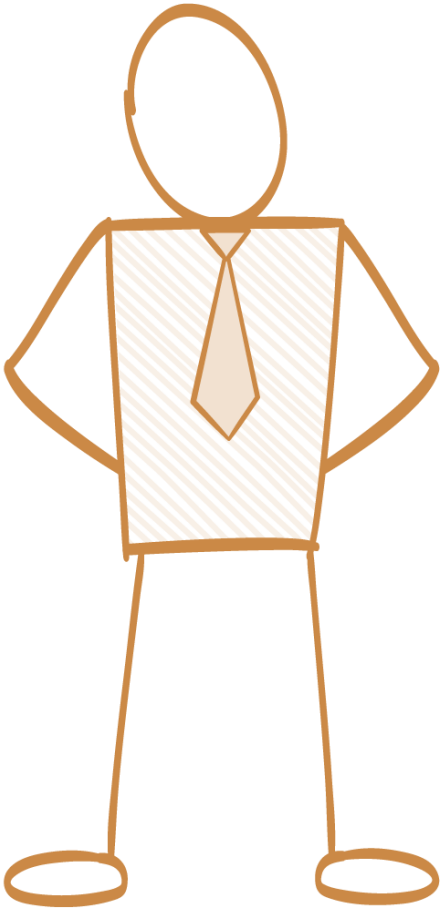
Architecting on AWS
Student Guide
Version 3.1

100-ARC-31-EN-SG



Module 7: Elasticity, Scalability, and Bootstrapping

Topics

- Basic tenets of AWS
- Patterns and (anti-patterns) for creating scalable architectures in AWS
- EC2 Instances
- Components of Auto Scaling

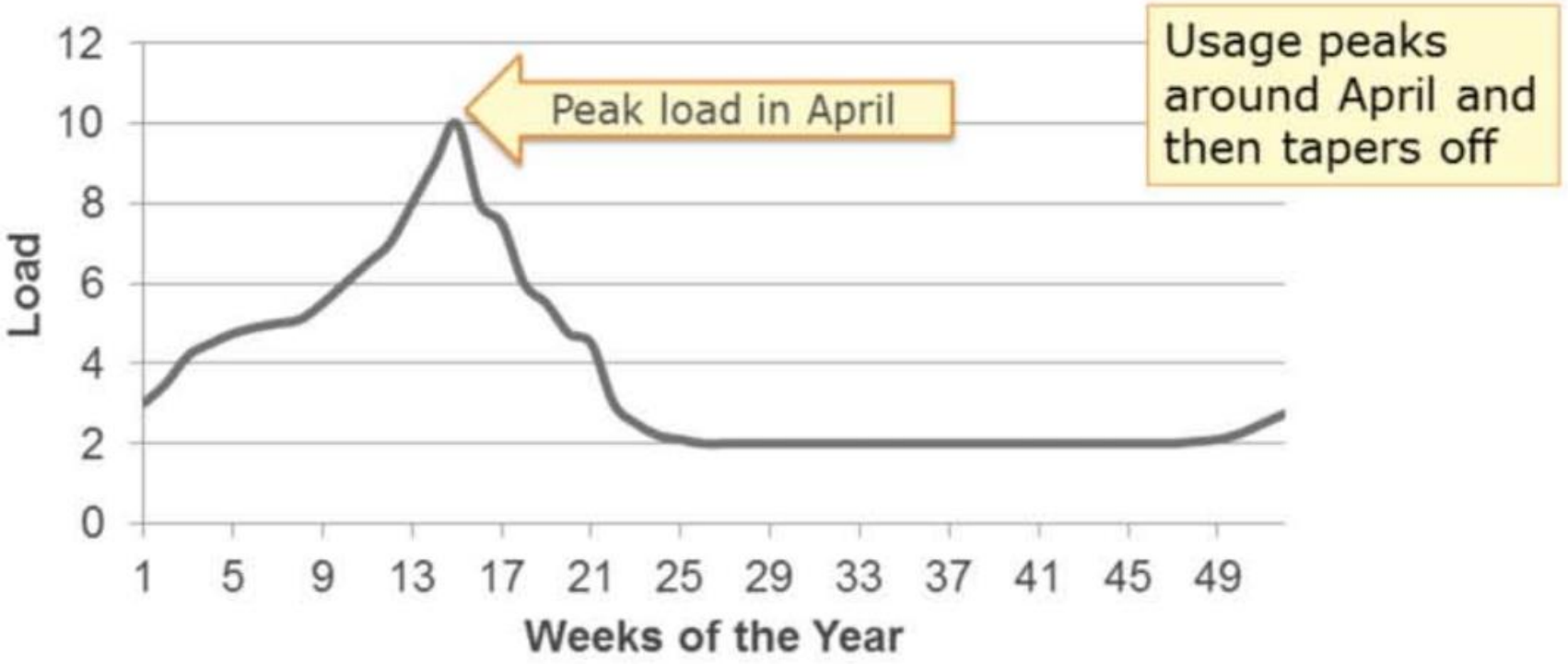
Topics

- Basic tenets of AWS
  - Review how traditional architectures accommodate expected load variation
  - Anti-patterns for elastic, scalable architectures
  - 4 patterns for elastic, scalable architectures
- Patterns and (anti-patterns) for creating scalable architectures in AWS
- EC2 Instances
- Components of Auto Scaling

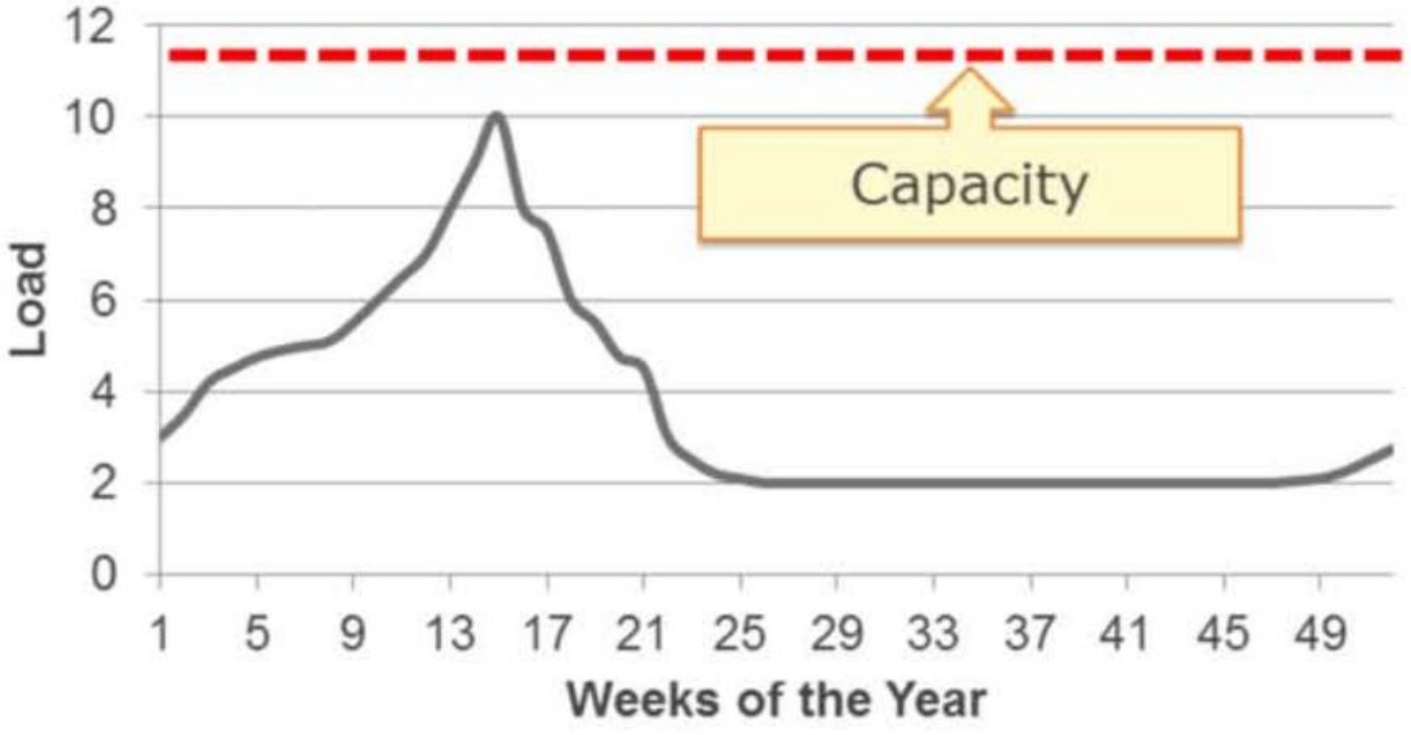
Expected Load Variation

- Non-cloud systems typically over-provision and under-utilize
- Under-utilization costs:
  - Capital
  - Space
  - Power
  - Cooling
  - Maintenance

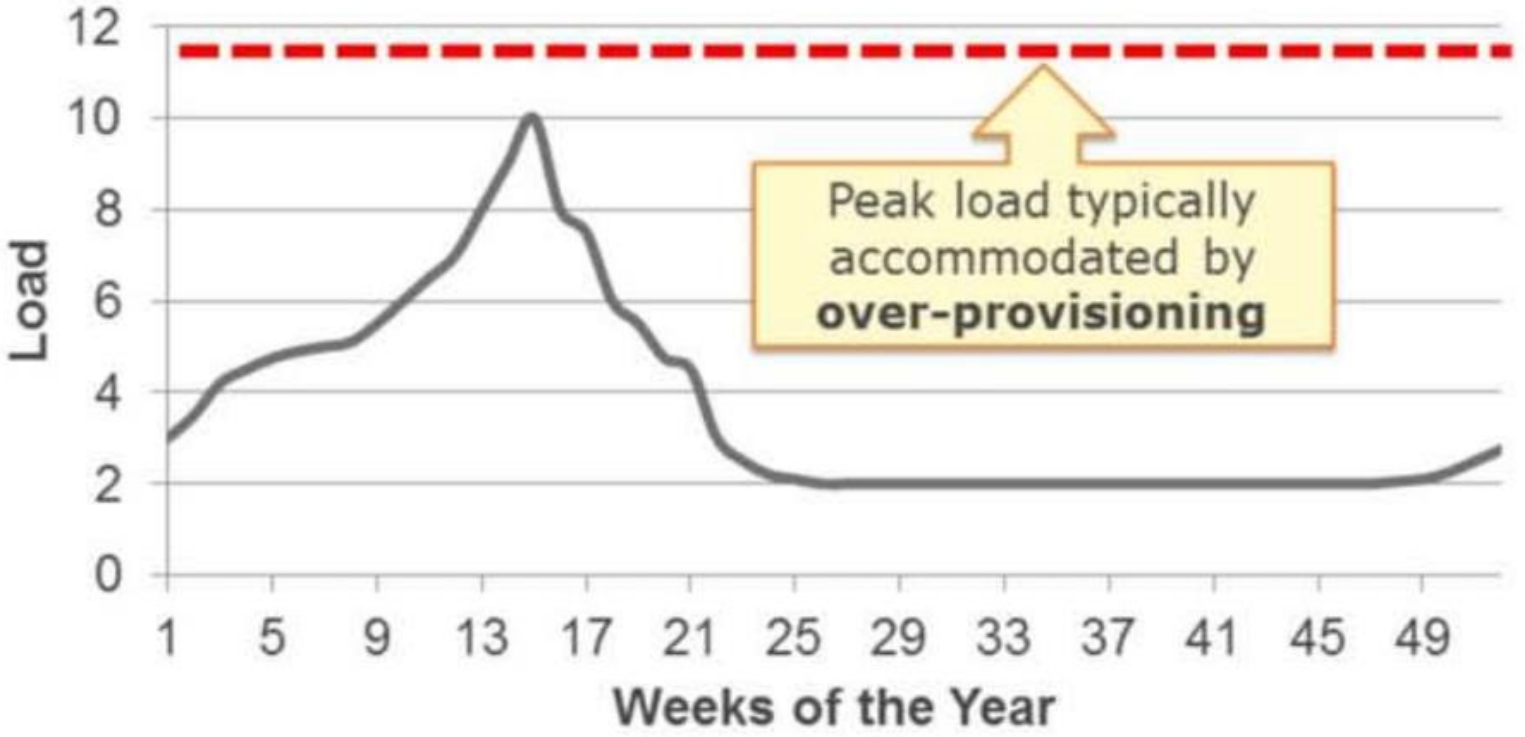
Expected Load Variation Example (1 of 6)



