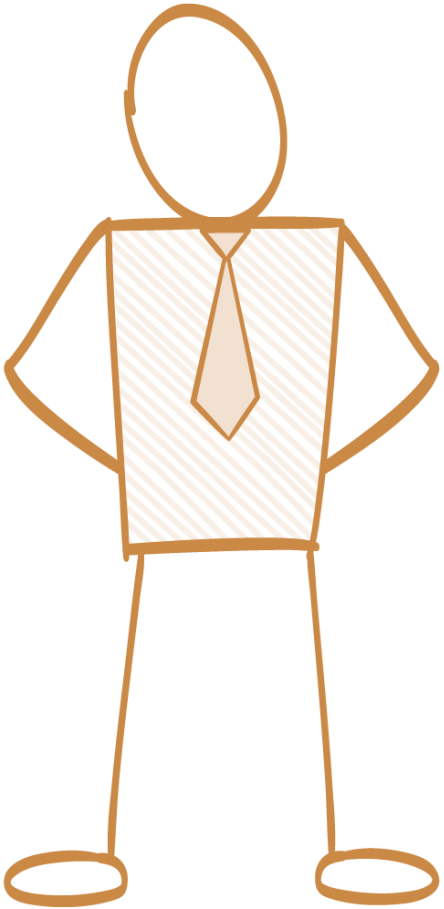
Architecting on AWS
Student Guide
Version 3.1

100-ARC-31-EN-SG



Module 12: Designing for Cost

Topics

- Cost model
- Services and feature costs
- Billing options
- Best practices

Topics

- Cost model
- Services and feature costs
- Billing options
- Best practices

Cost model

- Amazon wants customers to pay for exactly what they use
- Do not pay for unutilized feature or services
- This model translates into a very granular cost structure
- Every Application has different component bounding (CPU, Memory, Disk I/O), pay for what you use
- Customers have control of how they utilize our products and service, which leads to control over cost expenditures

Topics

- Cost model
- Services and feature costs
- Billing options
- Best practices

ELB, EIP, and CloudWatch Costs

- EIP
  - Free when associated with an EC2 instance
  - $0.005 per hour unassociated
  - $0.10 per 100 remaps
- CloudWatch
  - Detailed monitoring $3.50 per instance per month
  - Custom metrics $0.50 per metric per month
- ELB
  - $0.025 per ELB-hour
  - $0.008 per GB of bandwidth

Note: Pricing defers from region to region and subject to change. See our web site for the most up-to-date pricing information

ESB Service and Feature Costs

- Standard EBS Volume
  - $0.05 per GB/month
  - $0.05 per million I/O requests
- Provisioned IOPS EBS Volumes
  - $0.125 per GB-month of provisioned storage
  - $0.10 per Provisioned IOPS-month
- EBS Snapshots to Amazon S3
  - $0.095 per GB-month of data stored

S3, S3 RRS, Glacier Costs

- Pay for capacity used. For the first 1TB per month:
  - S3 Standard Storage - $0.03/GB
  - S3 Reduced Redundancy Storage - $0.024/GB
  - Glacier - $0.01/GB
- S3 PUT, COPY, POST, LIST - $0.05 per 1,000 requests
- Glacier Archive/Restore - $0.05 per 1,000 requests
- DELETE – Free
- GET and all other requests - $0.004 per 10,000 requests
- Data transfer OUT from S3 to Internet - $0.12 per GB up to 10TB per month

RDS Service and Feature Costs

- Multiple instance types to choose from
- Provisioned IOPS (up to 30,000 per DB) optional
- Data Transfer Out of a Region - $0.02/GB
- Reserved billing model available

DynamoDB Service Costs

- Provision IOPS capacity
  - $0.0065 per hour for every 10 units of write capacity
  - $0.0065 per hour for every 50 units of read capacity
- Indexed data storage
  - First 100MB stored per month is free
  - $0.25 per GB-month thereafter
- Data transfer OUT - $0.12/GB up to 10TB/month
- Reserved billing model available

R53 and CloudFront Service Costs

- Route 53
  - $0.50 per hosted zone/month for the first 25 hosted zones
  - $0.50 per million queries for the first 1 billion queries/month
  - $0.75 per million for latency based queries for the first 1 billion queries/month
  - $0.50 per health check per month inside AWS, $0.75 outside (S3 Endpoints are free) – Basic Health Check
- CloudFront
  - Bandwidth out $0.12 per GB (US and Europe) for the first 10TB/month
  - $0.0075 per 10,000 requests (US) for all HTTP methods
  - Pricing based on edge location. For details see: http://aws/amazon.com/cloudfront/pricing

SQS, SNS, SES Service and Feature Costs

- SQS
  - First 1 million SQS requests/month are free. $0.05 per 1 million SQS requests/month thereafter
- SNS
  - $0.06 per million HTTP/s notifications
  - $2.00 per 100,000 Emails notifications
  - $0.75 per 100 SMS notifications
  - No charge for SQS notifications
- SES
  - $0.10 per 1,000 Emails out
  - Data transfer inside a region is free
  - $0.12 per GB of attachments sent

