

Name: Execute policy when:  Take the action:   instancesAnd then wait:  seconds before allowing another scaling activity

### Decrease Group Size

Name: Execute policy when:  Take the action:   instancesAnd then wait:  seconds before allowing another scaling activity

## Create Alarm

You can use CloudWatch alarms to be notified automatically whenever metric data reaches a level you define.

To edit an alarm, first choose whom to notify and then define when the notification should be sent.

 Send a notification to:  Whenever:  of Is:   PercentFor at least:  consecutive period(s) of Name of alarm: 

CPU Utilization Percent



# Cloud Watch Alarm:



CPU  $\geq$  50% for 5 mins



CPU < 30% for 10 mins

# Takes action:



SNS Topic



Deliver message to Q



Post to endpoint



Send Email

**Comprehensive**  
Billing, technical, aggregate &  
custom metrics

**SNS  
Integration**  
Push alarms to  
SNS topics

**Alarms**  
Set custom alarms  
and thresholds

# CloudWatch

**HTTP**  
Poke HTTP  
endpoints for  
custom alarm  
actions

**Custom Metrics**  
Write your own metrics in  
via SDKs

**Email  
integration**  
Send alarm  
notifications to  
emails

Other topics to look at:

## Other topics...

### Resource tagging

Tag resources like EC2 and have it appear on billing reports

### Route 53

Front EC2 and ELBs with Route 53 for control over DNS

### Rolling deployments

Use Route 53 and ELBs to do rolling deployments, A/B testing

## Other topics...

### Beanstalk

Manage an entire  
autoscaling stack for  
popular containers such  
as ruby, python etc

### OpsWorks

Manage stacks as layers  
and implement Chef recipes  
to automate EC2  
configuration

### CloudFormation

Template everything from  
configuration of CloudWatch  
alarms, SNS topics, EC2  
instances

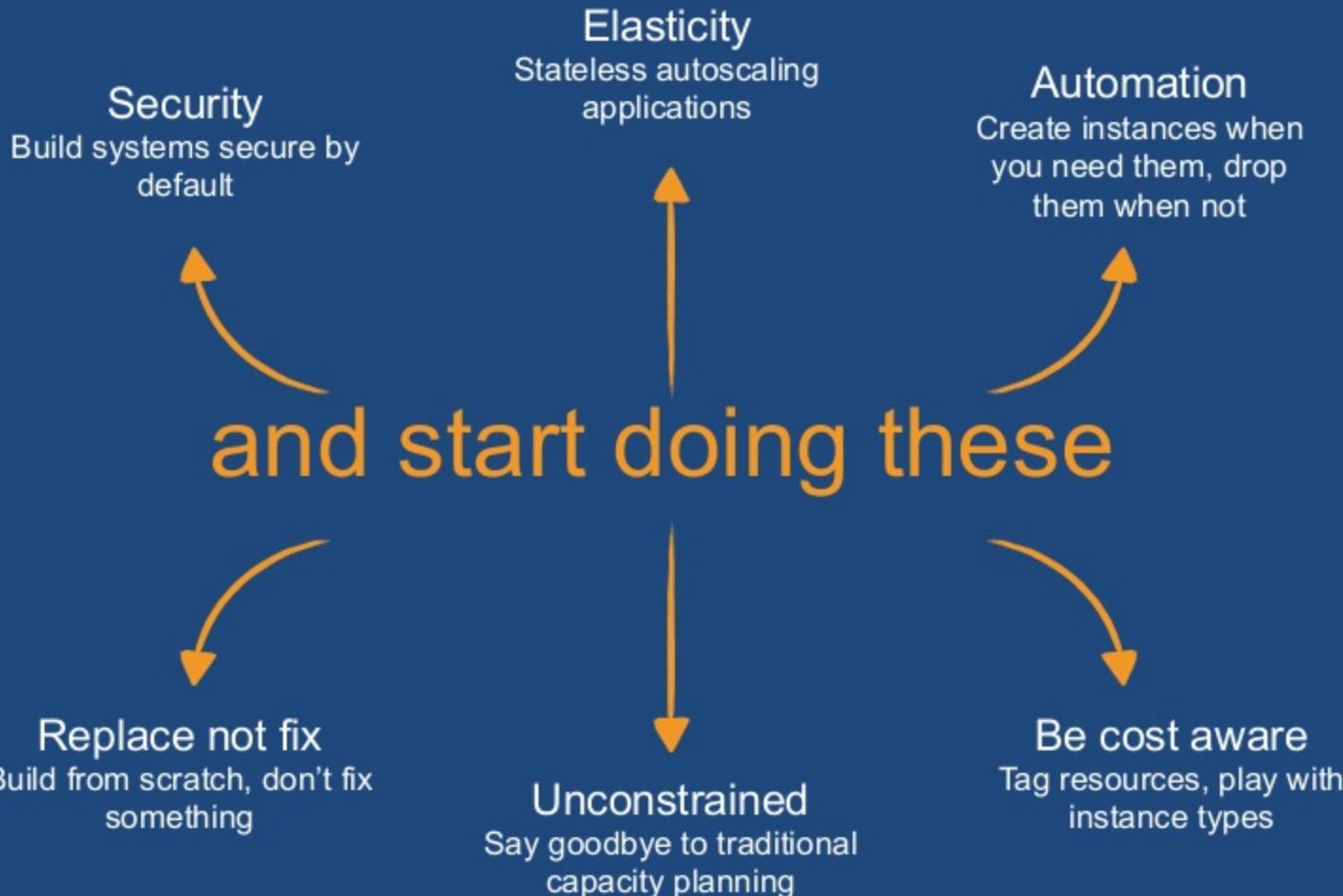
# Summary

# Stop doing these:

Provisioning and fixing servers

Treating compute as physical things

Thinking of compute as a finite commitment



aws.amazon.com

## Some Resources and Notes

### Stickers & Badges

I have some AWS Stickers and pin badges! Ask me if you want some.