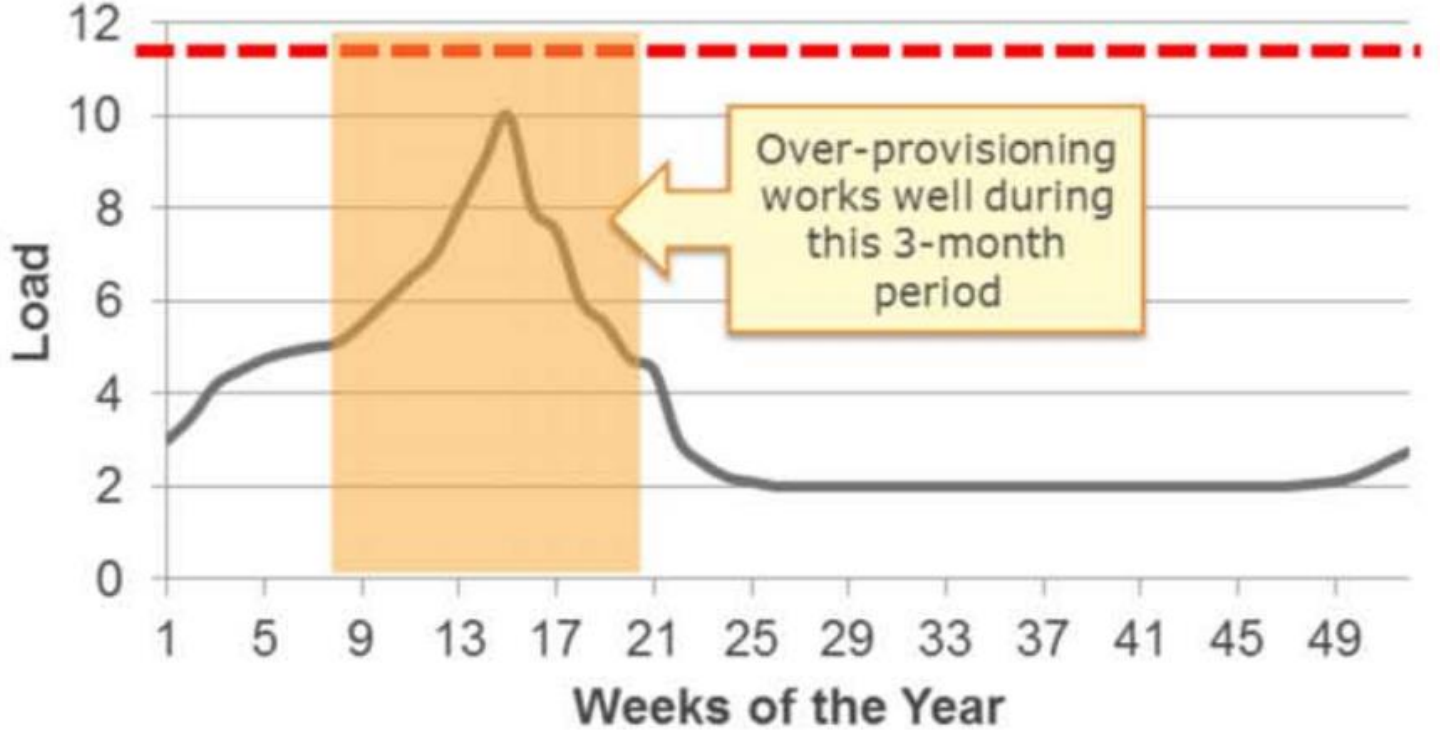
Expected Load Variation Example (2 of 6)



Expected Load Variation Example (3 of 6)



Expected Load Variation Example (4 of 6)



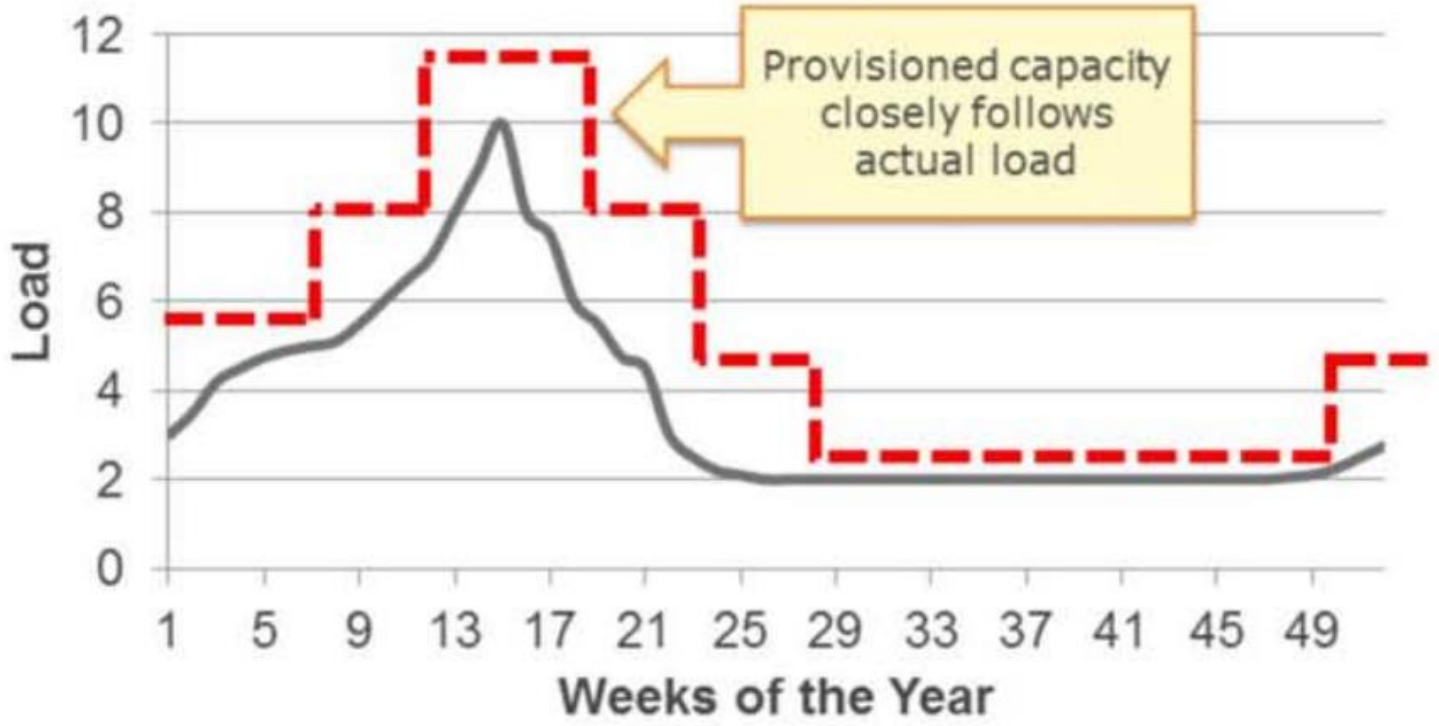
Expected Load Variation Example (5 of 6)

Expected Load Variation Example (6 of 6)

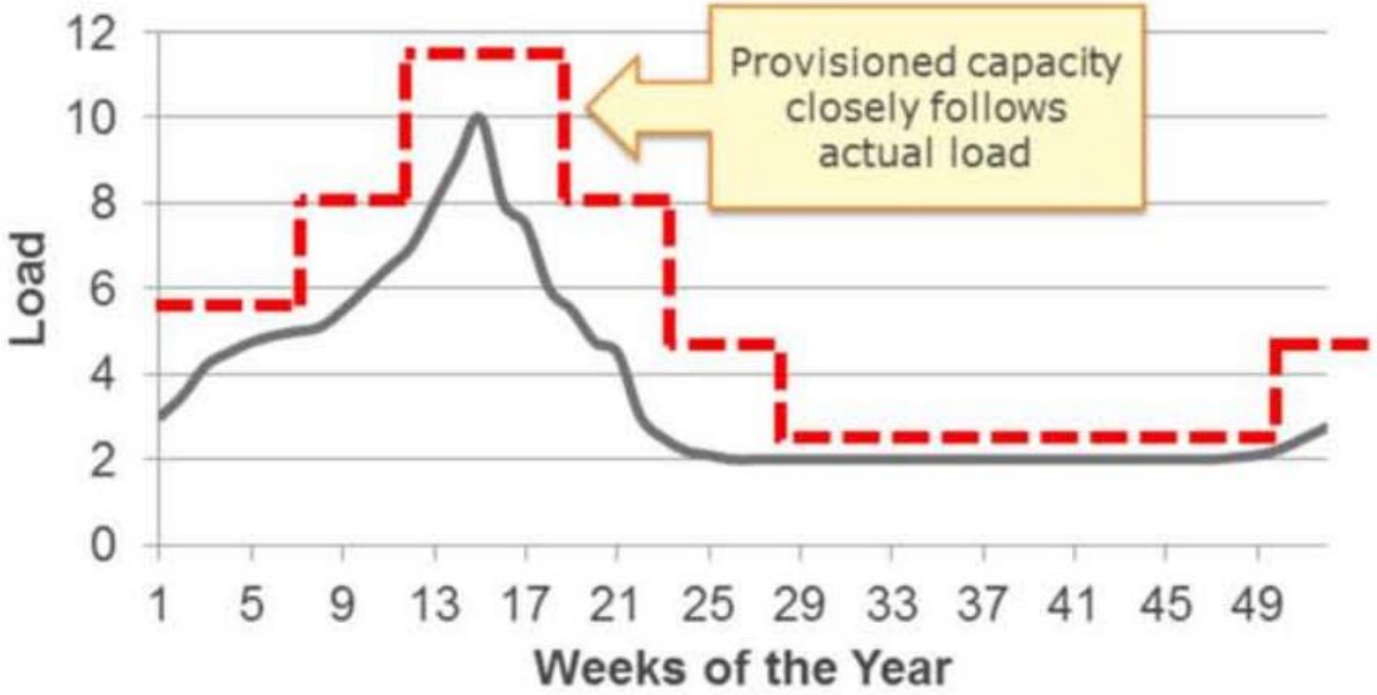
Elastic environments

- Highly utilized all the time
- Provision resources “just in time”

