

CCA Unit 1 – Introduction to Cloud Computing

CCA 1.02: Leveraging Cloud Computing

CCA 1.01: What is Cloud Computing?

▶ **CCA 1.02:** Leveraging Cloud Computing

CCA 1.03: Cloud Economics and Total Cost of Ownership

Welcome to Module CCA 1.02 –Leveraging Cloud Computing

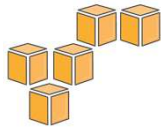
What's In This Module?

📦 Putting the Pieces Together

📦 Auto Scaling

This module covers...

- Putting the Pieces Together
- Auto Scaling

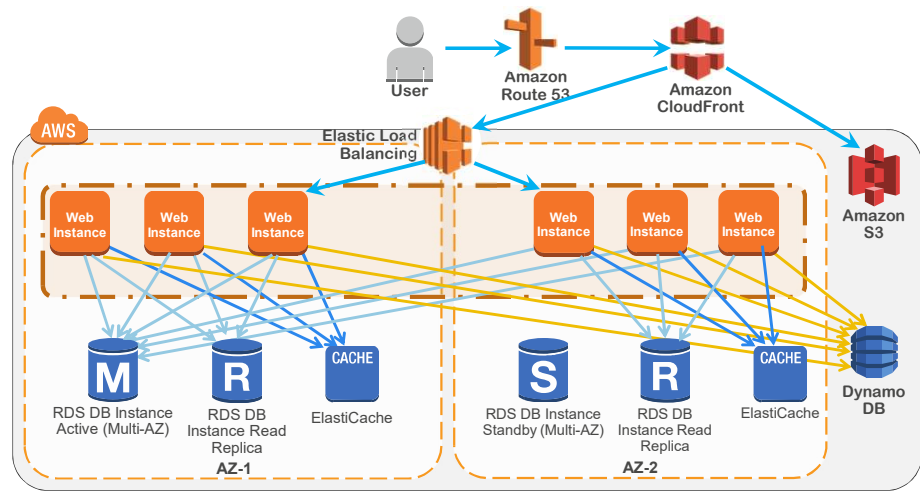
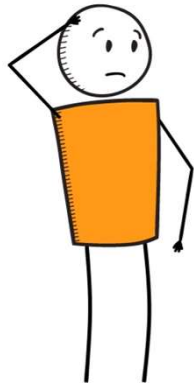


Part 1

Putting the Pieces Together

Part 1: Putting the Pieces Together

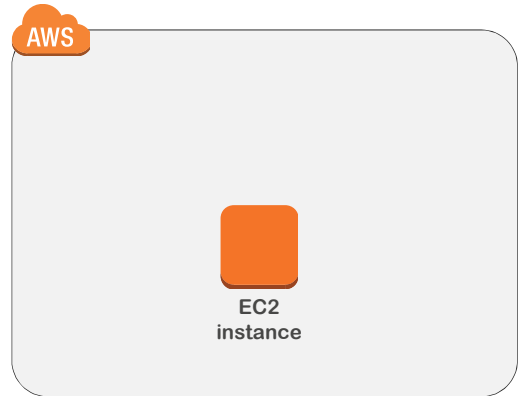
Cloud Computing Infrastructure



AWS Services can be used in a wide variety of ways to help organizations achieve diverse computing needs in a secure, highly available, and fault tolerant way. Each AWS solution is highly customizable to suite the unique needs of each customer. These architectures are highly customizable so that you pay only for what you need. So, how do all these pieces fit together?

Amazon Elastic Compute Cloud (EC2)

- Virtual servers (compute instances)
- Launch and manage from your AWS console.
- Windows or Linux
- Full administrative control



Amazon Elastic Compute Cloud (EC2) is the essential AWS compute service. It enables you to launch virtual servers or compute instances from your AWS management console. You can launch as many instances as you would like from a template which can be based on Windows or Linux. The instances spin up in just minutes giving you full administrative control just like any other server.

For a quick, 4-minute introductory video, visit: <https://aws.amazon.com/ec2/>

Amazon Machine Image (AMI)

- 📦 Templates for launching EC2 instances
- 📦 Defines a software configuration.
- 📦 Launch different types of instances
- 📦 Launch multiple instances from an AMI



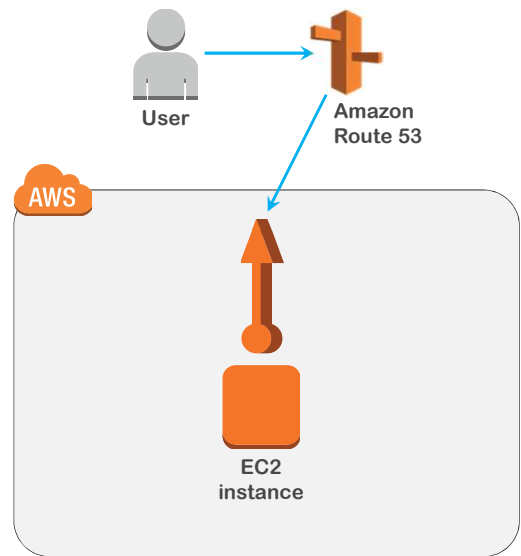
Amazon EC2 provides templates known as Amazon Machine Images (AMIs) that define a software configuration such as an operating system, application server, and applications. You use these templates to launch an instance which is a copy of the AMI running as a virtual server.

You can launch different types of instances from a single AMI. An instance type essentially determines the hardware capabilities of the virtual host computer for your instance. Each instance type offers different compute and memory capabilities. Select an instance type based on the amount of memory and computing power that you need for the application or software that you plan to run on the instance. You can launch multiple instances from an AMI.

Your instance keeps running until you stop or terminate it, or until it fails. If an instance fails, you can launch a new one from the AMI.

Cloud Computing Infrastructure: 1 User

- Amazon Route 53 for DNS
- A single Amazon EC2 instance
 - With full stack on this host
 - Web app
 - Database
 - Management
 -



Lets start with a simple implementation for a single user...

This is the most basic set up you would need to serve up a web application.

Every user first encounters Amazon Route53 for DNS resolution.

Behind the DNS service is an EC2 instance running a full stack including your web app and database on a single server.

To scale this infrastructure, the only real option you have is to get a bigger EC2 instance.

AWS Core Services Summary

Compute



Amazon
EC2

Networking



Amazon
Route 53

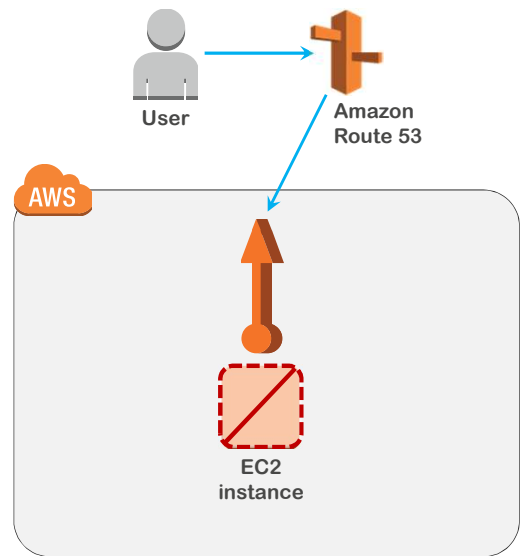
Storage

Database

So far we've looked at EC2 and Route 53...

Cloud Computing Infrastructure: 1 User

Challenges: Single Point of Failure



Let's consider what might happen if there were some kind of failure in the instance. There is no failover, no redundancy, and too much reliance on one instance. Your database and web app are on the same instance.

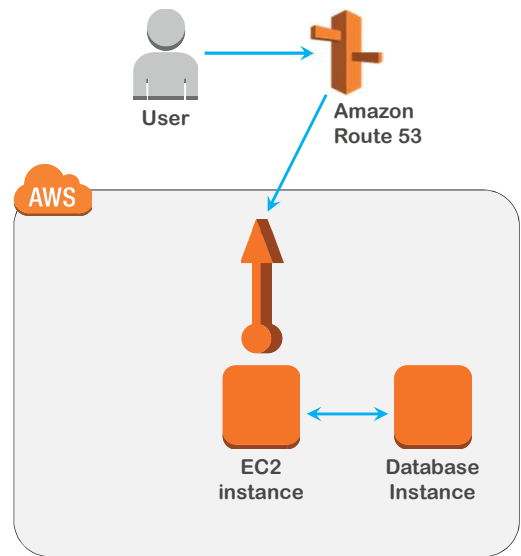
You need to be more strategic and start to break apart the application for both scaling and redundancy reasons.

Cloud Computing Infrastructure: Users > 1

Separate out the single host into:

- 📦 Web host
- 📦 Database host

Make use of a database service?



The first thing you can do to address the issue of too much reliance on one instance is to split out your web app and database into two instances. This gives you more flexibility in scaling these two tiers independently. And because you are breaking out the database, this is a great time to think about making use of a database service instead of managing the DB yourself. So what options do you have?

AWS Database Services: Database Options

Self-managed



Database server on Amazon EC2

- Bring Your Own License (BYOL)

Fully managed



Amazon RDS

- BYOL or
- License included
- MySQL, Oracle, Postgres, SQL Server
- [Amazon Aurora](#)



Amazon DynamoDB

- Seamless scalability
- Zero administration
- NoSQL



Amazon Redshift

- Petabyte-scale data
- Easy to scale, fast

At AWS there are many different options for running databases. One is to just install any database you can think of on an EC2 instance, and manage all of it yourself. If you are really comfortable doing DBA like activities, such as backups, patching, security, and tuning, this could be an option for you. If you need something highly specialized or customized and need to manage the hardware to achieve this, again this might be for you.

If not, then AWS has a few options that we think are a better idea:

First is **Amazon RDS**, or Relational Database Service. With RDS you get a **managed database instance** of either **MySQL**, **Oracle**, **Postgres**, or **SQL Server**, with features such as automated daily backups, simple scaling, patch management, snapshots and restores, high availability, and read replicas—depending on the engine you go with. Amazon Aurora is also currently in preview.

