AWS Essentials

12. 설계 방안

Tacademy

CONTENTS

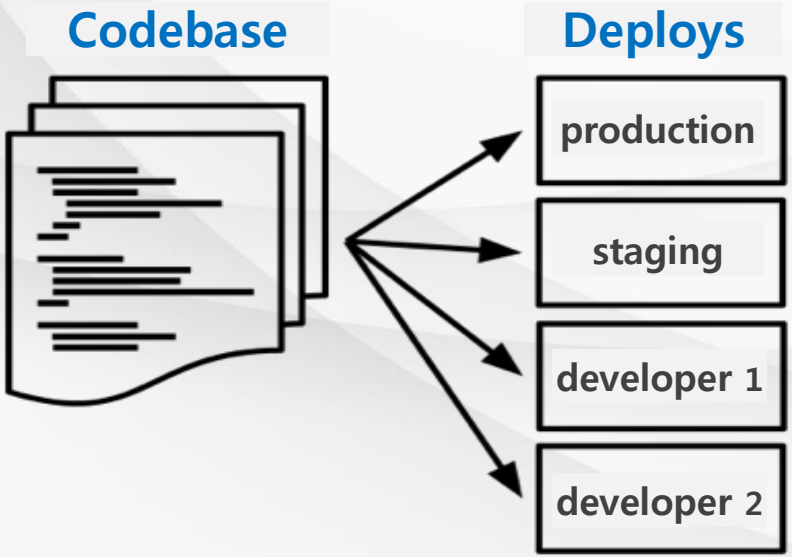
학습 목표

- Cloud Native App의 설계 요소들을 이해할 수 있습니다.
- AWS의 서비스 모범 사례를 통해 효과적인 설계를 이해할 수 있습니다.
- AWS상에서 비용 최적화를 위한 방법을 파악할 수 있습니다.

1. Cloud Native App 설계 요소

코드베이스(Code Base)

- Cloud Native App은 항상 Git, Subversion 등의 **버전 관리 시스템을 사용하여 코드의 버전을 추적**하며 코드베이스와 App 사이는 항상 **1 대 1 관계**가 성립된다.



*참고 12 factors of cloud native app

종속성

- 시스템의 패키지 종속성을 명시적으로 선언하고, 종속선언 Mainifest를 통해 엄격하게 선언한다.
- 누구든 종속성에 따라 간단히 설치 및 개발할 수 있다.

설정

- 각각 배포 환경(Dev, Stage, Prod) 별로 달라지는 모든 것을 포함한다.
- 환경 설정은 서비스 코드와 엄격히 분리시켜 환경 변수를 코드 변경 없이 배포 때마다 쉽게 변경하도록 설계한다.

백엔드 서비스

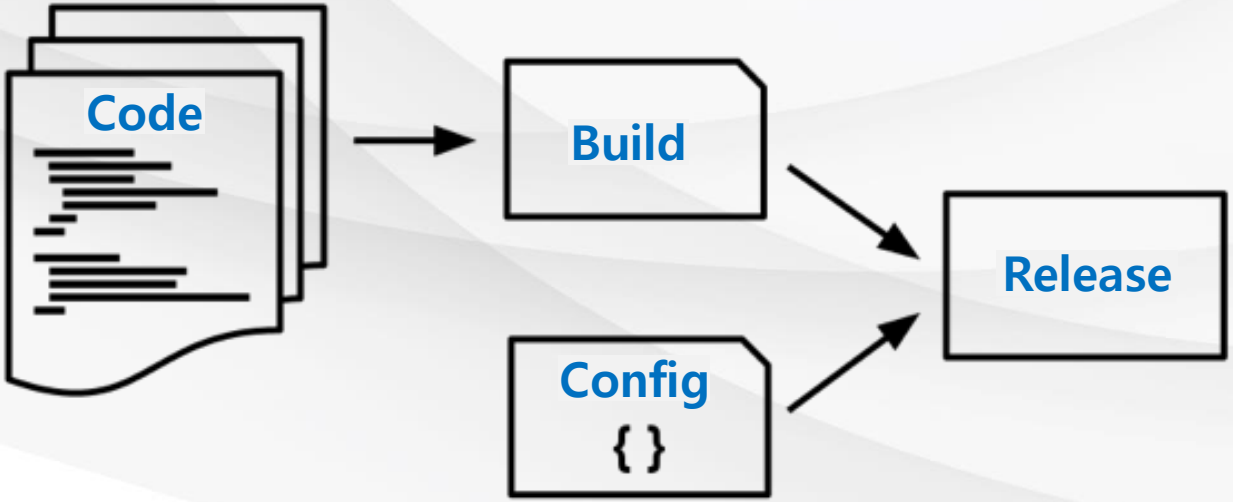
- 데이터베이스와 메시지 큐 같은 백엔드 서비스들을 **서비스에 연결된 리소스로 취급**하며 리소스 간에는 **소결합**을 시킨다.

프로세스

- Cloud Native App의 프로세스는 **무상태(Stateless)**여서 내부적으로 저장하지 않으며, 유지될 필요가 있는 데이터는 **백엔드 서비스에 저장**한다.