Topics

- Cost model
- Services and feature costs
- Billing options
- Best practices

Free Services and Features

- Free Tier Utilization
- VPC
- Auto Scaling
- Cloud Watch standard metrics
- CloudFormation
- IAM
- OpsWorks
- Elastic Beanstalk

EC2 Billing Options

- Prices vary by instance type – optimized to fit different use cases
- On Demand prices should be considered retail rate
- Reserved Instance billing model available
- Unique Spot Marker available

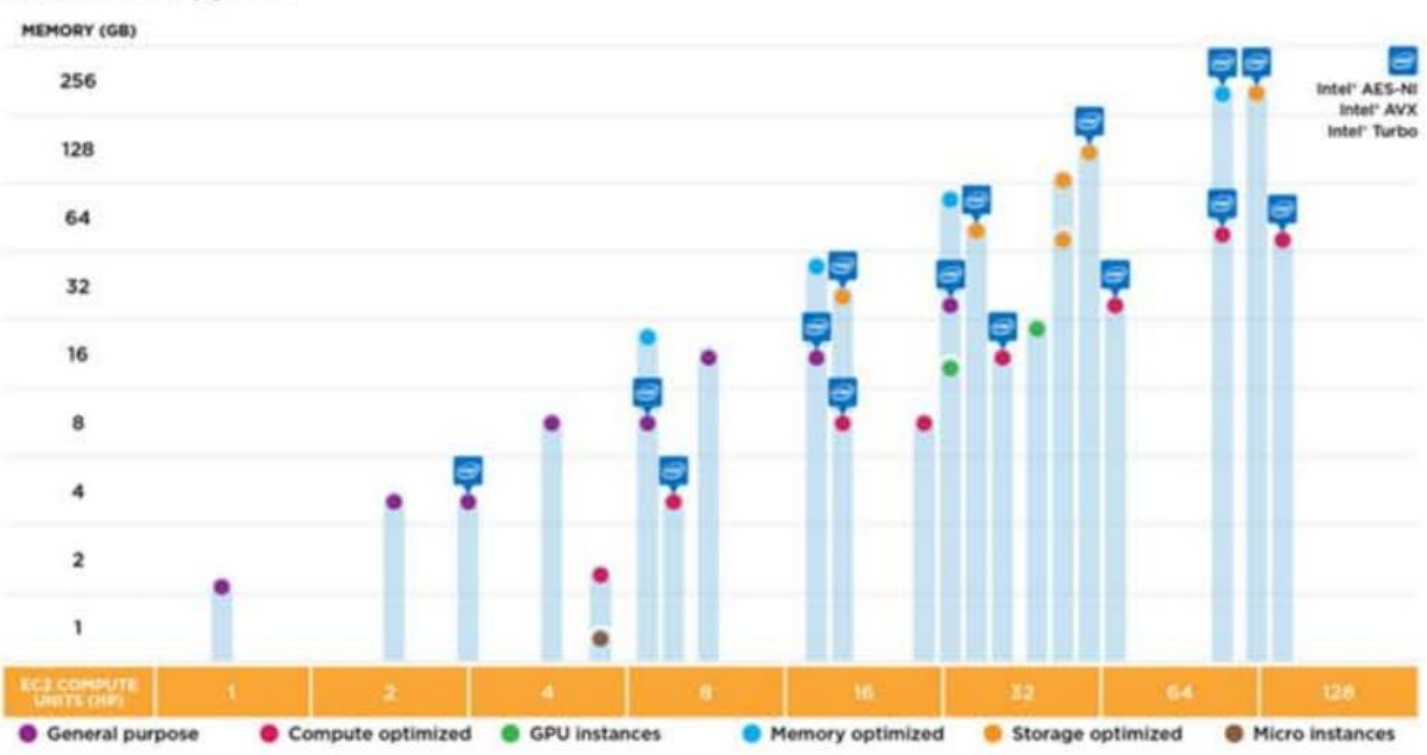
Your choice of Amazon EC2 instances matters

- A larger compute instance will sometimes save you not only time but money too. Paying more per hour for a shorter amount of time can be less expensive
- Instances come in multiple sizes, allowing you to optimally scale resources to the requirements of your workload. As you choose an instance type, consider the following:
  - Core count
  - Memory size
  - Storage size & type
  - Network performance

Additional Features that impact your workload

- Intel AES-NI1 – Intel processors that support these new encryption instructions allow you to enable encryption for enhanced data security without paying a performance penalty
- Intel AVX – Get dramatically better performance for parallel HPC workloads such as life science engineering, data mining, financial analysis, or other technical computing applications. AVX also enhances image, video, and audio processing.
- Intel Turbo Boost Technology2 – Get a turbo boost of compute speed, accelerating performance for peak loads. This Instance is appropriate for traditional non-parallel workloads.