Amazon Aurora is a **MySQL-compatible relational database** that combines the speed and availability of high-end commercial databases with the simplicity and cost-effectiveness of open source databases. Aurora provides up to five times better performance than MySQL at a price point one-tenth that of a commercial relational databases while delivering similar performance and availability.

Next is **DynamoDB**, a **NoSQL database**, built on top of SSDs. DynamoDB is based on the Dynamo whitepaper published by Amazon.com back in 2003. This whitepaper was

considered the grandfather of most modern NoSQL databases like Cassandra. DynamoDB is an evolution of that whitepaper. One of the key concepts to DynamoDB is what we call “Zero Administration.” With DynamoDB, the only knobs to tweak are the reads and writes per second you want the DB to be able to perform. You set it, and it will give you that capacity with query responses averaging in single-digit milliseconds. We’ve had customers with loads such as half a million reads and writes per second without DynamoDB even blinking.

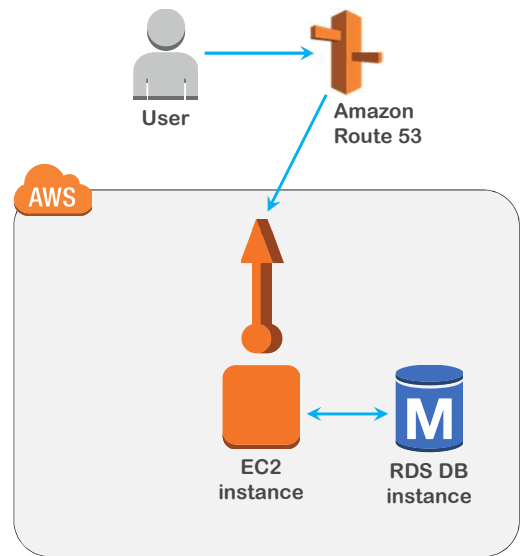
Finally, **Amazon Redshift** is a **multi-petabyte-scale data warehouse service**. Redshift is managed and massively parallel, and you use **ANSI SQL** with it over the wire. With Redshift, much like most AWS services, the idea is that you can start small and scale as you need to, while only paying for what you are using. What this means is that you can start on a smaller single node with Redshift and scale your DW cluster as your workload requires. Redshift is also several times cheaper than most other data warehouse providers.

Cloud Computing Infrastructure: Users > 100

Separate out the single host into:

- Web host
- Database host

Amazon RDS: make your life easier



So for this scenario today and based upon this discussion, you will use RDS and MYSQL as your database engine.

AWS Core Services Summary

Compute



Amazon
EC2

Networking



Amazon
Route 53

Storage

Database

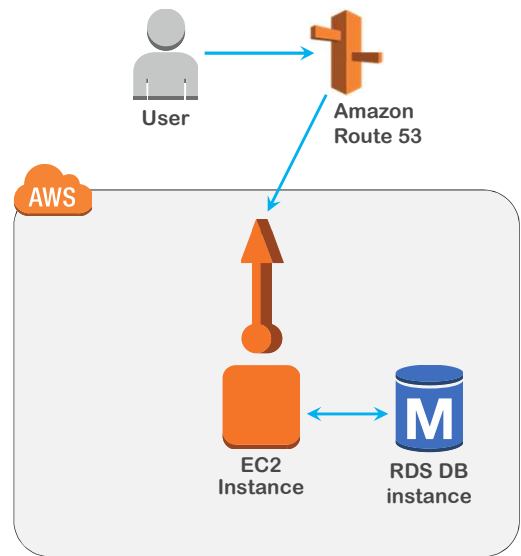


Amazon
RDS
Amazon Aurora

So, we'll add RDS and Aurora to our list.

Cloud Computing Infrastructure: Users > 1000

Challenge: No failover, no redundancy

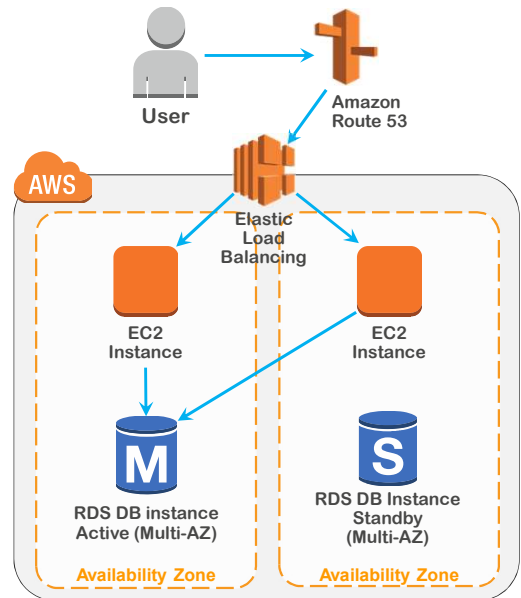


As we ramp-up to over 1,000 users, this limited application doesn't scale well. There is no failover and no redundancy providing **high availability**.

Cloud Computing Infrastructure: Users > 1000

Solution: High Availability

- Another web instance
In another Availability Zone
- RDS Multi-AZ
- Elastic Load Balancing (ELB)



To address the lack of failover and redundancy in your infrastructure, let's consider distributing resources over multiple Availability Zones (AZs).

You will do this by adding another web app instance and enabling the **Multi-AZ** feature of RDS, which will give you a standby instance in a different AZ from the primary zone. By provisioning your compute resources across multiple AZs, **problems in a single AZ will not affect resources in another AZ.**

You will use an **Elastic Load Balancing** to share the load between your two web instances.

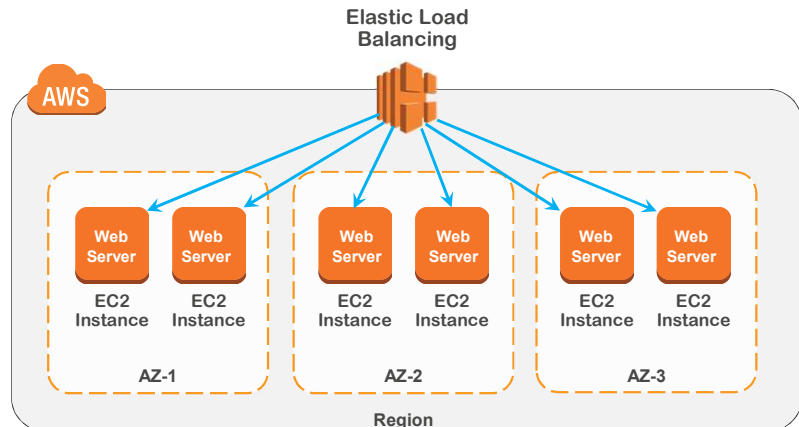
Now you have an app that is a bit more scalable and also has some fault tolerance built in.

AWS Compute Services



Elastic Load Balancing

- Health checks on hosts
- Distribution of traffic
- Dynamic addition and removal of EC2 hosts



Two Types: { **Classic Load Balancer:** balance traffic on network layer (HTTP(S), TCP/SSL)
Application Load Balancer: balance traffic on application level

Elastic Load Balancing **automatically distributes incoming application traffic** across multiple Amazon EC2 instances. It enables you to achieve **fault tolerance** in your applications and seamlessly provides the required amount of load balancing capacity needed to route application traffic.

Elastic Load Balancing offers two types of load balancers that both feature high availability, automatic scaling, and robust security. These include the *Classic Load Balancer* that routes traffic based on either application or network level information, and the *Application Load Balancer* that routes traffic based on advanced application-level information that includes the content of the request. The Classic Load Balancer is ideal for simple load balancing of traffic across multiple EC2 instances, and the Application Load Balancer is ideal for applications that need advanced routing capabilities, microservices, and container-based architectures. The Application Load Balancer offers the ability to route traffic to multiple services or load-balance across multiple ports on the same EC2 instance.

AWS Core Services Summary

Compute



Amazon
EC2



Elastic Load
Balancing

Networking



Amazon
Route 53

Storage

Database

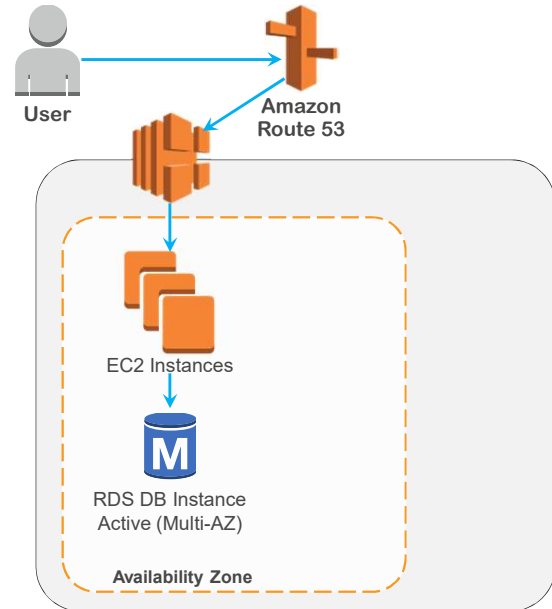


Amazon
RDS
Amazon Aurora

So, we'll add Elastic Load Balancing to the list.

Cloud Computing Infrastructure: Users > 10,000s–100,000s

Challenge: High Performance and
Cost Efficiency



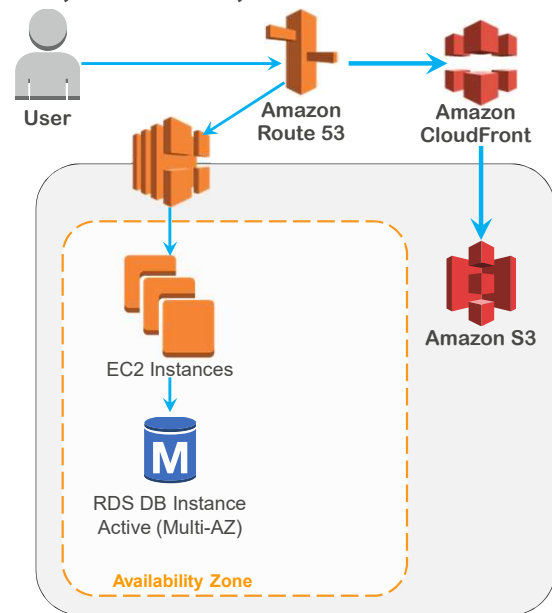
This will work, but there is more we can do to design the system for high performance while still achieving greater cost efficiency.