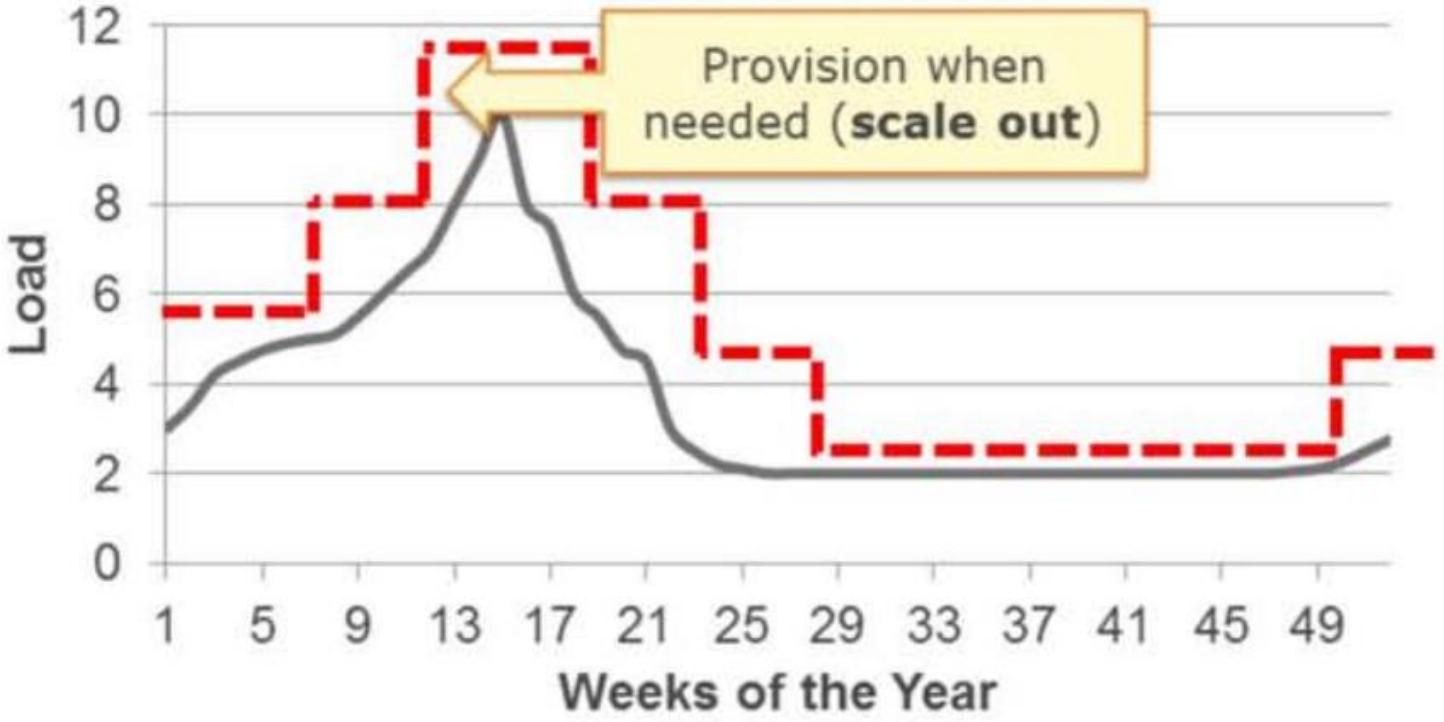
Elastic environments scenario (1 of 4)



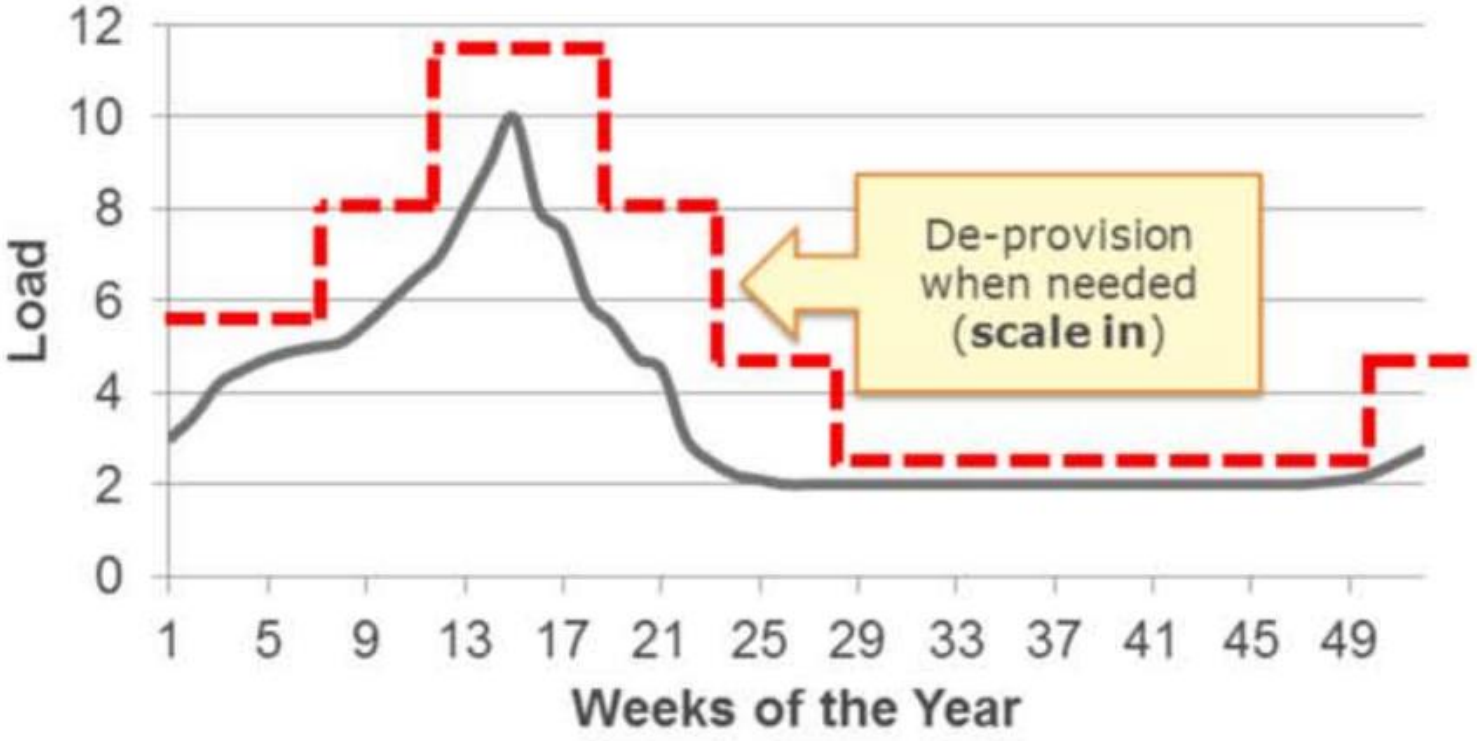
Elastic environments scenario (2 of 4)



Elastic environments scenario (3 of 4)



Elastic environments scenario (4 of 4)



Topics

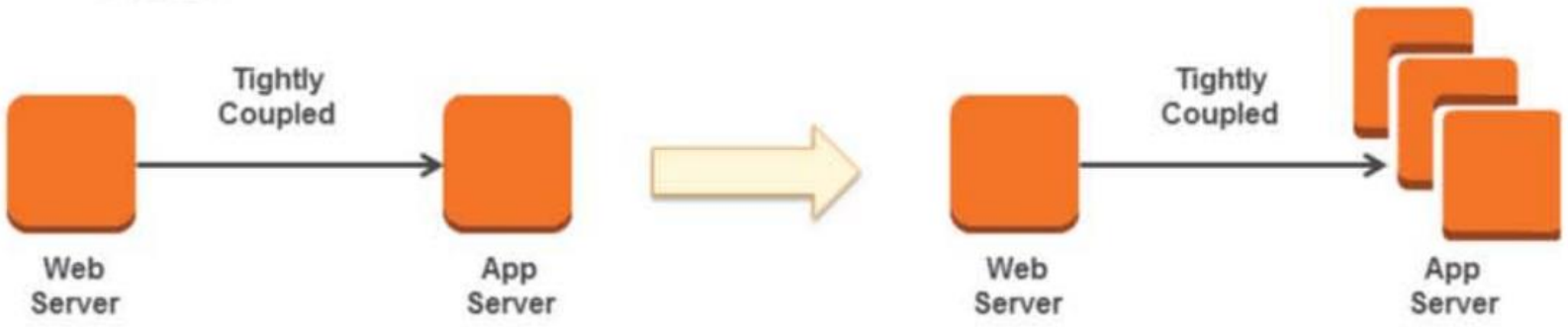
- Basic tenets of AWS
- Patterns and (anti-patterns) for creating scalable architectures in AWS
- EC2 Instances
- Components of Auto Scaling

Patterns and anti-patterns

- Anti-Pattern: **Manual Processes**
  - When direct, manual intervention is required to start new resources – or scale existing one – will be a blocker at scale
- Pattern: **Automated Processes**

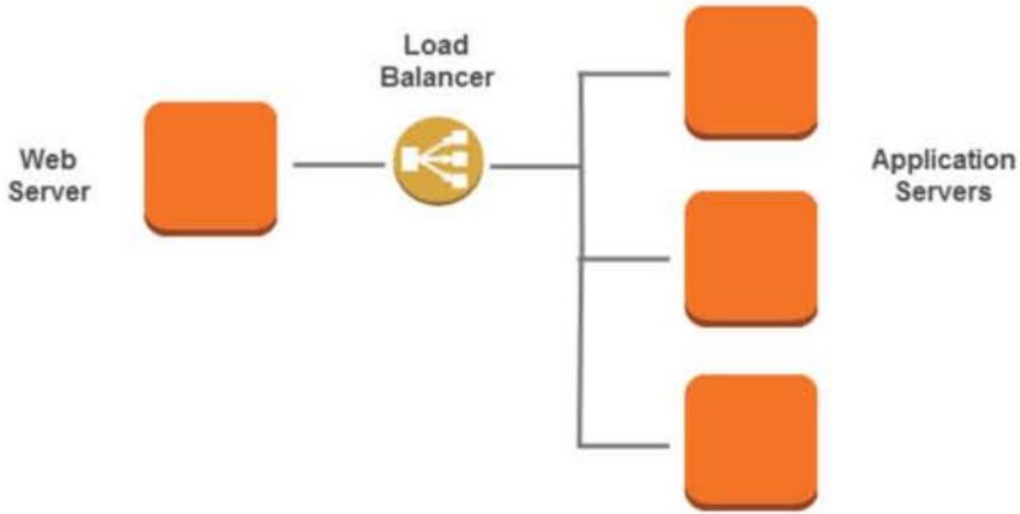
Patterns and anti-patterns: Integration

- Anti-Pattern: **Tightly-coupled**
  - Applications components in which a single unit depends on another specific single unit behave poorly when the dependency fails or needs to **subdivide (for example, grow horizontally)** to scale