Instances with All Three Intel® Xeon® Processor Technologies



amazon web services

Amazon EC2 Intel Processor Specifications

Instance Family	Instance Type	Processor Arch	vCPU	ECU	Physical Processor	Intel® AES-NI	Intel® AVX†	Intel® Turbo
General purpose	m3.medium	64-bit	1	3	Intel Xeon E5-2670	Yes	Yes	Yes
General purpose	m3.large	64-bit	2	6.5	Intel Xeon E5-2670	Yes	Yes	Yes
General purpose	m3.xlarge	64-bit	4	13	Intel Xeon E5-2670	Yes	Yes	Yes
General purpose	m3.2xlarge	64-bit	8	26	Intel Xeon E5-2670	Yes	Yes	Yes
Compute optimized	c3.large	64-bit	2	7	Intel Xeon E5-2680 v2	Yes	Yes	Yes
Compute optimized	c3.xlarge	64-bit	4	14	Intel Xeon E5-2680 v2	Yes	Yes	Yes

EC2 Billing Options

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

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

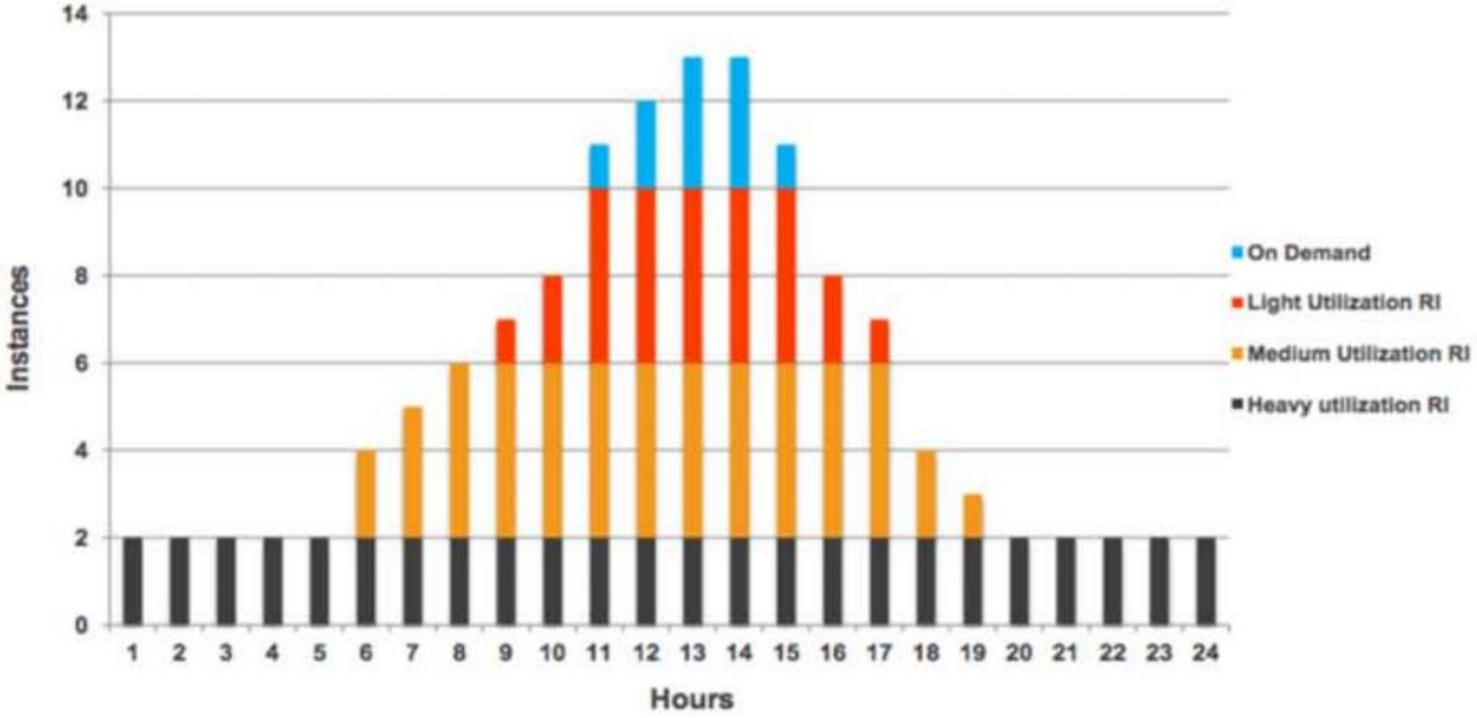
EC2 Billing Options

Reserved Instance Cost Savings Over On-Demand (m1.large – Linux – One Year RI)