Cloud Computing Infrastructure: Users > 10,000s–100,000s

Challenge: **High Performance and Cost Efficiency**

Static content to:

- Amazon S3
- Amazon CloudFront



As mentioned, you can start by moving any **static assets** from your web app instances to **Amazon S3** and then serve those objects via **CloudFront**. This would include all of your **images, videos, CSS, JavaScript files**, and any other **heavy static content**.

These files can be served via an S3 origin (more on S3 in the next slide) and then globally cached and distributed via Cloudfront. This will take the load off your web servers and allow you to reduce your footprint in that web tier.

AWS Storage Services



Amazon
S3



- Object storage and distribution for the internet
- 99.999999999% durability
- Storage classes
 - ✓ Standard
 - ✓ Standard – Infrequent Access
 - ✓ Glacier

As mentioned before, you can use S3 to lighten the load. What is S3?

S3 is cloud object storage for the Internet where files are reachable via a restful URL. Files can be locked down so they are only reachable by a specific IAM user or role, or they can be made public and can be served to public Internet users; this is what you need.

It has 11 9s (99.999999999%) of durability – what does that mean?

S3 ties in tightly with many services that can store and get data from S3.

it is also a great logging endpoint for many of our services such as ELB, CloudTrail, and CloudFront.

Data can also be tiered off to **Glacier** which is our **archival storage service** at 1 cent /GB / month.

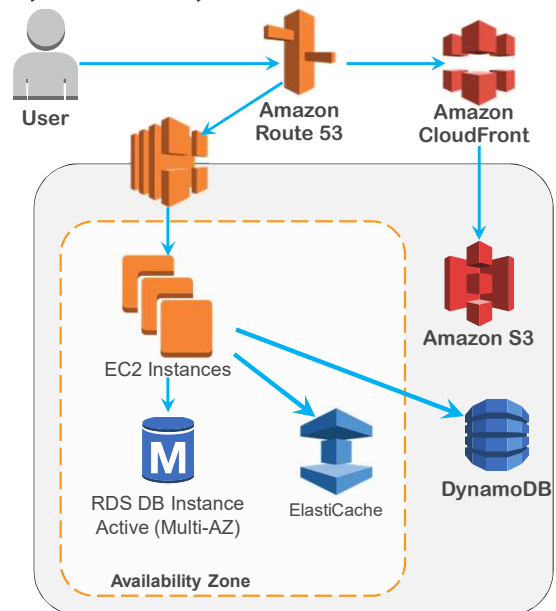
Lastly, S3 can be used as the origin for our global CDN, CloudFront.

Amazon S3 offers a range of storage classes designed for different use cases, including Amazon S3 Standard for general-purpose storage of frequently accessed data, Amazon S3 Standard - Infrequent Access (Standard - IA) for long-lived, but less frequently accessed data, and Amazon Glacier for long-term archiving.

Cloud Computing Infrastructure: Users > 10,000s–100,000s

Shift some load around

- Static content to Amazon S3 and Amazon CloudFront
- Session/state to Amazon DynamoDB
- DB caching to Amazon ElastiCache



You can also move things like session information to a NoSQL DB like **DynamoDB** or to a cache like **ElastiCache**. For this scenario, you can use DynamoDB for this because there are easy connectors in many of the AWS SDKs.

You can also use **ElastiCache** to **store some of your common database query results**, which will prevent you from accessing the database too much.

This should take a load off of your DB tier.

Removing session state from your web/app tier is also very important because it allows you to scale up and down **without losing session information** when horizontal scaling happens. This is called making your tier “**stateless**.”

AWS Core Services Summary

Compute



Amazon
EC2



Elastic Load
Balancing

Networking



Amazon
Route 53

Storage



Amazon S3



CloudFront

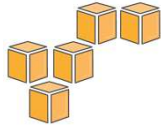
Database



Amazon
RDS
Amazon Aurora



DynamoDB



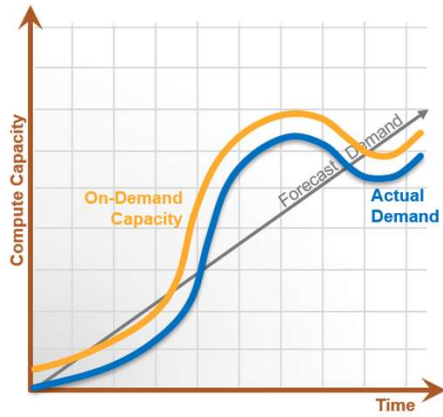
Part 2: Auto Scaling

Part 2: Auto Scaling

In order to **gain agility, elasticity and scalability**, you can leverage Auto Scaling.

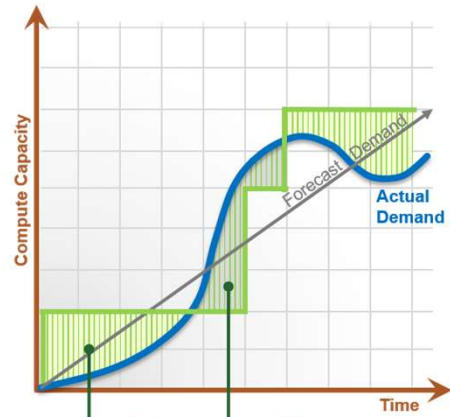
Cloud vs. On-Premises Comparison

Cloud

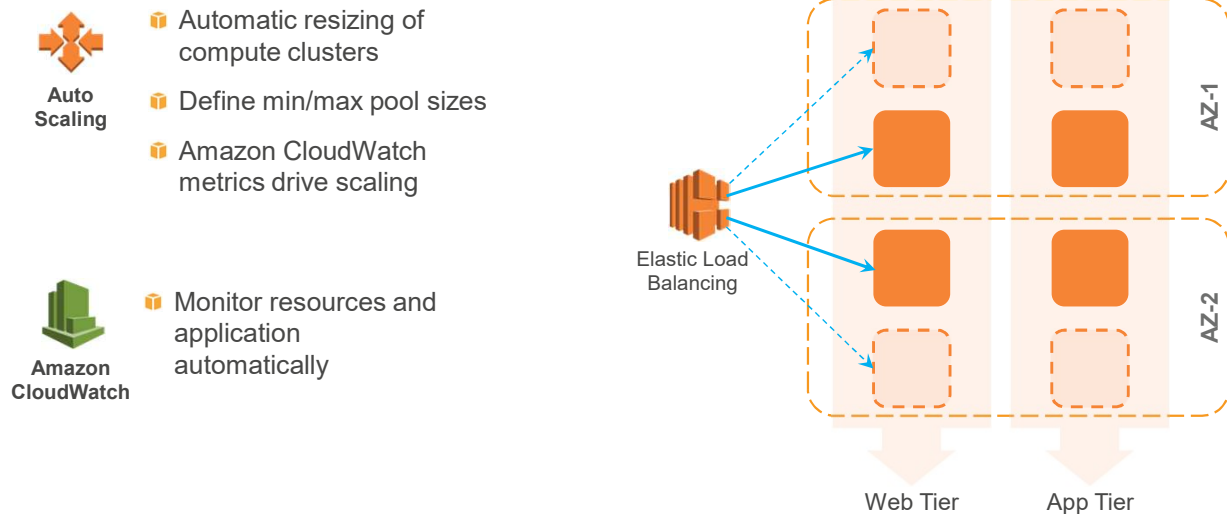


- ✓ No waste
- ✓ Meets demand

On-Premises



AWS Compute and Management Services



Auto Scaling is the automatic resizing of application tiers or scaling of compute clusters or tiers – so you can grow or shrink your web and app tiers as needed.

You first create a launch configuration, which defines what each instance you launch will look like – this includes features like a choice of **AMI**, what instance type you want to use, any bootstrapping, etc.

Next, you create an **Auto Scaling group**. In this group you define the min and max servers in the group, the Availability Zones you want the tier to operate, and the Launch config name.

Finally, you specify your scaling policies with metrics or a schedule.

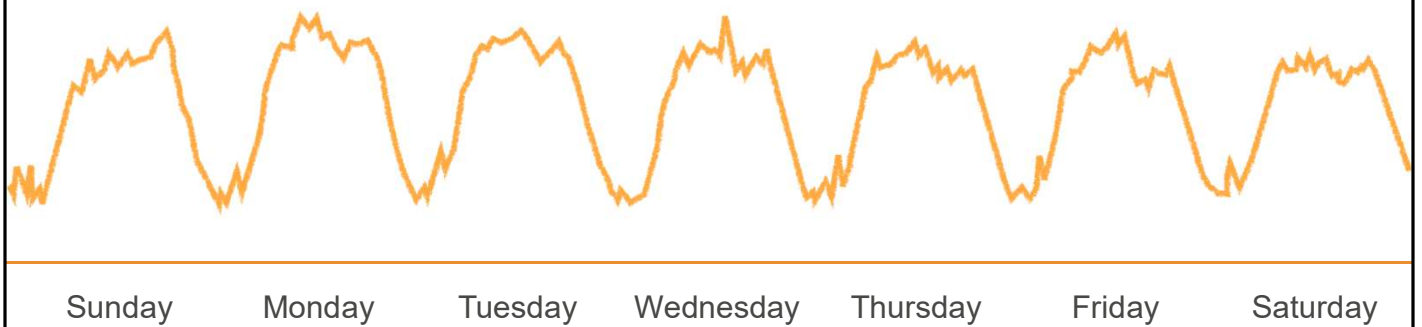
Using CloudWatch, you can easily **monitor** as much or as little metric data as you need.