### Regional Account Manager

We have an AWS Account Manager covering the Edinburgh area.

Rebeca is her name. Ask me if you want her details

### AWS RoadShow & Lunch&Learn

The AWS Roadshow is in Edinburgh on 17<sup>th</sup> June at the Apex Hotel

Lunch&Learn at Codebase the same day

+ AWS CREDITS

Follow us for more  
events & webinars



Ian Massingham – Technical Evangelist

 @IanMmmm



@AWS\_UKI for local AWS events & news



@AWScloud for Global AWS News and Announcements

# AWS Training & Certification

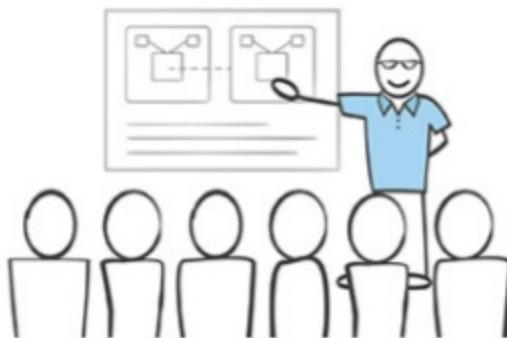
## Self-Paced Labs



Try products, gain new skills, and get hands-on practice working with AWS technologies

[aws.amazon.com/training/  
self-paced-labs](https://aws.amazon.com/training/self-paced-labs)

## Training



Skill up and gain confidence to design, develop, deploy and manage your applications on AWS

[aws.amazon.com/training](https://aws.amazon.com/training)

## Certification



Demonstrate your skills, knowledge, and expertise with the AWS platform

[aws.amazon.com/certification](https://aws.amazon.com/certification)

# We typically see customers start by trying our services

All Products

Compute & Networking

Storage

Database

Application Services

Development & Management

AWS Marketplace Software

FAQ »

Find answers to common questions about the AWS Free Tier.



## Amazon EC2 »

Web service that provides resizable compute capacity in the cloud.



## Amazon S3 »

Highly-scalable, reliable, and low-latency data storage.



## Amazon RDS »

Managed MySQL, Oracle and SQL Server databases.



## Amazon CloudWatch »

Monitoring for AWS cloud resources and applications.



## AWS Data Pipeline »

Orchestration for data-driven workflows.



## Amazon DynamoDB »

Fully managed NoSQL database service with seamless scalability.



## Amazon EBS »

Highly available, highly reliable, predictable storage volumes.



## Amazon ELB »

Web service that provides scalability and high availability.



## Amazon ElastiCache »

Managed scale-out caching.



## Amazon SNS »

Web service to set up, operate, and send notifications from the cloud.



## Amazon Elastic Transcoder »

Convert your media files easily, at low cost and at scale.



## Amazon SWF »

Workflow service for building scalable, resilient applications.



## AWS Marketplace »

Partner software pre-configured to run on AWS.



## Amazon SQS »

Scalable queue for storing messages as they travel between computers.

Get started now at : [aws.amazon.com/getting-started](https://aws.amazon.com/getting-started)



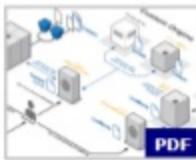
# Design your application for the AWS Cloud

## AWS Reference Architectures

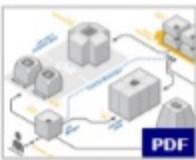
The flexibility of AWS allows you to design your application architectures the way you like. AWS Reference Architecture Datasheets provide you with the architectural guidance you need in order to build an application that takes full advantage of the AWS cloud. Each datasheet includes a visual representation of the architecture and basic description of how each service is used.



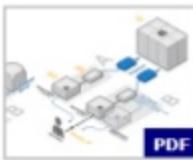
**Web Application Hosting**  
Build highly-scalable and reliable web or mobile-web applications ([PDF](#))



**Content and Media Serving**  
Build highly reliable systems that serve massive amounts of content and media ([PDF](#))



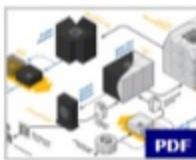
**Batch Processing**  
Build auto-scalable batch processing systems like video processing pipelines ([PDF](#))



**Fault tolerance and High Availability**  
Build systems that quickly failover to new instances in an event of failure ([PDF](#))



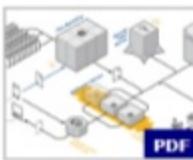
**Large Scale Processing and Huge Data sets**  
Build high-performance computing systems that involve Big Data ([PDF](#))



**Ad Serving**  
Build highly-scalable online ad serving solutions ([PDF](#))



**Disaster Recovery for Local Applications**  
Build cost-effective Disaster Recovery solutions for on-premises applications ([PDF](#))



**File Synchronization**  
Build simple file synchronization service ([PDF](#))

More details on the AWS Architecture Center at : [aws.amazon.com/architecture](http://aws.amazon.com/architecture)