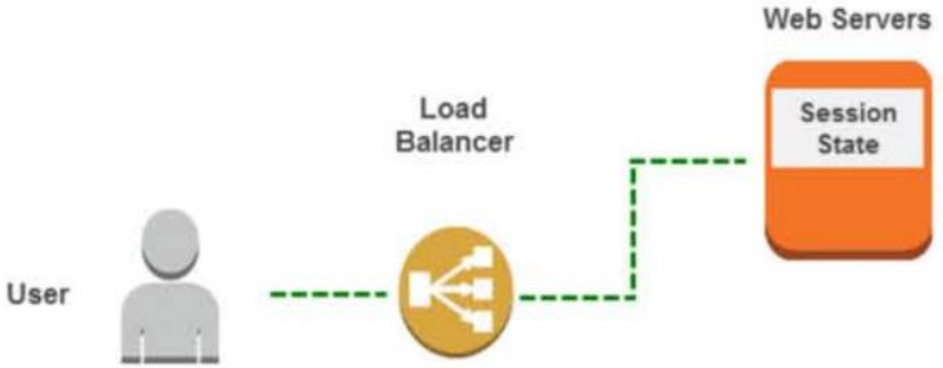
Patterns and anti-patterns: Integration (continue)

- Pattern: **Loosely-coupled**



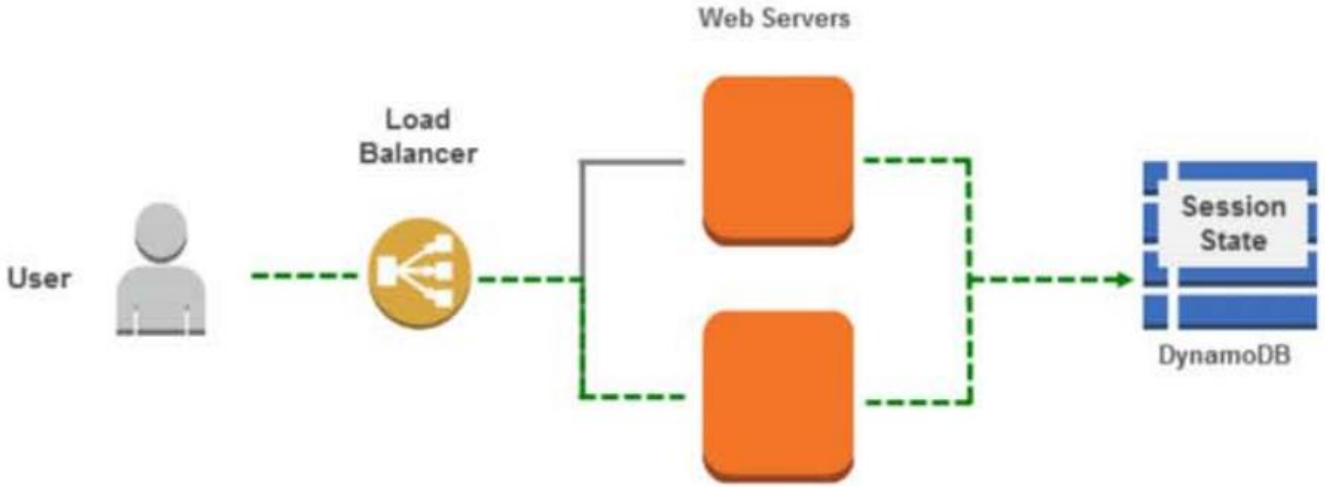
Patterns and anti-patterns: Session state

- **Anti-Pattern**: Stateful
  - Applications that store state on one instance are more challenging to scale horizontally



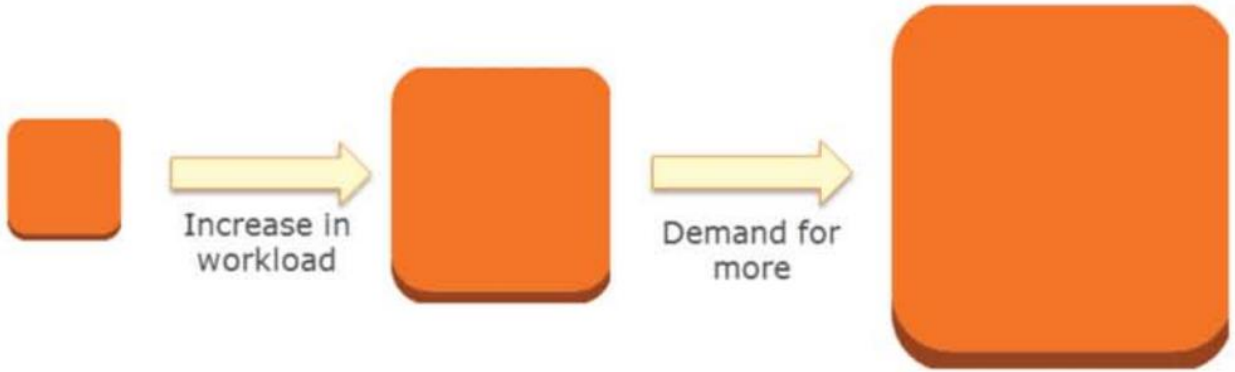
Patterns and anti-patterns: Session state (continue)

- **Pattern**: Stateless
  - Move state to a shared, accessible location



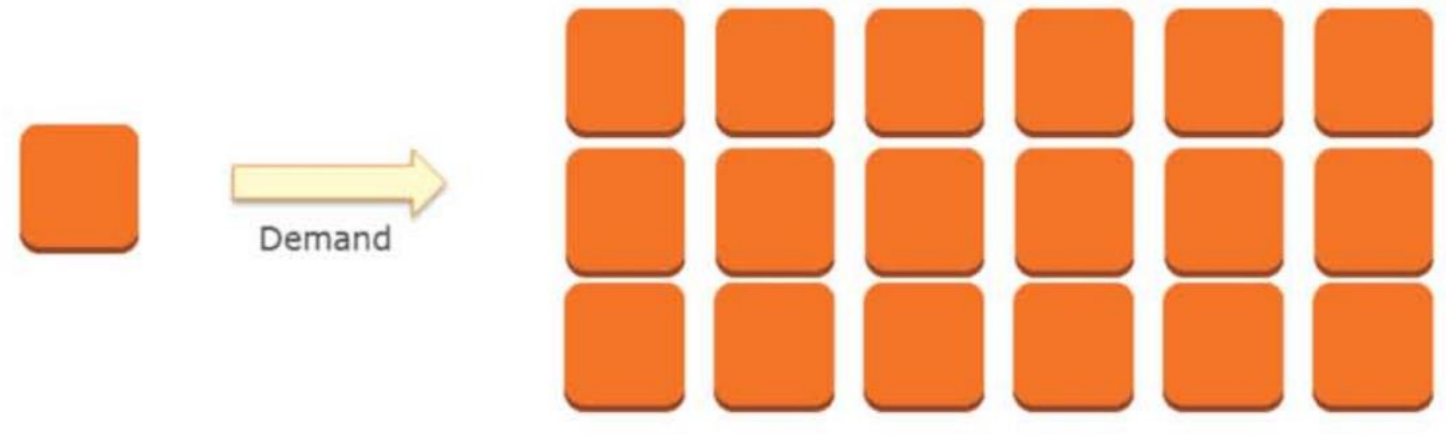
Patterns and anti-patterns

- **Anti-Pattern**: Vertical
  - Vertical scaling (more CPU, memory, and so on) will eventually run out of room



Patterns and anti-patterns

- **Pattern**: Horizontal
  - Add and remove instances as needed



Topics

- Basic tenets of AWS
- Patterns and (anti-patterns) for creating scalable architectures in AWS
- EC2 Instances
- Components of Auto Scaling

AWS Solutions Architect Associate

EC2

Elastic Cloud Compute Introduction

amazon web services

Elastic Compute Cloud (EC2)

Cloud Computing Service

Choose your **OS, Storage, Memory, Network Throughput.**

Launch and SSH into your server **within minutes**.

amazon web services

Introduction to EC2

Elastic Compute Cloud (EC2) is a **highly configurable server**.
EC2 is resizable **compute capacity.** It takes **minutes** to launch new instances.
Anything and everything on AWS uses EC2 Instance underneath.

Choose your OS **via**
Amazon Machine Image (AMI)

Choose you **Instance Type**

t2.nano
$0.0065/hour ($4.75/month)
1 vCPU 0.5GB Mem

C4.8xlarge
$1.591/hour ($1161.43/month)
36 vCPU 60GB Mem 10 Gigabit performance

Add Storage **(EBS, EFS)**

SSD **HDD** **Virtual Magnetic Tape** **Multiple Volumes**

Configure your Instance

Security Groups, Key Pairs, UserData, IAM Roles, Placement Groups

amazon web services

AWS Solutions Architect Associate

EC2

Instance Types

amazon web services

EC2 – Instance Types and Usage

General Purpose

A1 T3 T3a T2 M5 M5a M4

balance of compute, memory and networking resources

Use-cases web servers and code repositories

Compute Optimized

C5 C5n C4

Ideal for compute bound applications that benefit from high performance processor

Use-cases scientific modeling, dedicated gaming servers and ad server engines

Memory Optimized

R5 R5a X1e X1 High Memory z1d

fast performance for workloads that process large data sets in memory.

Use-cases in-memory caches, in-memory databases, real time big data analytics

Accelerated Optimized

P3 P2 G3 F1

hardware accelerators, or co-processors

Use-cases Machine learning, computational finance, seismic analysis, speech recognition

Storage Optimized

I3 I3en D2 H1

high, sequential read and write access to very large data sets on local storage

Use-cases NoSQL, in-memory or transactional databases, data warehousing

AWS Solutions Architect Associate

EC2

Instance Sizes

amazon web services

EC2 – Instance Sizes

EC2 Instance Sizes **generally double** in price and key attributes

Name	vCPU	RAM (GIB)	On-Demand per hour	On-Demand per month
t2.small	1	12	$0.023	$16.79
t2.medium	2	24	$0.0464	$33.87
t2.large	2	36	$0.0928	$67.74
t2.xlarge	4	54	$0.1856	$135.48

AWS Solutions Architect Associate

EC2

Instance Profiles

amazon web services

EC2 - Instance Profile

Instead of embedding your AWS credentials (Access Key and Secret) in your code so your Instance has permissions to access certain services you can **Attach a role to an instance** via an **Instance Profile**

You want to always avoid embedding your AWS credentials when possible.