CloudWatch lets you programmatically retrieve your monitoring data, view graphs, and set alarms to help you troubleshoot, spot trends, and take automated action based on the state of your cloud environment.

CloudWatch provides a reliable, scalable, and flexible monitoring solution that you can start using within minutes. You no longer need to set up, manage, or scale your own monitoring systems and infrastructure.

CloudWatch is accessible via the AWS Management Console, APIs, SDKs, or a Command Line Interface.

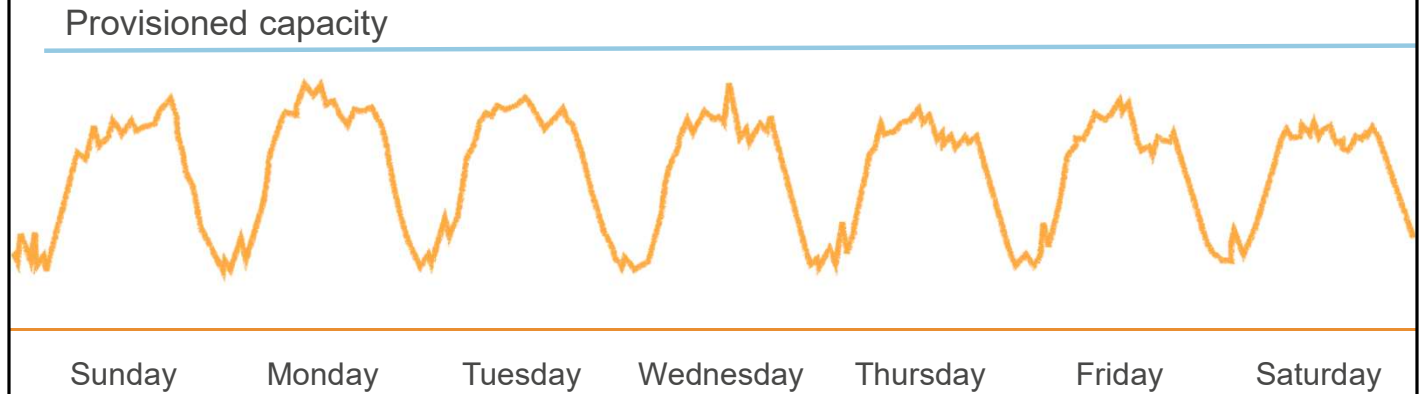
Typical Weekly Traffic to Amazon.com



© 2017 Amazon Web Services, Inc. or its affiliates. All rights reserved.

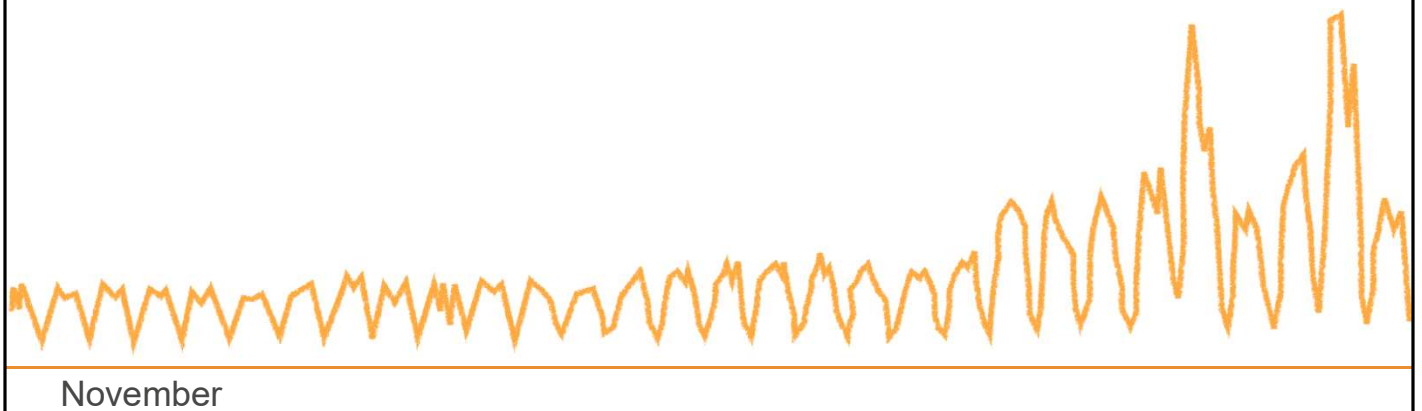
Imagine a “typical” week of traffic to Amazon.com. This pattern might look a lot like your own traffic, with a peak during the middle of the day and a valley in the middle of the night.

Typical Weekly Traffic to Amazon.com



Given this pattern, it becomes really easy to do capacity planning. Amazon.com can provision 15% over what we see our normal peak to be, and be happy with the capacity we have for a while, so long as our traffic matches this pattern.

November Traffic to Amazon.com



And this was the month of November! You can see a pretty big growth here at the end of the month with Black Friday and Cyber Monday sales in the US.

November Traffic to Amazon.com

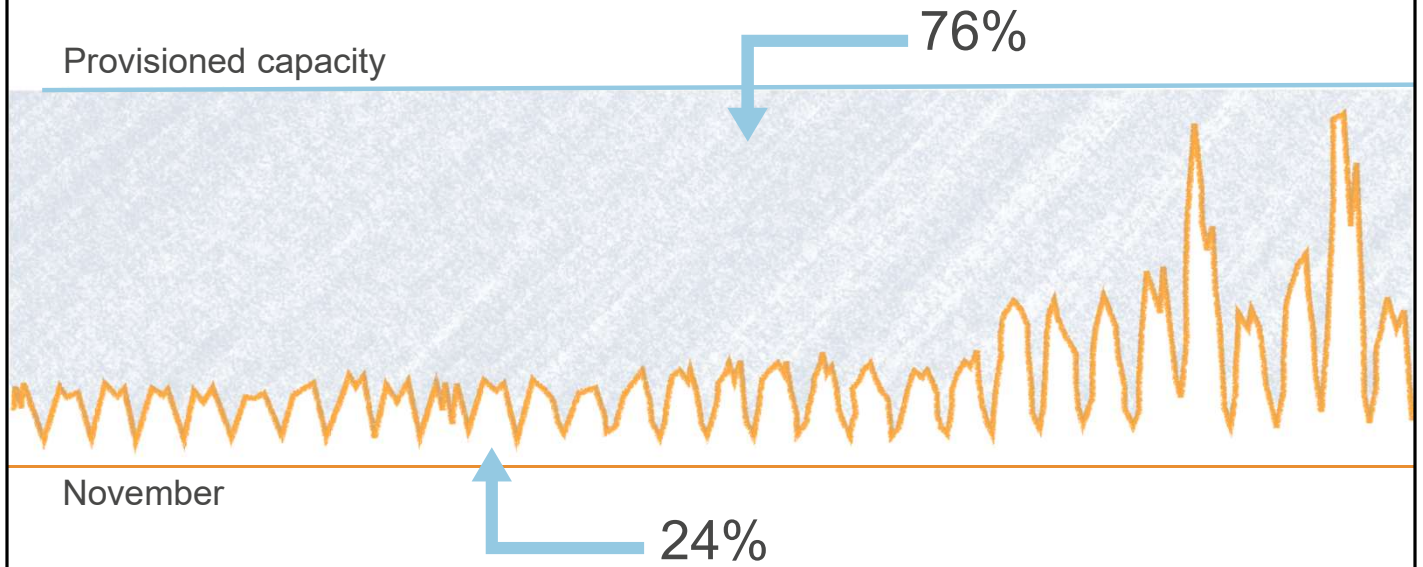
Provisioned capacity



November

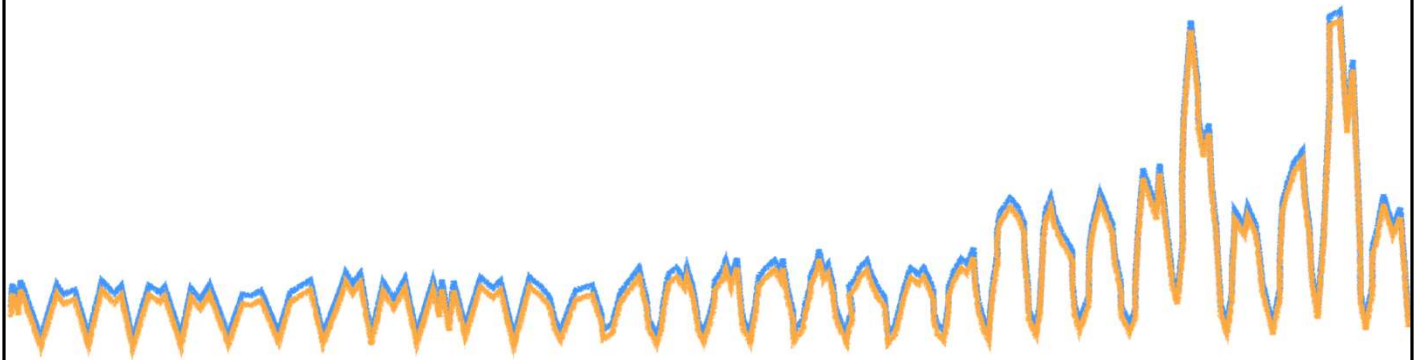
IF we attempted to implement our "add 15% capacity for spikes" rule, we'd be in trouble for the month of November.

November Traffic to Amazon.com



That's a lot of potential wasted infrastructure and cost. 76% wasted potentially, while only 24% of it on average for the month gets utilized. Traditionally this is how IT did things. You bought servers for a 6-12 month vision on what growth might be.

November Traffic to Amazon.com



November

What if we could map our capacity directly to our needs based on our end users? We could make sure that at any point in time, we were only provisioning what we needed, vs. some arbitrary capacity guess. Auto Scaling lets you do this!

Note that Auto Scaling is not a paid service, but rather how you architect your system.