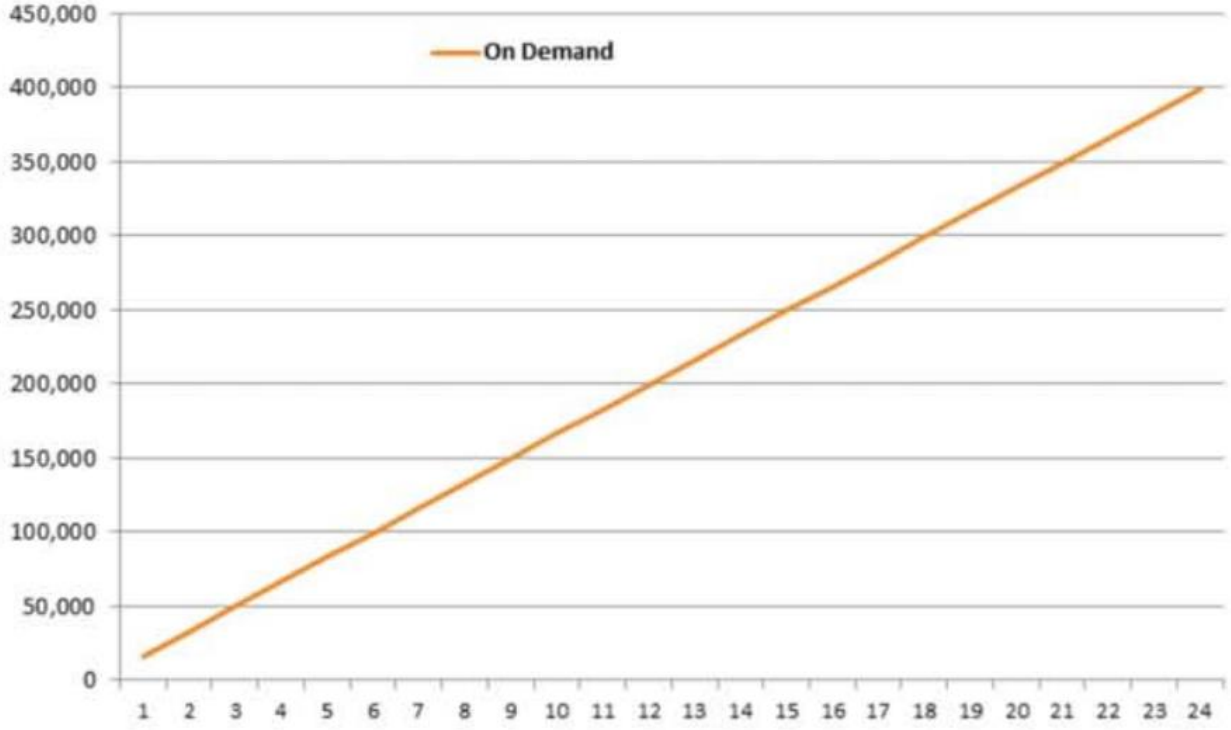
Annual Utilization	On Demand	Light Utilization RI	Medium Utilization RI	Heavy Utilization RI
10%	$234.00	-77.95%	-210.43%	-479.49%
20%	$468.00	-18.97%	-73.68%	-189.74%
30%	$702.00	0.68%	-28.09%	-93.16%
40%	$936.00	10.51%	-5.30%	-44.87%
50%	$1,170.00	16.41%	8.38%	-15.90%
60%	$1,404.00	20.34%	17.49%	3.42%
70%	$1,638.00	23.15%	24.00%	17.22%
80%	$1,872.00	25.26%	28.89%	27.56%
90%	$2,106.00	26.89%	32.69%	35.61%
100%	$2,340.00	28.21%	35.73%	42.05%