An **Instance Profile** holds a reference to a role. The EC2 instance is associated with the Instance Profile. When you select an IAM role when Launching an EC2 instance, AWS will automatically create the Instance Profile for you. Instance Profiles are not easily viewed via the AWS Console.

AWS Solutions Architect Associate

EC2

Placement Groups

amazon web services

EC2 – Placement Groups

Placement Groups let you to choose **the logical placement** of your instances to optimize for **communication**, **performance** or **durability**. Placement groups are **free**.

Cluster

- packs instances close together inside an **AZ**
- low-latency network performance for tightly-coupled node-to-node communication
- well suited for High Performance Computing (HPC) applications
- Clusters cannot be multi-AZ

Partition

- spreads instances across logical partitions
- each partition do not share the underlying hardware with each other (rack per partition)
- well suite for large distributed and replicated workloads (Hadoop, Cassandra, Kafka)

Spread

- Each instance is placed on a different rack
- When critical instances should be keep separate from each other
- You can spread a max of 7 instances. Spreads can be multi-AZ

AWS Solutions Architect Associate

EC2

Userdata

amazon web services

EC2 – UserData

You can provide an EC2 with **UserData** which is a **script** that will be automatically run when launching an EC2 instance. You could install package, apply updates or anything you like.

This example sets up an apache web-server

From within the EC2 instance, if you were to SSH in and CURL this special URL you can see the UserData script eg. **curl** **http://169.254.169.254/latest/user-data**

amazon web services

AWS Solutions Architect Associate

EC2

Metadata

amazon web services

EC2 – MetaData

From within your EC2 instance you can access information about the EC2 via a special url endpoint at

169.254.169.254

You would SSH into your EC2 instance and can use the CURL command:

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

/public-ipv4 get the current public IPV4 address
/ami-id the AMI ID used to launch this EC2 instance
/instance-type the Instance Type of this EC2 instance

Combine metadata with userdata scripts to perform all sorts of advanced AWS staging automation

```
[ec2-user ~]$ curl
http://169.254.169.254/latest/meta-data/
ami-id
ami-launch-index
ami-manifest-path
block-device-mapping/
events/
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/
```

AWS Solutions Architect Associate

EC2

EC2 Cheat Sheet

amazon web services

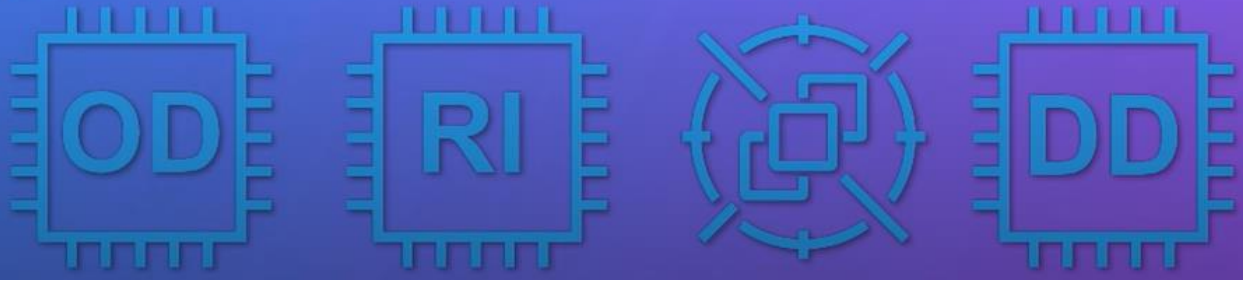
EC2 CheatSheet

- **Elastic Compute Cloud (EC2)** is a Cloud Computing Service
- Configure your EC2 by choosing your **OS, Storage, Memory, Network Throughput.**
- Launch and SSH into your server **within minutes**.
- EC2 comes in variety Instance Types specialized for different roles:
    - **General Purpose** balance of compute, memory and networking resources
    - **Compute Optimized** Ideal for compute bound applications that benefit from high performance processor
    - **Memory Optimized** fast performance for workloads that process large data sets in memory.
    - **Accelerated Optimized** hardware accelerators, or co-processors
    - **Storage Optimized** high, sequential read and write access to very large data sets on local storage
- Instance Sizes **generally double** in price and key attributes
- **Placement Groups** let you to choose the logical placement of your instances to optimize for communication, performance or durability. Placement groups are free.
- **UserData** a script that will be automatically run when launching an EC2 instance.
- **MetaData** meta data about the current instance. You access this meta data via a local endpoint when SSH'd into the EC2 instance. eg. curl http://169.254.169.254/latest/meta-data meta data could be the instance type, current ip address etc...
- **Instance Profiles** a container for an IAM role that you can use to pass role information to an EC2 instance when the instance starts.

AWS Solutions Architect Associate

EC2 Pricing Models

EC2 Pricing Introduction



amazon web services

EC2 – Pricing Model

On-Demand — Least Commitment

- low cost and flexible
- only pay per hour
- short-term, spiky, unpredictable workloads
- cannot be interrupted
- For first time apps

Spot upto 90% — Biggest Savings

- request spare computing capacity
- flexible start and end times
- Can handle interruptions (server randomly stopping and starting)
- For non-critical background jobs

Reserved upto 75% off — Best Long-term

- steady state or predictable usage
- commit to EC2 over a 1 or 3 year term
- Can resell unused reserved instances

Dedicated — Most Expensive

- Dedicated servers
- Can be on-demand or reserved (upto 70% off)
- When you need a guarantee of isolate hardware (enterprise requirements)

AWS Solutions Architect Associate

EC2 Pricing Models

On-Demand Pricing

amazon web services

EC2 - On-Demand Instances

Least Commitment

When you launch an EC2 instance it is by default using **On-Demand** Pricing

On-demand has **no up-front payment** and **no long-term commitment**

You are charged by the **hour** or by the **minute** (varies based on EC2 Instance Types)

On-Demand is for applications where the workload is for **short-term**, **spikey** or **unpredictable**.
When you have a **new app** for development or you want to run experiment.

amazon web services

AWS Solutions Architect Associate

EC2 Pricing Models

Reserved Instances (RI) Pricing

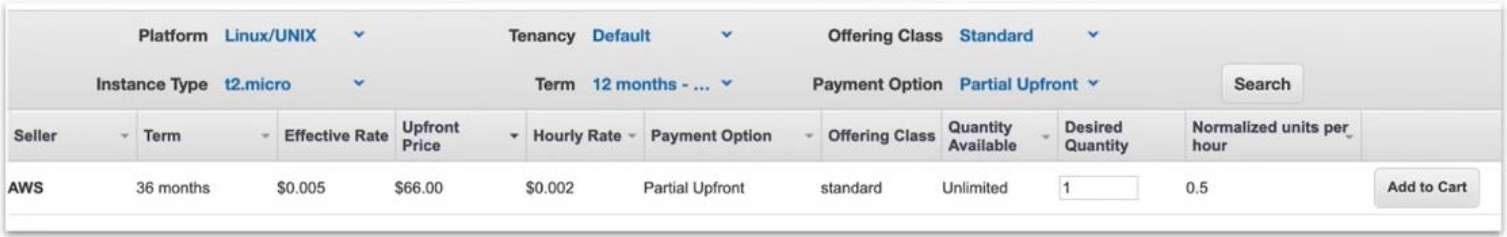
amazon web services

EC2 – Reserved Instances (RI)

Best Long-term

Designed for applications that have a **steady-state**, **predictable usage**, or require **reserved capacity**.

Reduced Pricing is based on **Term** x **Class Offering** x **Payment Option**



Standard Up to **75%** reduced pricing compared to on-demand.
Cannot change RI Attributes.

Convertible Up to **54%** reduced pricing compared to on-demand.
Allows you to change RI Attributes if greater or equal in value.

Scheduled You reserve instances for specific time periods eg. once a week for a few hours. Savings vary

Terms
You commit to a **1 Year** or **3 Year** contract.
The longer the term the greater savings.

Payment Options
All Upfront, **Partial Upfront**, and **No Upfront**
The greater upfront the great the savings

RIs can be **shared between multiple accounts** within an org

Unused RIs can be sold in the **Reserved Instance Marketplace**

amazon web services

AWS Solutions Architect Associate

EC2 Pricing Models

Spot Instances Pricing

amazon web services

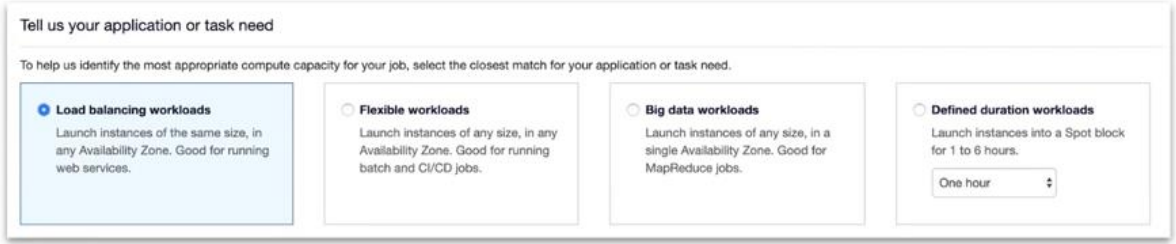
EC2 – Spot Instances

Biggest Savings

AWS has **unused compute capacity** that they want to maximize the utility of their idle servers.
It's like when a hotel offers discounts for to fill vacant suites or planes offer discount to fill vacant seats.

Spot Instances provide a discount of **90%** compared to On-Demand Pricing
Spot Instances can be terminated if the computing capacity is needed by on-demand customers.

Designed for applications that have flexible start and end times or applications that are only feasible at **very low** compute costs.



AWS Batch is an easy and convenient way to use Spot Pricing

Termination Conditions

Instances can be terminated by AWS **at anytime**
If your instance is **terminated by AWS**, **you don't get charged** for a partial hour of usage.
If **you terminate** an instance **you will still be charged** for any hour that it ran.

amazon web services

AWS Solutions Architect Associate

EC2 Pricing Models

Dedicated Host Instance Pricing

amazon web services

EC2 – Dedicated Host Instances

Most Expensive

Designed to meet regulatory requirements. When you have strict **server-bound licensing** that won't support multi-tenancy or cloud deployments.