AWS Core Services Summary

Compute



Amazon
EC2



Elastic Load
Balancing



Auto
Scaling

Networking



Amazon
Route 53

Storage



Amazon S3



CloudFront

Database



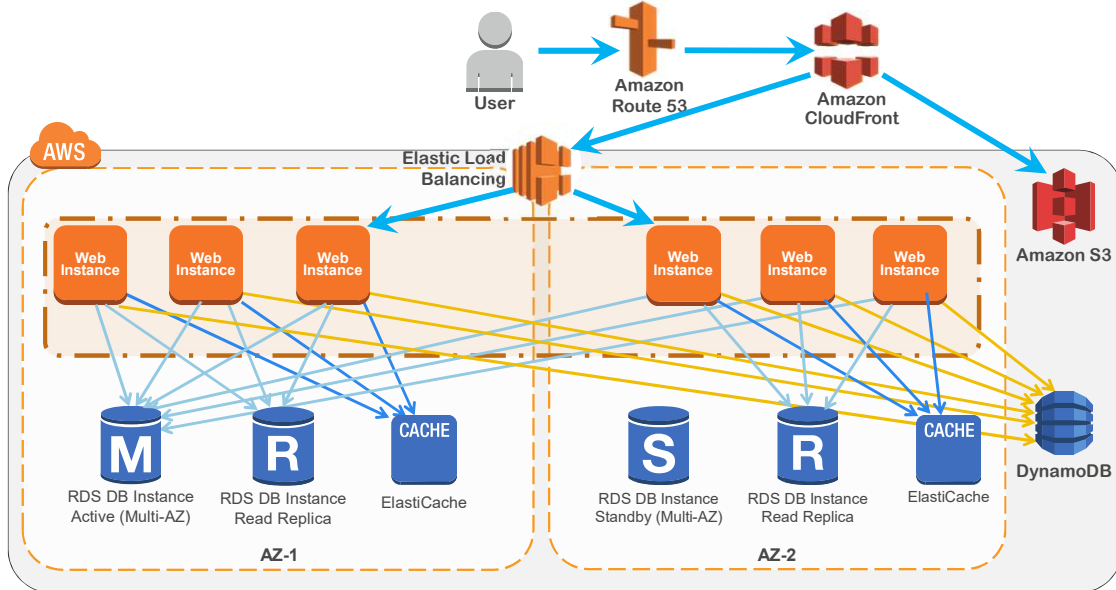
Amazon
RDS
Amazon Aurora



DynamoDB

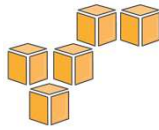
We've now added Auto Scaling to our list.

Cloud Computing Infrastructure: Users > 500,000





From our earlier example, we were able to handle a large load after adding Auto Scaling to the caching layer (both inside and outside the infrastructure), and the read-replicas with MySQL. But this is a monolith. ***All of the application logic is running on each server.***

Can you make this better?



In review...

-  Putting the Pieces Together
-  Auto Scaling

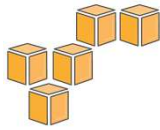


Knowledge Assessment

In review...

- Putting the Pieces Together
- Auto Scaling

To complete this module, please remember to finish the corresponding knowledge assessment.



Up Next...

CCA 1.03: Cloud Economics

Up next is your first lab, **Creating an EC2 instance with Microsoft Windows**. Please refer back to the Welcome module for instructions on accessing the lab environment.

Be sure to complete the lab before continuing with **CCA 1.03** covering economic aspects of cloud computing and total cost of ownership (TCO).

© 2017 Amazon Web Services, Inc. or its affiliates. All rights reserved.

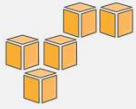
This work may not be reproduced or redistributed, in whole or in part, without prior written permission from Amazon Web Services, Inc. Commercial copying, lending, or selling is prohibited.

Errors or corrections? Email us at aws-course-feedback@amazon.com.

For all other questions, contact us at
<https://aws.amazon.com/contact-us/aws-training/>.

All trademarks are the property of their owners.

Do not speak over this slide – just let it play for 8 seconds.



CCA Unit 1 – Introduction to Cloud Computing

CCA 1.03: Cloud Economics

CCA 1.01: What is Cloud Computing?

CCA 1.02: Leveraging Cloud Computing

▶ **CCA 1.03: Cloud Economics**

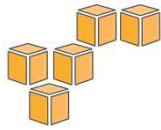
CCA 1.03: Cloud Economics

What's In This Module

- 📦 Cost Optimization
- 📦 Total Cost of Ownership (TCO)

This module covers...

- Cost Optimization
- Total Cost of Ownership (TCO)



Part 1 Cost Optimization

Part 1: Cost Optimization

What Is Cost Optimization?



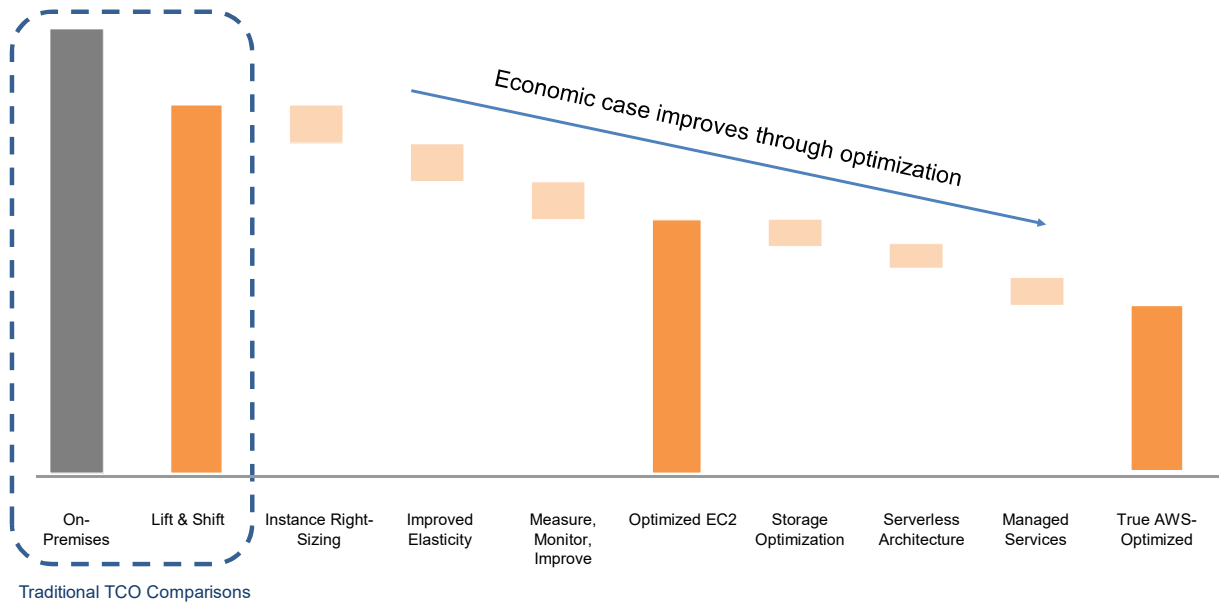
Reduce Costs...

Pay only for **what** you need
when you need it.

One of the most common reasons to move into the cloud is to **reduce costs**. In reducing costs it's very important to be able to **optimize spend** and **pay only for what you need** and **when you need it**. When you optimize costs, you can help your organization get the most out of your investments helping meet demand and capacity while using the most economically effective options.

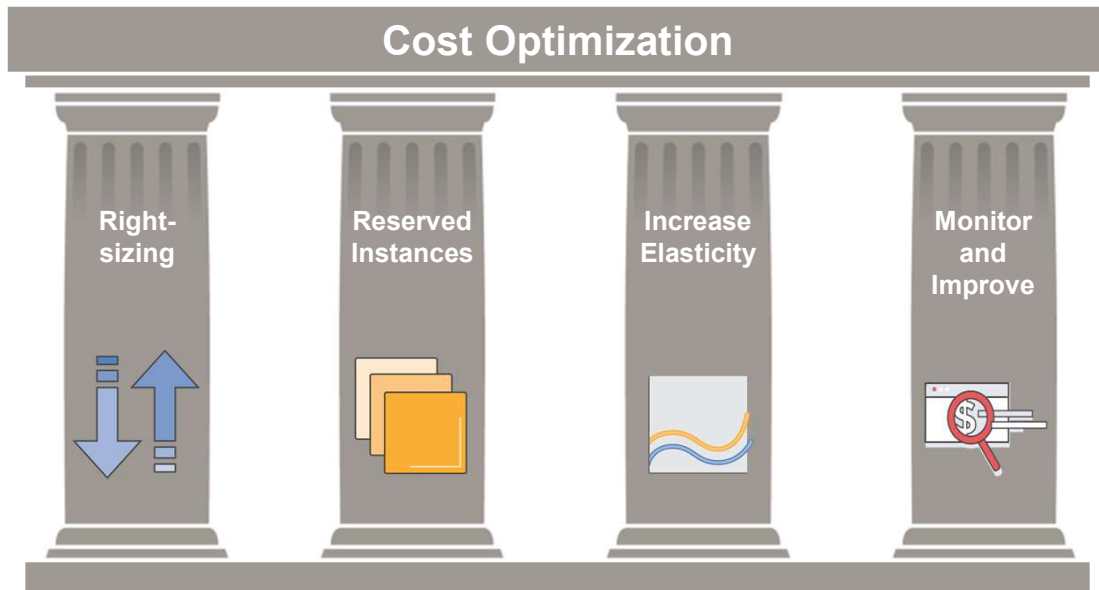
Cost optimization using cloud has brought about new business models enabling organizations to be lean with what they use and reduce their spend dramatically.

Lowering TCO Through Cost Optimization



The initial lift and shift model doesn't fully capture the on-going economic case for the cloud. Cost optimization over time continues to drive down costs through ongoing improvements, managed services, and an expanded scope of analysis beyond just EC2 (for example, RDS, Lambda, and storage), etc.