Optimal Savings Sub-Optimal Savings Least Savings

Reserved Instances

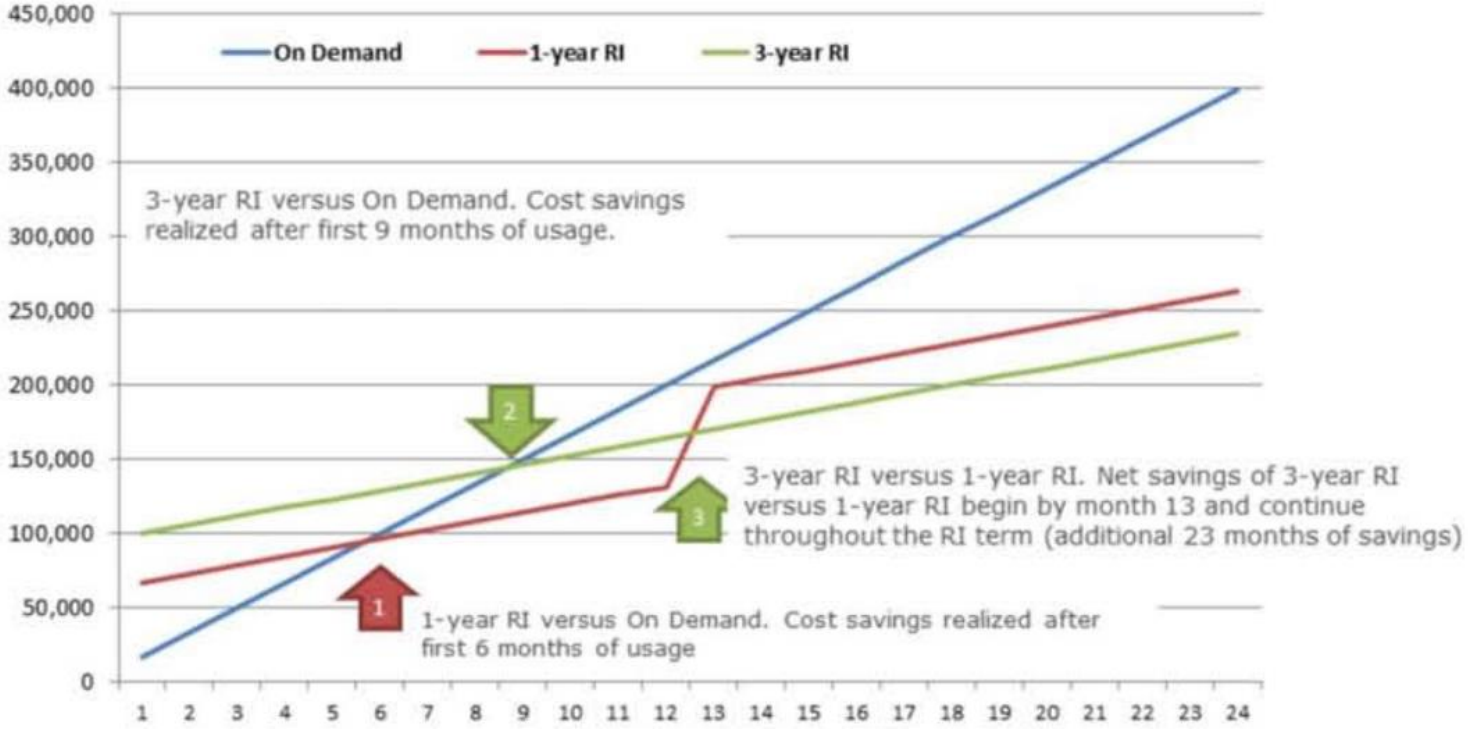


Reserved vs. On-Demand



Reserved vs. On-Demand

Reserved vs. On-Demand



Spot Market

- Spot instances often offer a significant savings over on-demand
- After an architecture is built for elasticity, leveraging spot instances can be a simple change

Spot Market

Spot Market

Use Case	Types of Applications
Batch Processing	Generic background processing (scale out computing)
Hadoop	Hadoop/MapReduce processing type jobs (Search, Big Data, and so on)
Scientific Computing	Scientific trials/simulations/analysis in chemistry, physics, and biology
Video and Image Processing/Rendering	Transform videos into specific formats
Testing	Provide testing of software, web sites, etc.
Web/Data Crawling	Analyzing data and processing it
Financial	Hedge fund analytics, energy trading, etc.
HPC	Utilize HPC servers to do embarrassingly parallel jobs
Cheap Compute	Backend servers for Facebook games

Spot Market

- Best practices for using spot instances:
  - Save your work frequently
  - Add checkpoints
  - Split up your work
  - Test your application
  - Use spot and on-demand in hybrid fashion → Master node in cluster is on-demand instance, worker nodes are Spot Instances

Topics

- Cost model
- Services and feature costs
- Billing options
- Best practices

Minimize Always On instances

- Elasticity is one of the fundamental properties of the cloud that drives many of its economic benefits
- Optimize your usage based on real-time demand
  - Reduce the number of web servers during off-peak periods
  - Shut down processing nodes unused at night
  - Purchase Reserved Instances for the Always On fleet