Multi-Tenant vs Single Tenant

When multiple customers are running workloads on the same hardware. **Virtual Isolation** is what separate customers. (think apartment)

Multi-Tenant

When a single customer has dedicated hardware. **Physical Isolation** is what separates customers (think house)

Single-Tenant Single-Tenant Single-Tenant

Offered in both **On-demand** and **Reserved** (70% off on-demand pricing)

Enterprises and **Large Organizations** may have security concerns or obligations about against sharing the same hardware with other AWS Customers.

amazon web services

AWS Solutions Architect Associate

EC2 Pricing Models

EC2 Pricing Models Cheat Sheet

amazon web services

EC2 Pricing – CheatSheet

- EC2 has for 4 pricing models **On-Demand**, **Spot**, **Reserved Instances (RI)** and **Dedicated**
- **On-Demand** (least commitment)
  - low cost and flexible
  - only pay per hour
  - **Use case:** short-term, spiky, unpredictable workloads, first time apps
  - Ideal when your workloads cannot be interrupted
- **Reserved Instances** upto 75% off (Best long-term value)
  - **Use case:** steady state or predictable usage
  - Can resell unused reserved instances (Reserved Instance Marketplace)
  - Reduced Pricing is based on **Term** x **Class Offering** x **Payment Option**
  - **Payment Terms**: 1 year or 3 year
  - **Payment Options**: All Upfront, Partial Upfront, and No Upfront
  - **Class Offerings**
    - **Standard** Up to 75% reduced pricing compared to on-demand. Cannot change RI Attributes.
    - **Convertible** Up to 54% reduced pricing compared to on-demand. Allows you to change RI Attributes if greater or equal in value.
    - **Scheduled** You reserve instances for specific time periods eg. once a week for a few hours. Savings vary

EC2 Pricing – CheatSheet

- **Spot Pricing** upto 90% off (Biggest Savings)
  - request spare computing capacity
  - flexible start and end times
  - **Use case:** Can handle interruptions (server randomly stopping and starting)
  - **Use case:** For non-critical background jobs
  - Instances can be terminated by AWS **at anytime**
  - If your instance is **terminated by AWS**, **you don't get charged** for a partial hour of usage.
  - If **you terminate** an instance **you will still be charged** for any hour that it ran.
- **Dedicated Hosting** (Most Expensive)
  - Dedicated servers
  - Can be on-demand or reserved (upto 70% off)
  - **Use case:** When you need a guarantee of isolate hardware (enterprise requirements)

AWS Solutions Architect Associate

Amazon Machine Image

Amazon Machine Image Introduction

amazon web services

Amazon Machine Image (AMI)

A template to configure new instances

EC2 - AMI

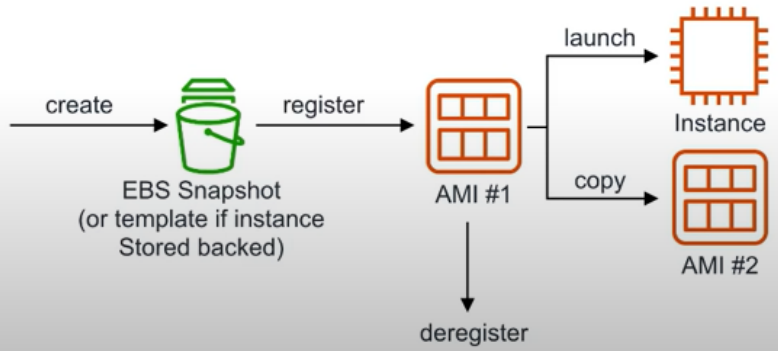
Amazon Machine Image **(AMI)** provides the information required to launch an instance.

You can **turn your EC2 instances into AMIs** so you can **create copies of your servers**

An AMI holds the following information:

- A template for the root volume for the instance (EBS Snapshot or Instance Store template) eg. an operating system, an application server, and applications
- Launch permissions that control which AWS accounts can use the AMI to launch instances.
- A block device mapping that specifies the volumes to attach to the instance when it's launched.

AMIs are **Region Specific!**



AWS Solutions Architect Associate

Amazon Machine Image

AMI Use Case

AMI – Use Cases

AMIs help you keep incremental changes to your OS, application code and system packages.

web-server-000
Ruby, Node, Postgres Client Installed

web-server-001
Redis for Sidekiq Installed

web-server-002
ImageMagick for Image Processing Installed

web-server-003
CloudWatch Agent Installed

Using **Systems Manager Automation** you can routinely patch your AMIs with security updates and bake those AMIS.

AMIs are used with **LaunchConfigurations**. When you want to roll out updates to multiple instances you make a copy of your LaunchConfiguration with new AMI

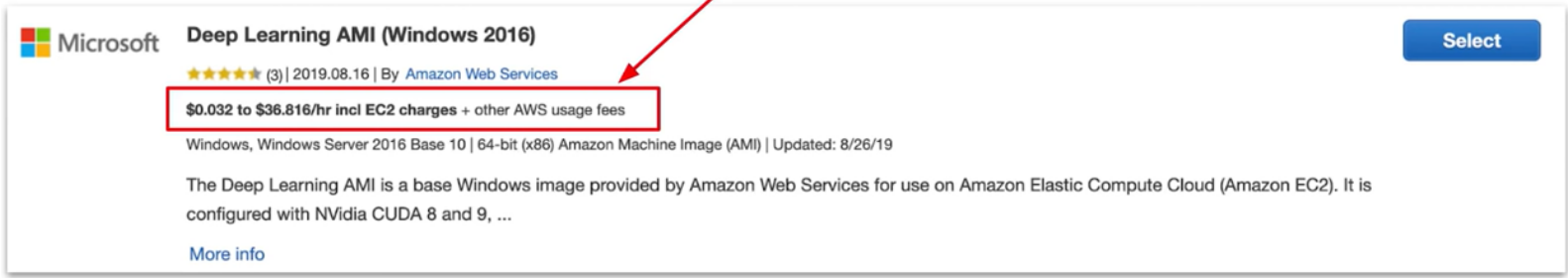
AWS Solutions Architect Associate

Amazon Machine Image

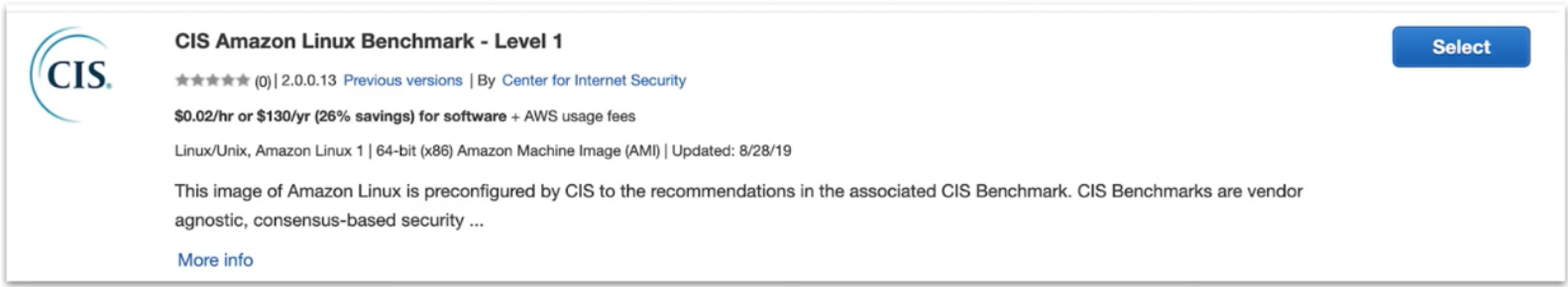
AMI Marketlpace

AWS Marketplace

The AWS Marketplace lets you **purchase subscriptions** to vendor maintained AMIs.



Security harden AMIs are very popular. eg. Center of Internet Security



amazon web services

AWS Solutions Architect Associate

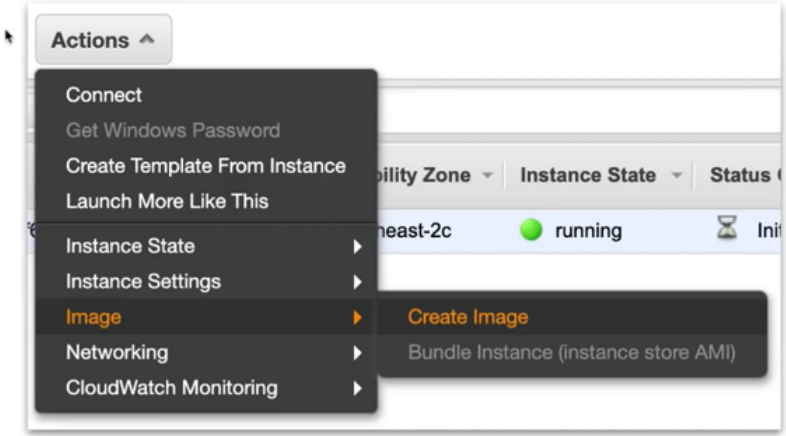
Amazon Machine Image

Creating an AMI

amazon web services

AMI – Creating an AMI

You can **create an AMI** from an existing EC2 instance that's either **running** or **stopped**.



AWS Solutions Architect Associate

Amazon Machine Image

Choosing an AMI

amazon web services

Choosing an AMI

AWS has hundreds of AMIs you can **search** and select from.

Community AMI are free AMIs maintained by the community
AWS Marketplace free or paid AMIs maintained by vendors

North Virginia

Amazon Linux 2 AMI (HVM), SSD Volume Type - ami-0b69ea66ff7391e80 (64-bit x86) / ami-09c61c4850b7465cb (64-bit Arm)

Amazon Linux
Free tier eligible

Amazon Linux 2 comes with five years support. It provides Linux kernel 4.14 tuned for optimal performance on Amazon EC2, systemd 219, GCC 7.3, Glibc 2.26, Binutils 2.29.1, and the latest software packages through extras.

Root device type: ebs Virtualization type: hvm ENA Enabled: Yes

Select

64-bit (x86)
64-bit (Arm)

AMIs have an **AMI ID.** AMIs are **region specific**. Will have different AMI ID per region.

Canada Central

Amazon Linux 2 AMI (HVM), SSD Volume Type - ami-085edf38cedbea498

Amazon Linux
Free tier eligible

Amazon Linux 2 comes with five years support. It provides Linux kernel 4.14 tuned for optimal performance on Amazon EC2, systemd 219, GCC 7.3, Glibc 2.26, Binutils 2.29.1, and the latest software packages through extras.