The Four Pillars of Cost Optimization



There are three consistent, powerful drivers of cost optimization:

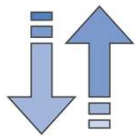
- Right-sizing - choose the right balance of instance types
- Leverage reserved instances
- Increase elasticity using auto scaling
- Monitor by measuring and analyzing your system. Continually improve and adjust as you go.

Right-Sizing

Four Pillars:

Right-Sizing

Reserved Instances
Increase Elasticity
Monitor & Improve



- ☐ Select the appropriate instance types
- ☐ Downsize instances
- ☐ Leverage Amazon CloudWatch metrics

AWS offers approximately 60 instance types and sizes (<https://aws.amazon.com/ec2/instance-types/>), which is great for customers because it allows them to select the best fit instance for their workload. It can also be a bit difficult to know where to start and what the best instance is from a cost perspective—not just technically.

- We define the right size as the cheapest instance or storage type available that meets performance requirements.
- Right-sizing is the process of looking at deployed resources and looking for opportunities to downsize when possible.
- Testing is cheap: you can easily provision any type and size of instance to test your application on; use this advantage.

Right-Sizing:

- Select the cheapest instance available while meeting performance requirements.
- Look at CPU, RAM, storage, and network utilization to identify potential instances that can be downsized.
- Leverage Amazon CloudWatch metrics and set up custom RAM metrics.

Rule of thumb: Right size, then reserve.

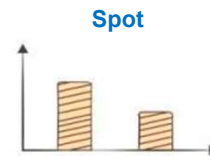
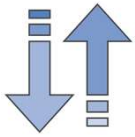
(But if you're in a pinch, reserve first.)

Optimize and Combine EC2 Purchase Types

Four Pillars:

Right-Sizing

Reserved Instances
Increase Elasticity
Monitor & Improve



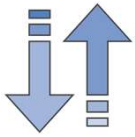
AWS provides a number of pricing models for EC2 to help customers save money. Customers can combine multiple purchase types to optimize pricing based on their current and forecast capacity needs.

Optimize and Combine EC2 Purchase Types

Four Pillars:

Right-Sizing

Reserved Instances
Increase Elasticity
Monitor & Improve



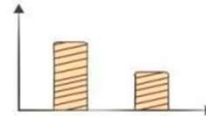
On-Demand



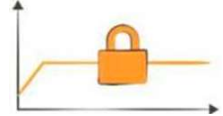
Reserved



Spot



Dedicated



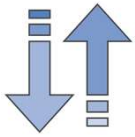
- Pay by the hour.
- No long-term commitments

Optimize and Combine EC2 Purchase Types

Four Pillars:

Right-Sizing

Reserved Instances
Increase Elasticity
Monitor & Improve



On-Demand



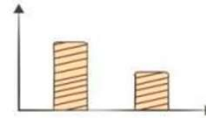
- Pay by the hour.
- No long-term commitments

Reserved

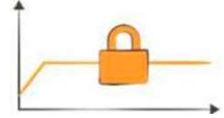


- Pay upfront
- 50-75% lower hourly rate

Spot



Dedicated



Reserved Instances give you the option to make one upfront payment for each instance you want to reserve at a significant discount. Using the reserved option, you can get a rate that is 50–75% lower than the on-demand rate.

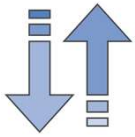
Reserved pricing works well for steady-state workloads with committed utilization.

Optimize and Combine EC2 Purchase Types

Four Pillars:

Right-Sizing

Reserved Instances
Increase Elasticity
Monitor & Improve



On-Demand



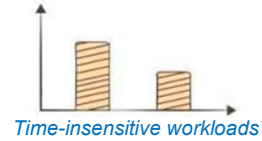
- Pay by the hour.
- No long-term commitments

Reserved



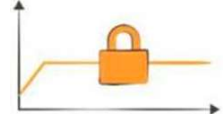
- Pay upfront
- 50-75% lower hourly rate

Spot



- Bid for unused EC2 capacity

Dedicated



Spot Instances enable you to bid for unused Amazon EC2 capacity. Instances are charged the Spot Price, which is set by Amazon and fluctuates depending on the supply of and demand for Spot Instance capacity.

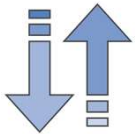
Spot pricing offers the best hourly rate and works best for workloads that are not time-dependent and which can afford to be interrupted.

Optimize and Combine EC2 Purchase Types

Four Pillars:

Right-Sizing

Reserved Instances
Increase Elasticity
Monitor & Improve



On-Demand



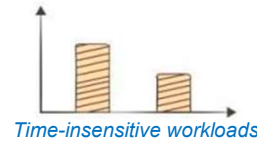
- Pay by the hour.
- No long-term commitments

Reserved



- Pay upfront
- 50-75% lower hourly rate

Spot



- Bid for unused EC2 capacity

Dedicated



- In your VPC
- Isolated, steady-state workloads

Dedicated Instances run on hardware dedicated to a single customer. Dedicated Instances ensure that your Amazon EC2 compute instances are isolated at the hardware level.

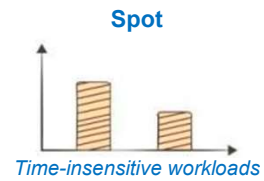
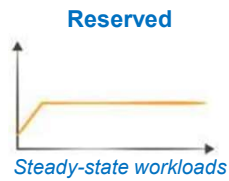
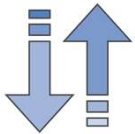
Some customers use Dedicated Instances to allow them to run third-party software, where the licensing model demands that the hardware is dedicated to one tenant. They then fill the rest of their workloads with either On-Demand or Spot Instances.

Optimize and Combine EC2 Purchase Types

Four Pillars:

Right-Sizing

Reserved Instances
Increase Elasticity
Monitor & Improve



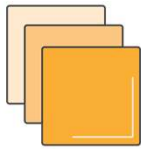
- ✓ Pay only for what you use
- ✓ On-demand, elastic provisioning
- ✓ Control and security

The AWS pricing models help you optimize your cost savings based on your unique requirements. You can take full advantage of cloud computing benefits and on-demand elastic provisioning with the control and security your applications require, while paying only for what you use.

Reserved Instance Capacity

Four Pillars:

Right-Sizing
Reserved Instances
Increase Elasticity
Monitor & Improve



Reserved Instances/Capacity

- Amazon EC2
- Amazon RDS
- Amazon DynamoDB
- Amazon Redshift
- Amazon ElastiCache

Commitment level

- 1 year
- 3 years

Up to 75%+
savings

* Dependent on specific AWS service, size/type, and region

After you have settled on an instance type, you have the option of purchasing a Reserved Instance. This is an **upfront commitment to purchase capacity** in a particular AWS region, which will **dramatically reduce your running costs**. A **Reserved Instance is a billing construct**; it ensures you have capacity available in the Availability Zones you have selected and purchased for that instance type.

Reserved instances are a great choice for predictable work loads. You can make capacity reservations for these work loads saving up to 75% over the on-demand hourly rates depending on the RI type you've chosen. For more information and a quick introductory video, visit <https://aws.amazon.com/ec2/pricing/reserved-instances/>.

Besides treating a Reserved Instance as a 24x7 resource, it is also possible for you to combine a Reserved Instance if your workload is time-dependent. For example, let's assume you have a Reserved Instance for a multi-purpose instance type like an m4.large. You only need to run this instance during office hours for a total of nine hours (8:00am to 5:00pm). However, you have another workload in the same Availability Zone that can use the same instance type and be run after office hours (5:00pm to 8:00am). You could select the same instance type (m4.large) and start the evening workload on that instance after the daytime instance has shut down. After the first instance is shut down, the Reserved Instance hourly rate will apply to the after-hours instance, thus maximizing your overall cost efficiency.

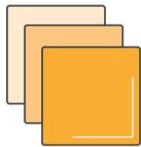
It's important to **continually reevaluate your instance selection**, because workloads and instance types will change over time.

Reserved Instances are currently offered as one- or three-year commitments, and your requirements may change before the Reserved Instance commitment expires.

Reserved Instances

Four Pillars:

Right-Sizing
Reserved Instances
Increase Elasticity
Monitor & Improve



Step 1: RI Coverage

- Cover always-on resources
- Target 70–80% always-on coverage

Step 2: RI Utilization

- Leverage RI flexibility to increase utilization
- Merge and split RIs as needed
- Target 95% RI utilization rate

Reserved Instances

Step 1: Reserved Instance Coverage - Cover always-on resources with standard or convertible Reserved Instances

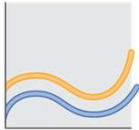
Step 2: Increase Reserved Instance Utilization

- Known architectures: Leverage Standard Reserved Instance flexibility to increase utilization.
- Growing or changing architectures: Leverage Convertible Reserved Instances across families, sizes, and OS.
- Regional Benefit: Consolidated billing, reservation not critical

Increase Elasticity

Four Pillars:

Right-Sizing
Reserved Instances
Increase Elasticity
Monitor & Improve



Turn off non-production instances

Example: Dev/test

Auto scale production

Use Auto Scaling to scale up and down based on demand and usage (e.g., spikes)

Elasticity is using an instance when you need it, but turning it off when you don't. It's one of the most central tenets of the cloud, but we often see customers go through a learning process to operationalize this in order to drive cost savings.

The easiest way for large customers to leverage this is to look for the “low-hanging fruit, such as non-prod environments or dev/test workloads. If you're running dev/test out of a single time zone, for example, you can easily **turn off those instances outside of business hours** and reduce their cost by 80%. There's a reason why the light switch is by the door: turn off the lights on your way out of the office each night.

For production workloads, getting more precise and granular with auto scaling is going to help ensure that you're able to take advantage of horizontal scaling in order to meet peak capacity needs, while not paying for peak capacity.

As a rule of thumb: You should be targeting 20-30% of your ec2 instances running on-demand or on spot, and you should be looking to maximize elasticity within this group.