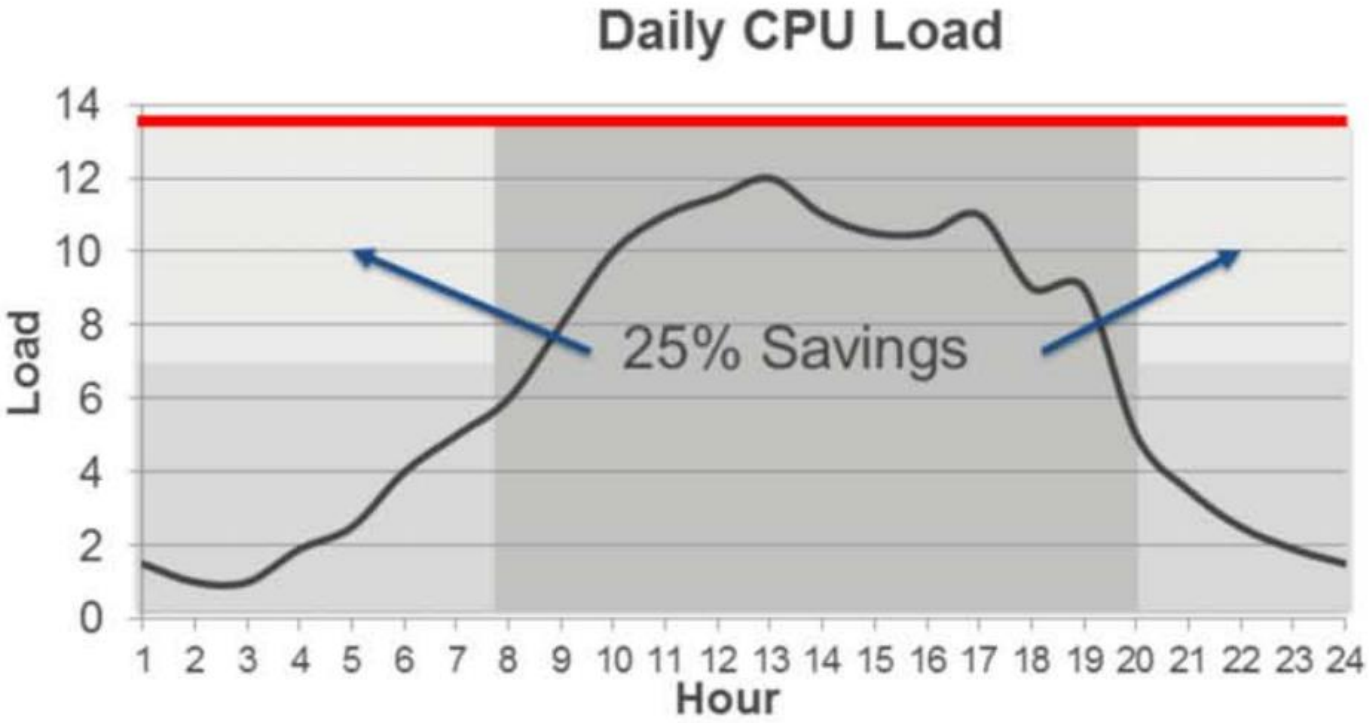
Scale-in automatically

- Scaling out is key to serving customer needs but scaling in is where the savings occur
  - Auto-scaling works well for stateless components
  - Scripted scaling makes sense for stateful components
- Examples
  - Auto-scaling based on Network I/O for web servers
  - De-scaling RDS instances on the weekend
  - Shutting down all workers when batch processing queues are empty

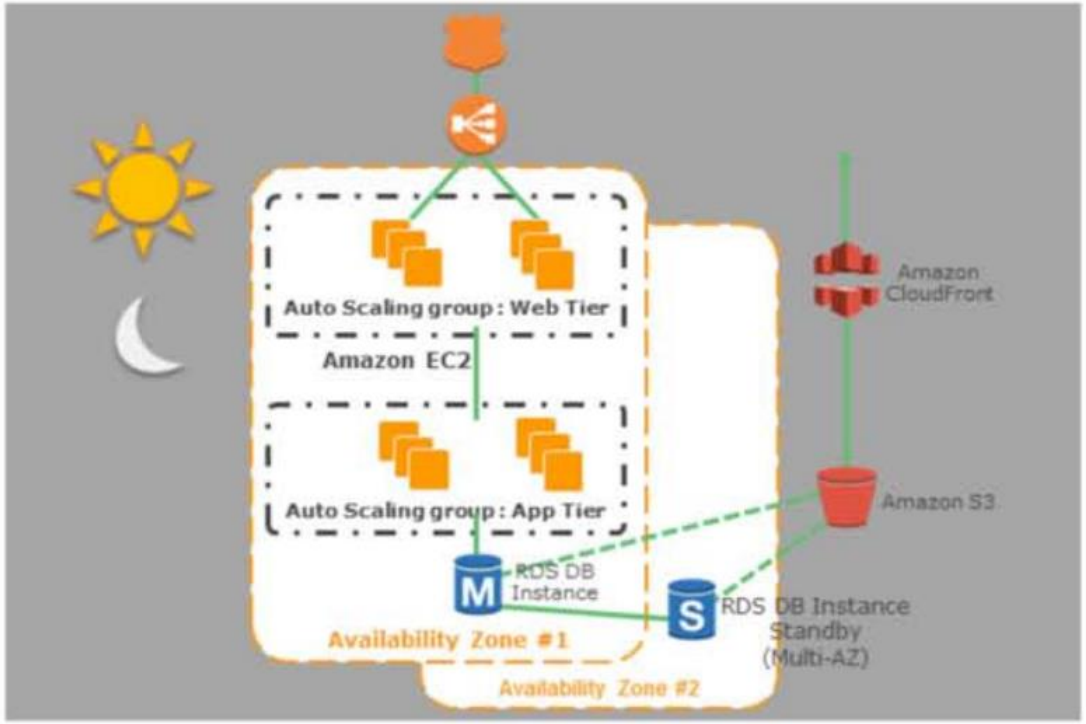
Scripted scaling

- Some scenarios do not lend themselves to automated de-scaling due to complex application logic
  - Shutting down worker nodes only when they are not currently working
  - Shutting down RDS read replicas during weekends
- Every AWS service has an API and command line tools for managing it
  - Scripting of scaling activities can be a significant cost savings for little effort