Root device type: ebs Virtualization type: hvm ENA Enabled: Yes

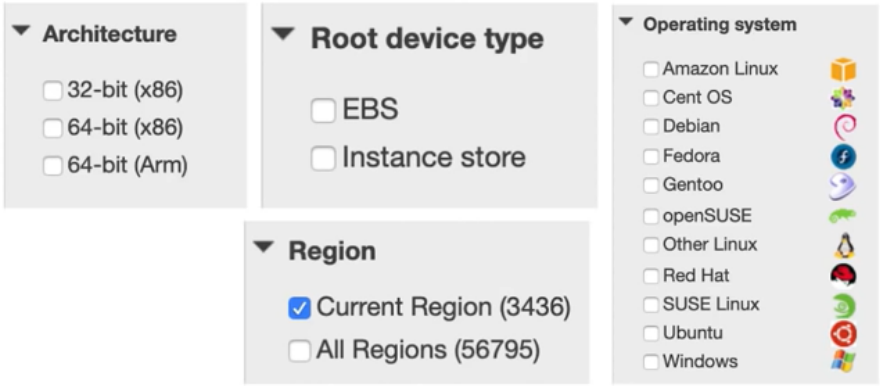
Select

64-bit (x86)

Choosing an AMI

Amazon Machine Images can be selected based on:

- Region
- Operating System
- Architecture (32-bit or 64-bit)
- Launch Permissions
- Root Device Volume
  - Instance Store (Ephemeral Storage)
  - EBS Backed Volumes



AMIs are categorized as either backed by Amazon EBS, or backed by Instance Store

AWS Solutions Architect Associate

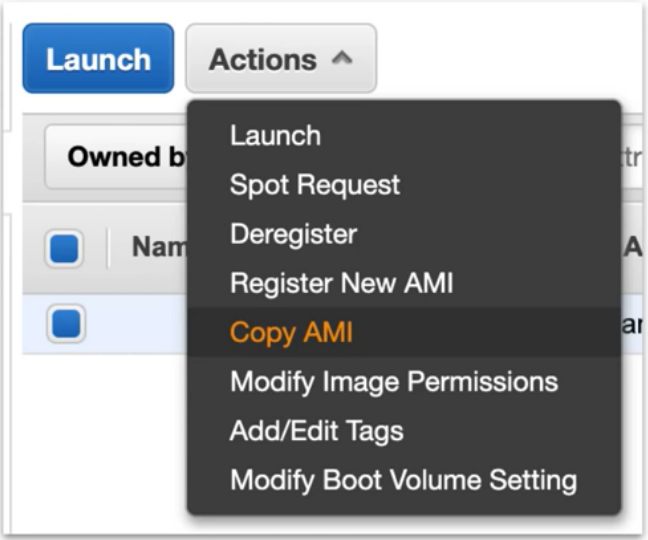
Amazon Machine Image

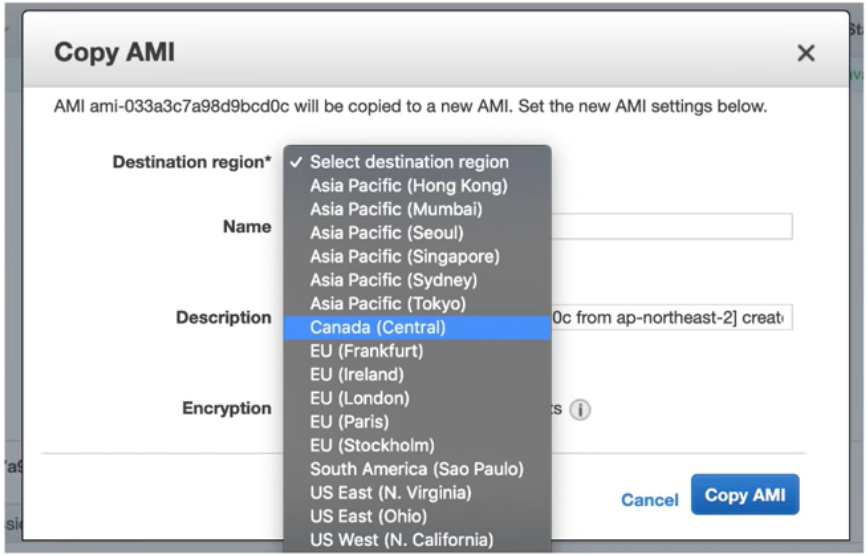
Copying an AMI

amazon web services

EC2 – Copying an AMI

AMIs are region specific. If you want to use an AMI from another region. You need to **Copy the AMI** and then select the destination region.





amazon web services

AWS Solutions Architect Associate

Amazon Machine Image

AMI Cheat Sheet

amazon web services

AMI CheatSheet

- **Amazon Machine Image (AMI)** provides the information required to launch an instance.
- AMIs are region specific, if you need to use an AMI in another region you can copy an AMI into the destination region via **Copy AMI**
- You can **create an AMI** from an existing EC2 instance that's either **running** or **stopped**.
- **Community AMI** are free AMIs maintained by the community
- **AWS Marketplace** free or paid subscription AMIs maintained by vendors
- AMIs have an **AMI ID.** The same AMI eg. (Amazon Linux 2) will vary in both AMI ID and options eg. Architecture options in different regions
- **An AMI holds the following information:**
    - A template for the root volume for the instance (EBS Snapshot or Instance Store template) eg. an operating system, an application server, and applications
    - Launch permissions that control which AWS accounts can use the AMI to launch instances.
    - A block device mapping that specifies the volumes to attach to the instance when it's launched.

Topics

- Basic tenets of AWS
- Patterns and (anti-patterns) for creating scalable architectures in AWS
- Bootstrapping EC2 Instances
- Building with CloudFormation
- Components of Auto Scaling

AWS Solutions Architect Associate

Auto Scaling Groups

Auto Scaling Groups Introduction

amazon web services

EC2 Auto Scaling Groups

Set scaling rules which will automatically launch additional EC2 instance or shutdown instances to meet current demand

Introduction to Auto Scaling Groups

Auto Scaling Groups (**ASG**) contains a collection of EC2 instances that are treated as a group for the purposes of automatic scaling and management.

Automatic scaling can occur via:

1. **Capacity Settings**
2. **Health Check Replacements**
3. **Scaling Policies.**

AWS Solutions Architect Associate

Auto Scaling Groups

Capacity Settings

amazon web services

ASG - Capacity Settings

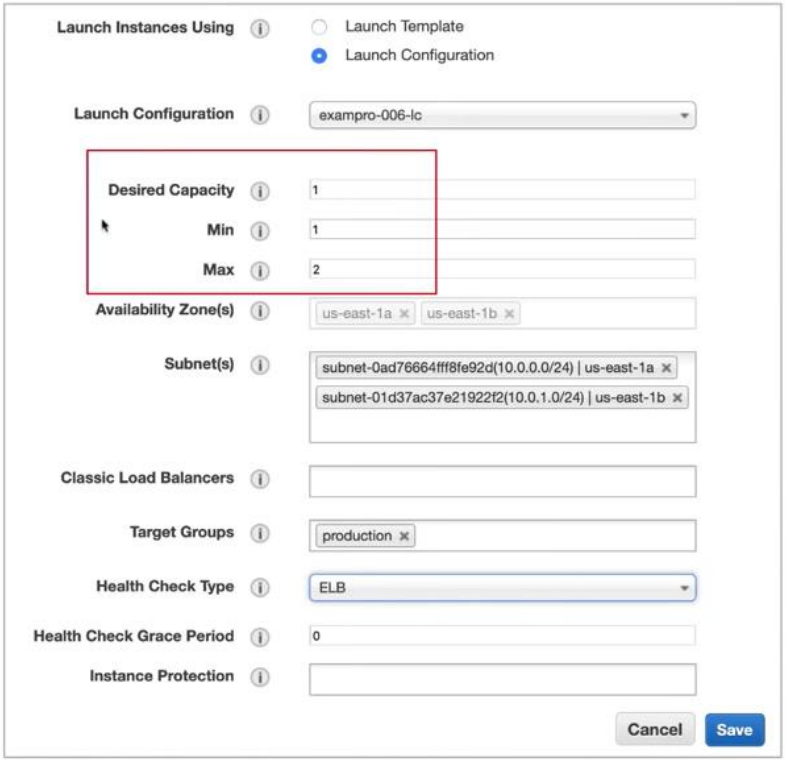
The size of an Auto Scaling Group is based on **Min**, **Max** and **Desired Capacity**.

Min is how many EC2 instances should at least be running.

Max is number EC2 instances allowed to be running.

Desired Capacity is how many EC2 instances you want to ideally run.

ASG will always launch instances to meet minimum capacity.



amazon web services

AWS Solutions Architect Associate

Auto Scaling Groups

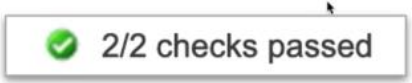
Health Check Replacements

amazon web services

ASG – Health Check Replacements

EC2 Health Check Type

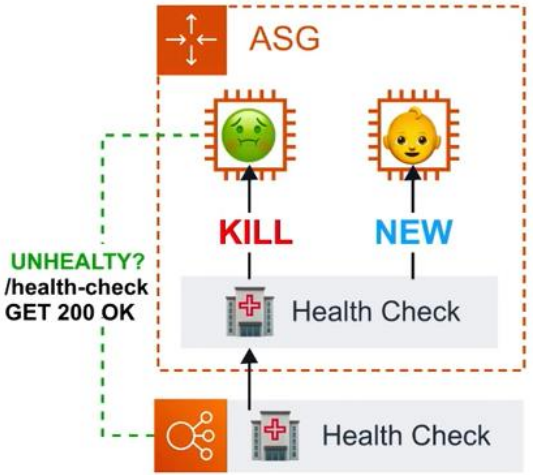
ASG will perform a health check on EC2 instances to determine if there is a software or hardware issue. This is based on the **EC2 Status Checks**. If an instance is considered unhealthy. ASG will terminate and launch a new instance.



ASG – Health Check Replacements

ELB Health Check Type

ASG will perform a health check based on the ELB health check. ELB can perform health checks by pinging an HTTP(S) endpoint with an expected response. If ELB determines a instance is unhealthy it forwards this information to ASG which will terminate the unhealthy instance.



AWS Solutions Architect Associate

Auto Scaling Groups

Scaling Policies

amazon web services

ASG – Scaling Policies

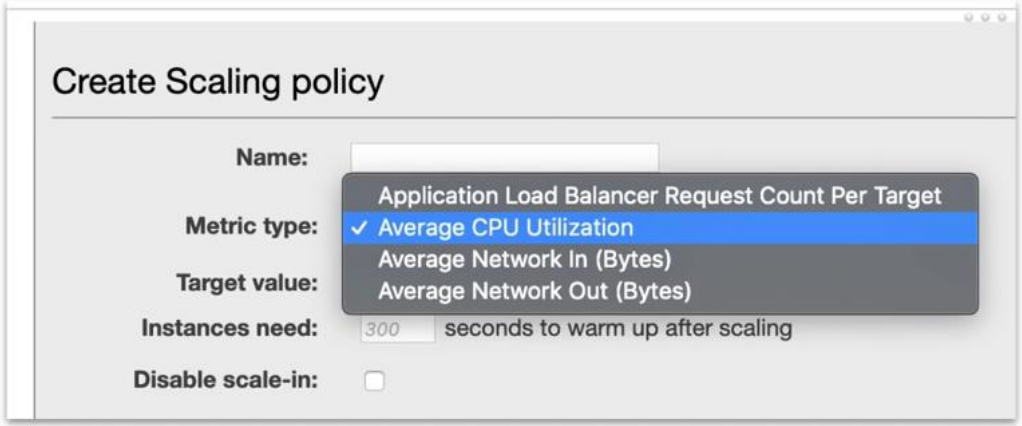
Scaling Out: Adding More Instances

Scaling In: Removing Instances

Target Tracking Scaling Policy

Maintains a specific metric at a target value.