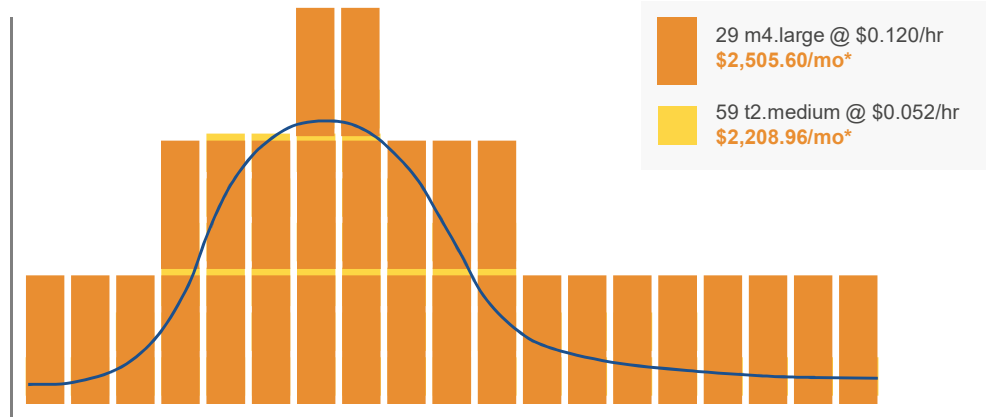
Using Right-sizing and Elasticity to Lower Cost

Four Pillars:

Right-Sizing
Reserved Instances
Increase Elasticity
Monitor & Improve



More, smaller instances vs. fewer, larger instances



*Assumes Linux instances in the US-East (N. Virginia) Region at 720 hours per month

Do not manage the cloud like you would manage a data center: this is a new operational model.

14% savings from using t2s vs. m4s above.

Measure, Monitor, and Improve

Four Pillars:

Right-Sizing
Reserved Instances
Increase Elasticity
Monitor & Improve



Cost Optimization Opportunities

1. Auto-tag resources
2. Identify always-on nonprod
3. Identify instances to downsize
4. Recommend RIs to purchase
5. Dashboard your status
6. Report on savings

At Amazon, automation is the key to success at scale. So let's talk about a few things to get you going to be able to provide the insights that are needed to drive cost optimization .

The first step is setting up tools to help understand the opportunity. Tagging helps provide information about *what resources* are being used *by whom* and *for what purpose*.

Second, you want tools that identify those resources that you can take action on quickly. Set up automated reports that identify instances that are not being turned off or that run at the wrong size.

Setting up an automated report to determine what instances to downsize is important, because if you're looking at thousands of instances and trying to determine what type of instance should be run, you want an automated tool or report to do that for you. Next, you need a tool to recommend which RIs to buy. AWS provides recommendations through Trusted Advisor, and several partners (including CloudAbility, Cloud Checkr, Cloudyn, and Cloud Health) also have good tools.

Last, it's important to report on cost optimization in order to show the opportunities that exist and show how you're progressing.

Tools to Measure, Monitor, and Improve

Four Pillars:

Right-Sizing
Reserved Instances
Increase Elasticity
Monitor & Improve



AWS Trusted Advisor

- 📦 Optimize your AWS environment
- 📦 Reduce cost, increase performance, and improve security

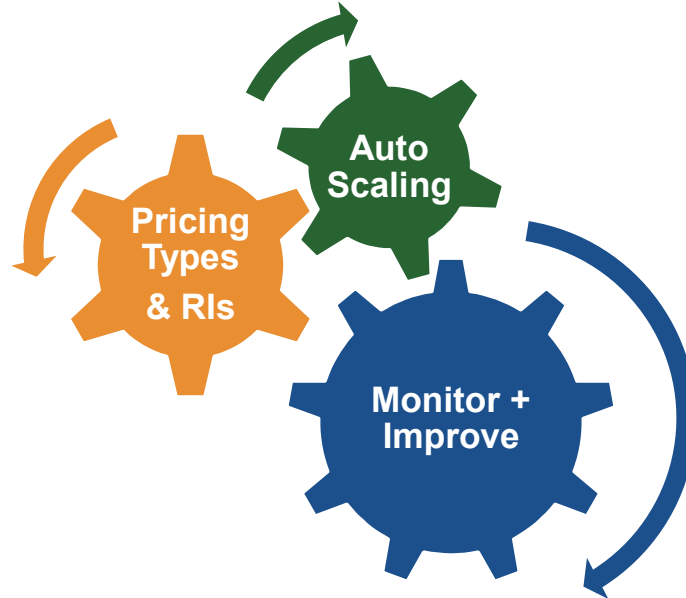


Cost Explorer

- 📦 View graphs of your costs: the last 13 months
- 📦 Forecast your likely costs: the next 3 months
- 📦 View time data by day or month

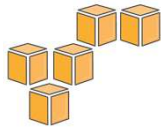
An online resource to help you reduce cost, increase performance, and improve security by optimizing your AWS environment, AWS Trusted Advisor provides real time guidance to help you provision your resources following AWS best practices. Cost Explorer is a free tool that you can use to view graphs of your costs (also known as spend data) for up to the last 13 months, and forecast how much you are likely to spend for the next three months. You can use Cost Explorer to see patterns in how much you spend on AWS resources over time, identify areas that need further inquiry, and see trends that you can use to understand your costs. You can also specify time ranges for the data you want to see, and you can view time data by day or by month.

Continual Process of Cost Optimization



Cost optimization is an continual and interdependant process.

- Select the appropriate pricing models (instance types), and leverage RIs according to your business application.
- Increase your elasticity by using auto scaling and turning off non-production instances.
- Leverage AWS tools to analyze, monitor, and improve your costs.



Part 2: Total Cost of Ownership (TCO)

Part 2: Total Cost of Ownership (TCO)

What Is TCO?

Comparative total cost of ownership analysis

- 1) On-premises/co-location vs. cloud deployment
- 2) Acquisition cost + operating costs
- 3) Entire infrastructure environment for a specific workload

What is Total Cost of Ownership, or TCO?

Definition: **Comparative** total cost of ownership analysis (acquisition and operating costs) for running an infrastructure environment end-to-end on-premises vs. cloud deployment.

Used for:

Comparing the costs of running an entire infrastructure environment for a specific workload in an **on-premises** or **co-location** facility to the same workload running on a **cloud-based** infrastructure.

Budgeting and building the business case for business decisions.

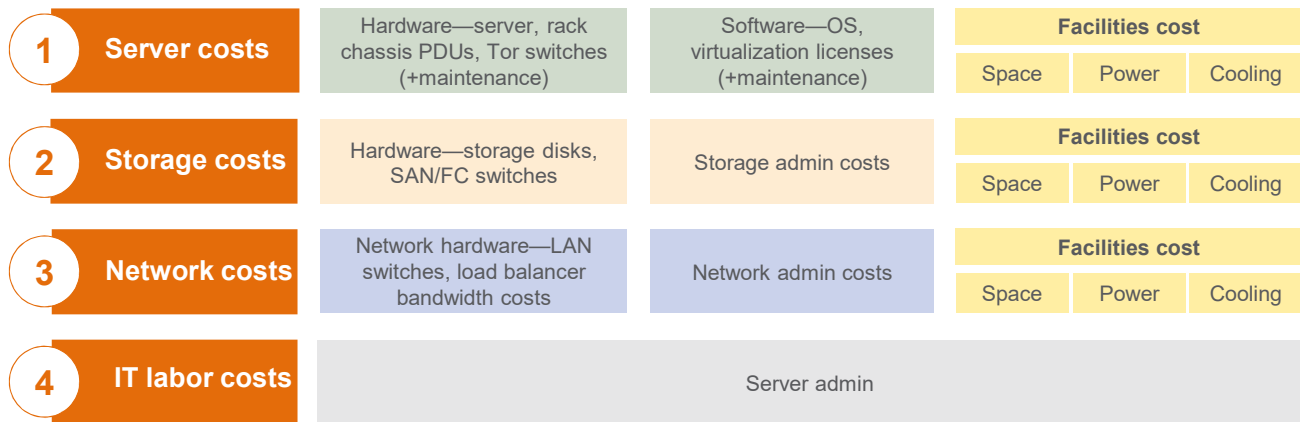
Not Easy to Compare!



On-premises/co-location



It is not easy to compare an on-premises IT delivery model with the AWS Cloud. The two are so very different that they use different languages: AWS is a discussion about flexibility, agility, and consumption bases costs: Units/Time. On-premises IT is a discussion based on capital expenditure, long planning cycles, and multiple components to buy, build, manage, and refresh over time.

Costs Involved in Data Center Maintenance

The diagram is conceptual and does not include every cost item. For example, software costs can include database, management, and middle-tier costs. Facilities costs can include upgrades, maintenance, building security, taxes, and so on. IT labor costs can include security admin and application admin costs. So, this is an abbreviated list to demonstrate what kinds of costs are involved in data center maintenance.

Case Study: Total Cost of Ownership



**Delaware
North**

Background:

- Growing, global company with over 200 locations
- 500 million customers, \$3 billion annual revenue

Background:

Delaware North originated in 1915 as a peanut and popcorn concessions vendor; today, it's a major food and hospitality company serving half a billion customers each year.

Although the company deliberately keeps a low profile, it is a leader in the food-service and hospitality industry, serving more than **500 million customers** annually at more than **200 locations** around the world, including venues as diverse as the Kennedy Space Center in Florida, London Heathrow Airport, Kings Canyon Resort in Australia, and the Green Bay Packers' Lambeau Field in Wisconsin. That global presence has turned Delaware North into a **\$3 billion enterprise**.

Case Study: Total Cost of Ownership



Delaware
North

Background:

- Growing, global company with over 200 locations
- 500 million customers, \$3 billion annual revenue

Challenge:

- Meet demand to rapidly deploy new solutions
- Constantly upgrade aging equipment

Challenge/Opportunity:

The company's on-premises data center was becoming too expensive and inefficient to support its global business operations, so it turned to AWS to move most of its enterprise applications and operations to the cloud.