Optimize by time of day



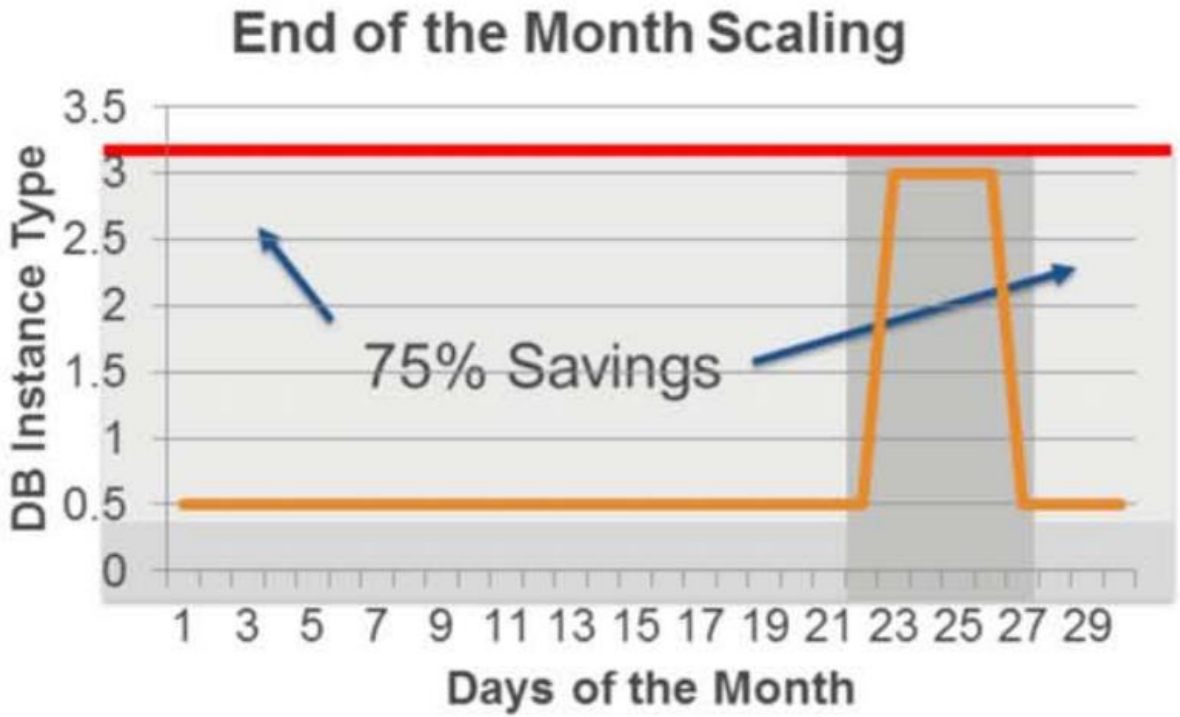
Optimize by time of day



End of month processing

- Expand the cluster at the end of the month
  - Expand/Shrink feature in Amazon Elastic MapReduce
- Vertically Scale up at the end of the month
  - Modify-DB-Instance (in Amazon RDS) (or a New RDS DB Instance)
  - CloudFormation Script (in Amazon EC2)

Optimize during the month



Leverage scalable, on-demand services

- EC2 can run almost anything but there are many cases where it is not cost effective
- AWS offers many scalable and cost-effective options for common application needs:
  - ELB instead of a software load balancer on EC2
  - SQS instead of a queue on EC2