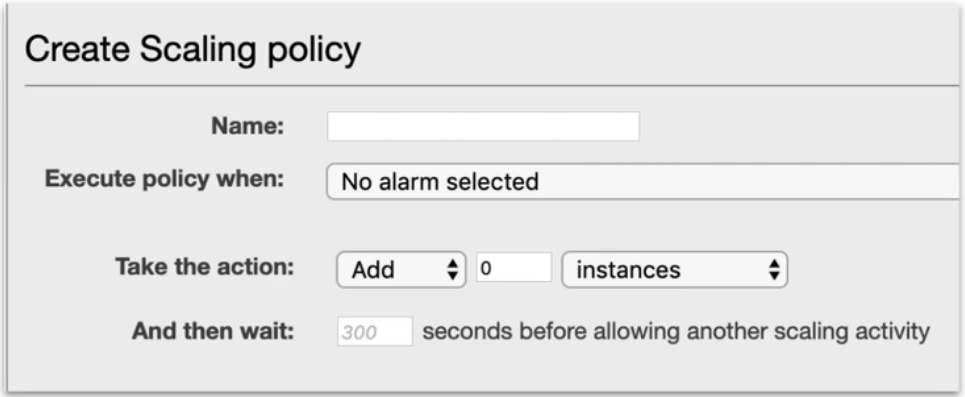
eg. If **Average CPU Utilization** exceeds 75% then add another server.



ASG – Scaling Policies

Simple Scaling Policy

Scales when an **alarm is breached**.

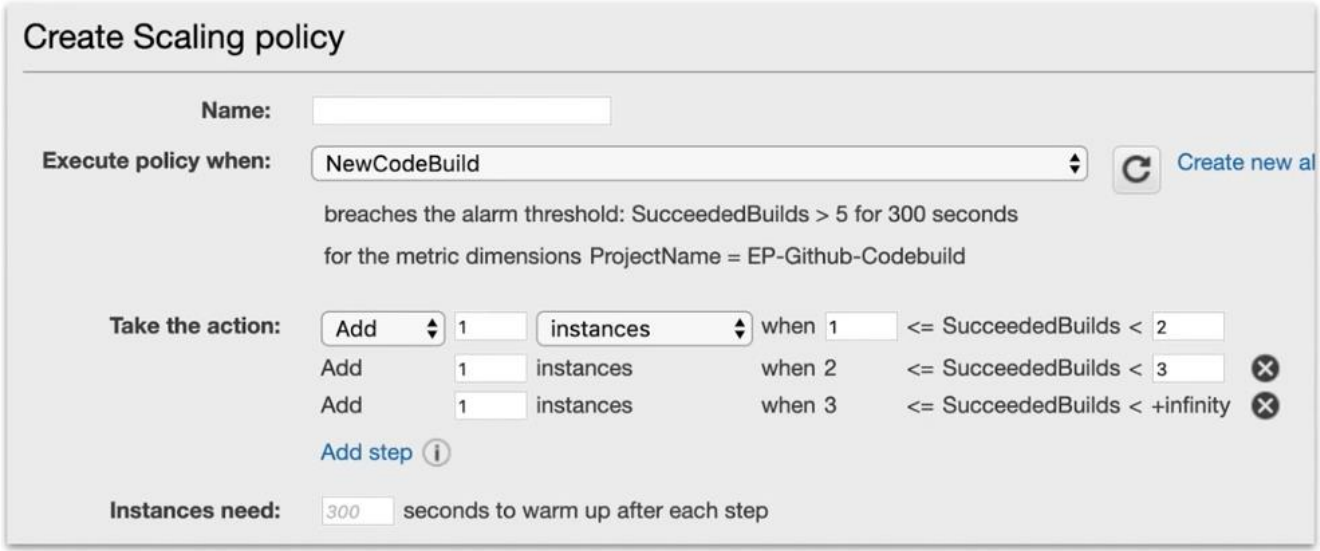


Not recommended, legacy scaling policy. Use scaling policies with steps now.

ASG – Scaling Policies

Scaling policies with steps

Scales when an **alarm is breached**, can **escalates based on alarm** value changing.



AWS Solutions Architect Associate

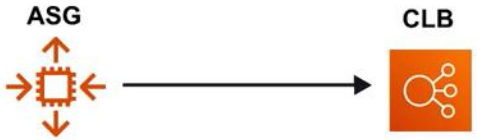
Auto Scaling Groups

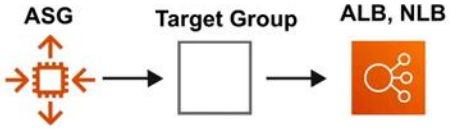
ELB Integration

ASG – ELB Integration

ASG can be associated with Elastic Load Balancers (ELB). When ASG is associated with ELB richer health checks can be set.

Classic Load Balancers are associated **directly** to the ASG





Application and Network Load Balancers are associated **indirectly** via their Target Groups.

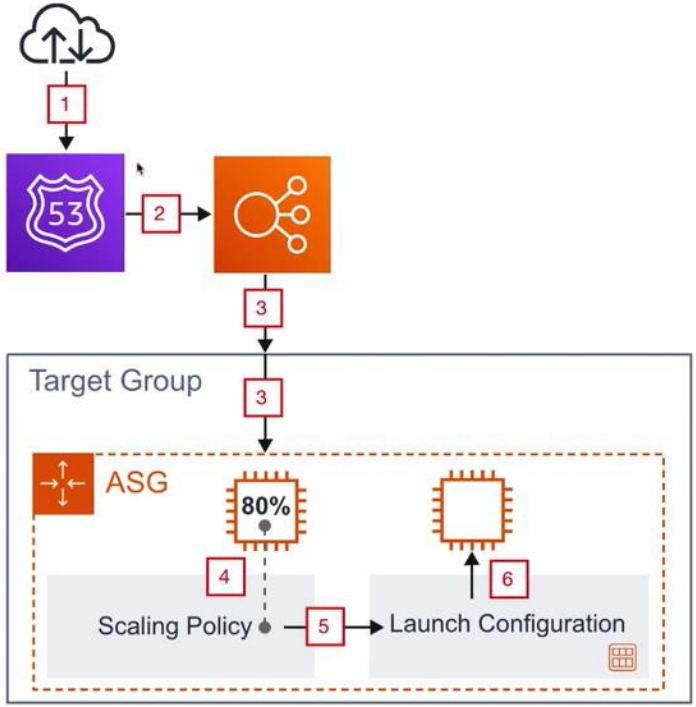
AWS Solutions Architect Associate

Auto Scaling Groups

ASG Use Case

ASG – Use Case

1. Burst of traffic from the internet hits our domain.
2. Route53 points that traffic to our load balancer.
3. Our load balancer passes the traffic to its target group.
4. The target group is associated with our ASG and sends the traffic to instances registered with our ASG
5. The ASG Scaling Policy will check if our instances are near capacity.
6. The Scaling Policy determines we need another instance, and it Launches an new EC2 instance with the associated Launch Configuration to our ASG



AWS Solutions Architect Associate

Auto Scaling Groups

Launch Configuration

amazon web services

Launch Configuration

A launch configuration is an instance configuration template that an Auto Scaling group uses to launch EC2 instances.

A Launch Configuration is the same process as Launching an EC2 instance except you are saving that configuration to Launch an Instance for later. Hence “Launch Configuration.

Launch Configurations **cannot be edited**, When you need to update your Launch Configuration you create a new one or clone the existing configuration and then manually associate that new Launch Configuration

Launch Templates are Launch Configurations with Versioning, Everyone appears to still use Launch Configurations

AWS Solutions Architect Associate

Auto Scaling Groups

Auto Scaling Groups Cheat Sheet

amazon web services

EC2 Auto Scaling Groups CheatSheet

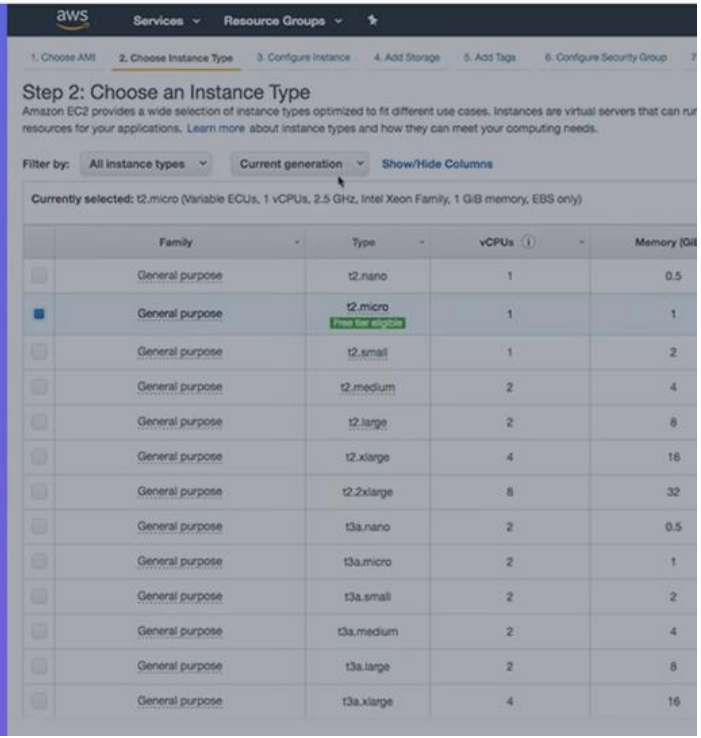
- An ASG is a collection of EC2 instances grouped for scaling and management
- Scaling Out is when add servers
- Scaling In is when you remove servers
- Scaling Up is when you increase the size of an instance (eg. updating Launch Configuration with larger size)
- Size of an ASG is based on a **Min, Max** and **Desired Capacity**
- **Target Scaling policy** scales based on when a target value for a metric is breached eg. Average CPU Utilization exceed 75%
- **Simple Scaling** policy triggers a scaling when an alarm is breached
- **Scaling Policy with Steps** is the new version of Simple Scaling policy and allows you to create steps based on eculation alarm values.
- Desired Capacity is how many EC2 instances you want to ideally run
- An ASG will always launch instances to meet minimum capacity
- Health checks determine the current state of an instance in the ASG
- Health checks can be run against either an ELB or the EC2 instances
- When an Autoscaling launches a new instance it uses a Launch Configuration which holds the configuration values for that new instance eg. AMI, InstanceType, Role
- Launch Configurations cannot be edited and must be cloned or a new one created
- Launch Configurations must be manually updated in by editing the Auto Scaling settings.

AWS Solutions Architect Associate

EC2 Follow Along

Launching an Instance

Follow Along



amazon web services

Lab 3

Getting Started with Auto Scaling

amazon web services