Kevin Quinlivan, Delaware North's Chief Information Officer stated: "As the company continued to grow, the **demand to rapidly deploy new solutions** to meet customer requirements increased as well. This fact, combined with the **need to constantly upgrade aging equipment**, required an even greater commitment of resources on our part. We had to find a better strategy."

Case Study: Total Cost of Ownership



Delaware North

Background:

- Growing, global company with over 200 locations
- 500 million customers, \$3 billion annual revenue

Challenge:

- Meet demand to rapidly deploy new solutions
- Constantly upgrade aging equipment

Criteria:

- Broad solution to handle all workloads
- Ability to modify processes to improve efficiency and lower costs
- Eliminate busy work (e.g. patching software)
- Achieve a positive return on investment (ROI)

Criteria:

After a successful migration of about 50 websites to AWS in 2013, Delaware North evaluated the cost benefit and total cost of ownership to move their IT infrastructure to AWS. Their focus was to answer C-level business demands for measurable benefits that could convince an executive committee that the AWS cloud was the right approach.

The evaluation process centered on three criteria:

- 1) A cloud solution needed a broad set of technologies that could **handle all of Delaware North's enterprise workloads** while delivering support for critical functions.
- 2) From an operational perspective, Delaware North wanted the features and flexibility to **modify core IT processes to improve efficiencies and lower costs**. This included **eliminating redundant or time-consuming tasks** like patching software and pushing test and development tasks through outdated systems that, in the past, added months to the deployment of new services.
- 3) Financial requirements needed to **demonstrate a return on investment (ROI)** with a solid cost-benefit justification for moving away from their existing data center environment.

Case Study: Total Cost of Ownership



Delaware North

Background:

- Growing, global company with over 200 locations
- 500 million customers, \$3 billion annual revenue

Challenge:

- Meet demand to rapidly deploy new solutions
- Constantly upgrade aging equipment

Criteria:

- Broad solution to handle all workloads
- Ability to modify processes to improve efficiency and lower costs
- Eliminate busy work (e.g. patching software)
- Achieve a positive return on investment (ROI)

Solution:

- Moved its on-premises data center to AWS
 - Eliminated 205 servers (90%)
 - Moved nearly all apps to AWS
- Three-year EC2 Reserved Instances

Solution:

A cost comparison completed by Delaware North demonstrated that it could save \$3.5 million based on a five-year run rate by **moving its on-premises data center to AWS** and using three-year Amazon EC2 Reserved Instances (RI) and Reserved Instance renewals.

Quinlivan noted that the deep technology stack available on AWS was more than sufficient to meet the company's technical and operational requirements. And the pricing structure of the AWS offerings, which includes paying only for what is used, provided total cost of ownership benefits which was presented to senior leaders.

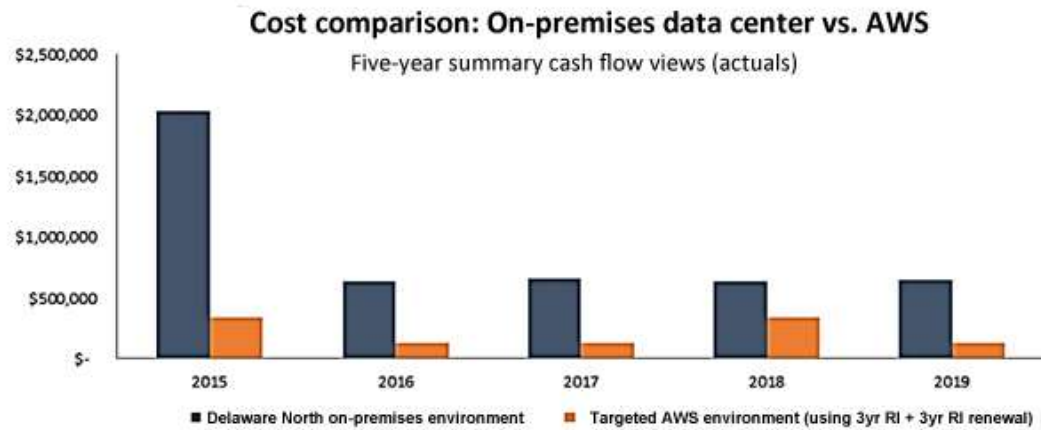
Quinlivan stated, "We compared the costs of keeping our on-premises data center versus moving to the AWS cloud, measuring basic infrastructure items such as hardware cost and maintenance," he says. "We estimate that moving to AWS will save us at least \$3.5 million over five years by **reducing our server hardware by more than 90 percent**. But the cost savings will likely be greater due to additional benefits, like the increased compute capacity we can get using AWS. That lets us continually add more and larger workloads than we could using a traditional data center infrastructure, and achieve savings by only paying for what we use."

Delaware North moved almost all of its applications to AWS, including enterprise software such as its Fiorano middleware, Crystal Reports and QLIK business intelligence

solutions, its Citrix virtual desktop system, and Microsoft System Center Configuration Manager, which is used to manage workstations.

The most dramatic physical change was the **elimination of 205 servers**; everything running on that hardware was migrated to AWS. The IT department decided to keep about 20 servers on-premises at the new headquarters building to run communications and file-and-print tasks. “We erred on the side of caution to ensure there is no latency with these tasks, but once we reach a certain comfort level, we may move these to the cloud as well,” Mercer says.

Case Study: Total Cost of Ownership



© 2017 Amazon Web Services, Inc. or its affiliates. All rights reserved.



Academy

Cost Comparison:

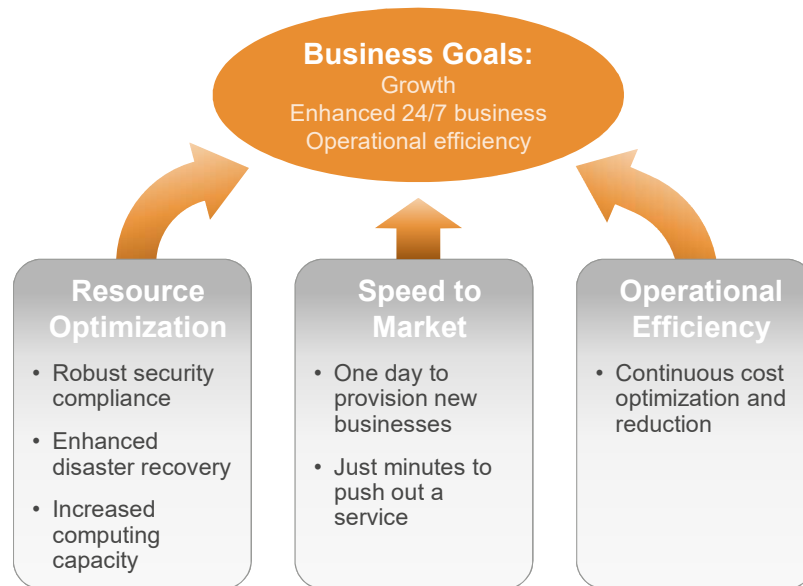
A cost comparison done by Delaware North showed that it could save \$3.5 million based on a five-year run rate by moving its on-premises data center to AWS and using three-year Amazon EC2 Reserved Instances (RI) and Reserved Instance renewals.

Case Study: Total Cost of Ownership



Delaware
North

Results:



Results:

Six months into its cloud migration, Delaware North was realizing benefits in addition to its data center consolidation, including cost-effective security compliance, enhanced disaster recovery, and faster deployment times for new services.

“Robust security in a retail environment is critical for us because of our many retail operations, and AWS is enormously helpful for that,” says Brian Mercer, the senior software architect for the project. “By leveraging the security best practices of AWS, we’ve been able to eliminate a lot of compliance tasks that in the past took up valuable time and money.”

He adds that the company also has increased its disaster recovery capabilities at a lower cost than what was available in its previous data center deployment. “It significantly improved our business continuity capabilities, including seamless failovers,” he says.

The solution is also helping Delaware North operate with greater speed and agility. For example, it can bring in new businesses—either through contracts or acquisitions—and get them online much faster than in the past by eliminating the need for traditional IT procurement and provisioning. It used to take between two and three weeks to provision new business units; now it takes one day. The Delaware North IT team is also using AWS to overhaul its operations by eliminating outdated and cumbersome

processes, cleaning up documentation, and leveraging the benefits of running test and development tasks in combination with rapid deployment of services through the cloud.

“Our DevOps team can now spin up the resources to push out a service in just minutes, compared to the weeks it used to take,” says Scott Mercer. “With AWS, we can respond much faster to business needs. And we can start repurposing time and resources to deliver more value and services to our internal teams and to our customers.”

Resources to Get You Started

AWS TCO Calculator

<https://awstcocalculator.com>

AWS Economics Center

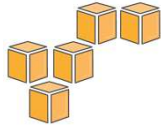
<http://aws.amazon.com/economics/>

Case studies and research



<http://aws.amazon.com/economics/>



Here are some links for you to look at later, try out the simple on-line calculator, and access additional resources.



In review...

-  Cost Optimization
-  Total Cost of Ownership (TCO)

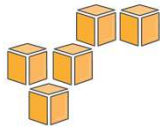


Knowledge Assessment

In review...

- Cost Optimization
- Total Cost of Ownership (TCO)

To complete this module, please remember to finish the corresponding knowledge assessment.



Up Next...

Unit 2: Getting Started with AWS

This completes Unit 1, Introduction to AWS where we have had a high-level overview of cloud computing and AWS.

Up next is Unit 2, Getting Started with AWS where we will take a deeper dive into the AWS infrastructure and components for building your cloud-based compute solution.

© 2017 Amazon Web Services, Inc. or its affiliates. All rights reserved.

This work may not be reproduced or redistributed, in whole or in part, without prior written permission from Amazon Web Services, Inc. Commercial copying, lending, or selling is prohibited.

Errors or corrections? Email us at aws-course-feedback@amazon.com.

For all other questions, contact us at
<https://aws.amazon.com/contact-us/aws-training/>.

All trademarks are the property of their owners.

Do not speak over this slide – just let it play for 8 seconds.