Best Practices

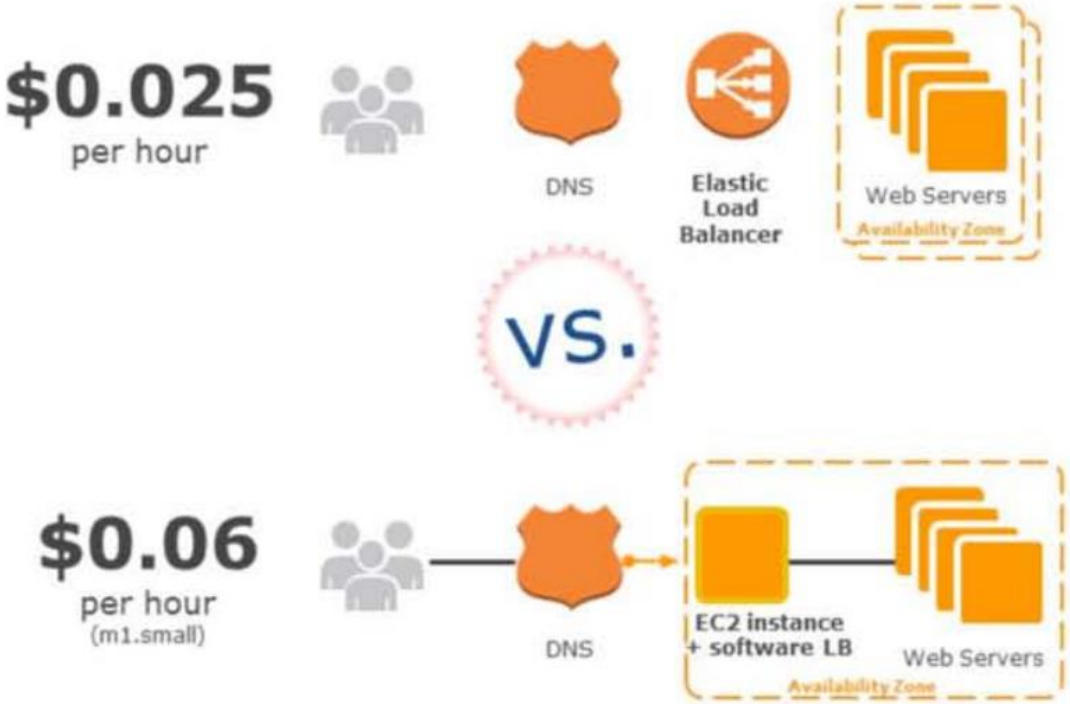
Software LB on EC2

- Pros
  - Application-tier load balancer
- Cons
  - SPOF
  - Elasticity has to be implemented manually
  - Not as cost-effective

ELB

- Pros
  - Pay as you go
  - Scalability
  - Availability
  - High performance

Best Practices (continue)



Best Practices

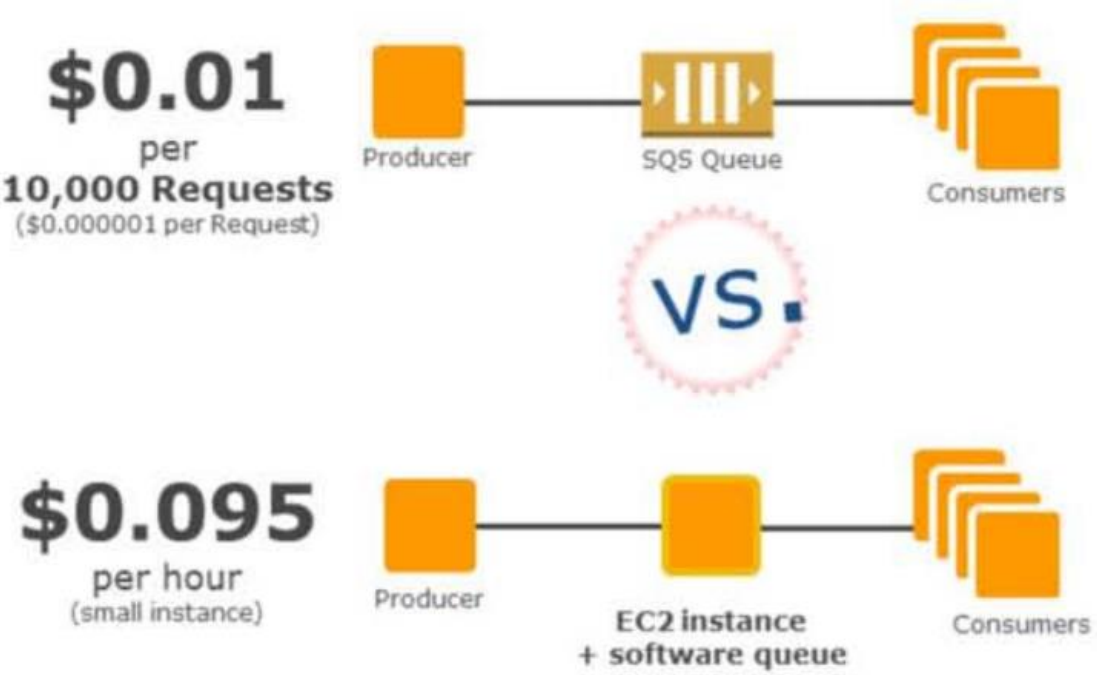
Software on EC2

- Pros
  - Custom features
- Cons
  - Requires an instance
  - SPOF
  - Limited to one AZ
  - DIY administration

SNS, SQS, SES

- Pros
  - Elastic and Fault-tolerant
  - Auto scaling
  - Monitoring included
  - IPV6
- Cons
  - Internal load balancing only in VPC

Best Practices



Clean up after yourself

- When it is easy to create resources, it can be easy to forget about them
  - Use tagging to identify the purpose of resources
  - Use CloudWatch to identify underutilized resources
  - Keep track of objects in S3 and clean up unused content
  - Release unused Elastic IPs
- Examples
  - Daily report on utilization of resources
  - Clean up script to delete old S3 objects
- Make use of Trusted Advisor

Economics Center

http://aws.amazon.com/economics

amazon web services