빌드, 릴리즈, 실행

- Cloud Native App은 항상 **빌드, 릴리즈, 실행(런타임) 단계를 엄격하게 서비스로 분리**시킨다.
- 실행 단계에서 코드 변경을 할 수 없으며 롤백 시 이전 버전으로 되돌릴 수 있도록 지원한다.



포트 바인딩

- Cloud Native App은 완전히 독립적이며 하나의 App이 다른 App을 위한 백엔드 서버가 될 수 있으므로 **포트를 바인딩하여 서로의 서비스를 공개**한다.

Disposability(일회성)

- 프로세스는 빠르게 시작하거나 종료될 수 있으며 이러한 특징은 신축성 있는 확장과 코드, 설정 변경의 배포를 빠르게 하며 **production 배포를 안정적**이게 한다.

Dev/Prod의 일치

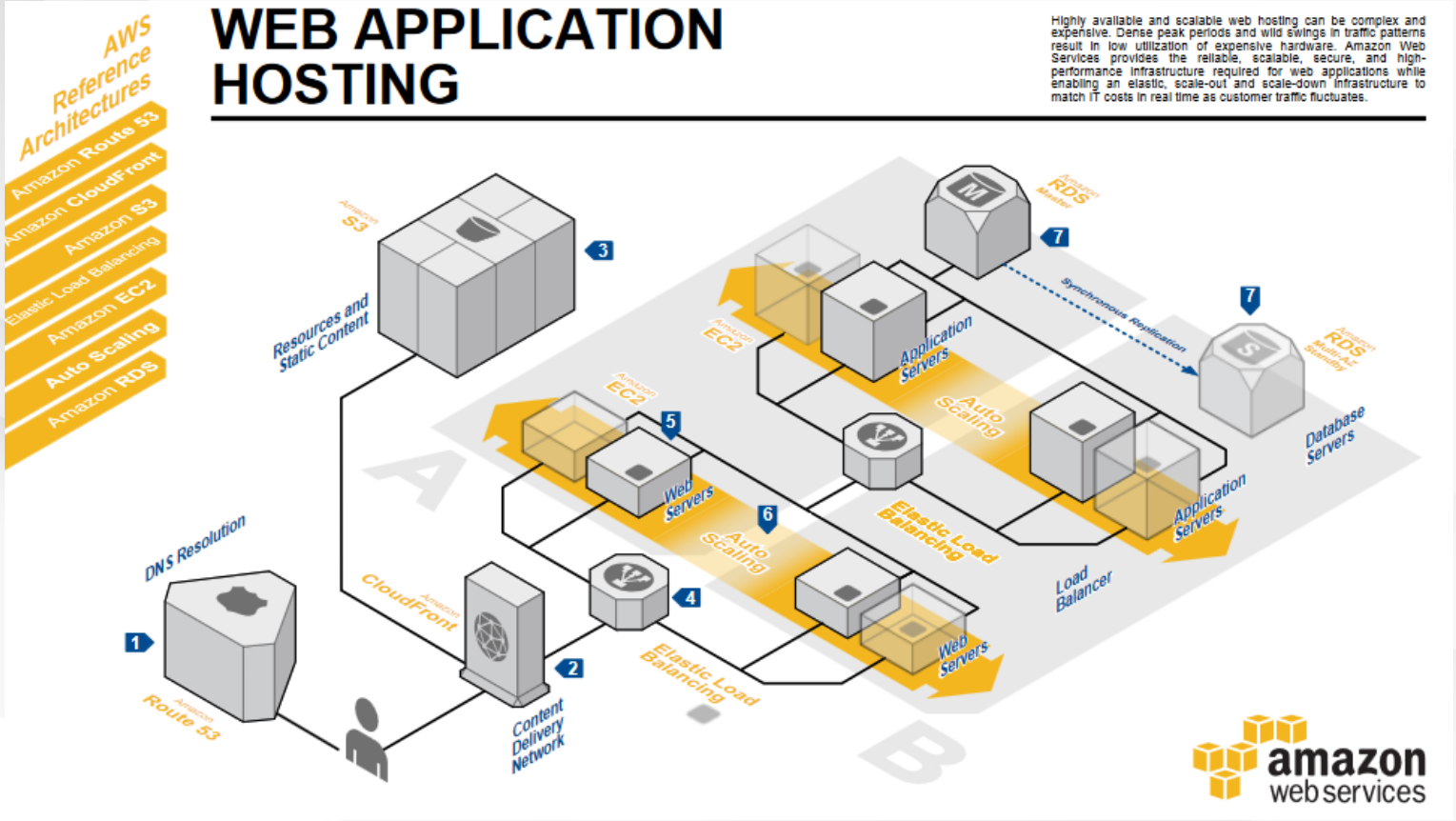
- 개발 환경과 production 환경에서 다른 백엔드 서비스의 설계를 지양한다.
- 같은 종류, 같은 버전의 백엔드 서비스를 이용하여 **환경의 차이를 최소**로 하고 **지속적인 배포**가 가능하도록 디자인한다.

로그

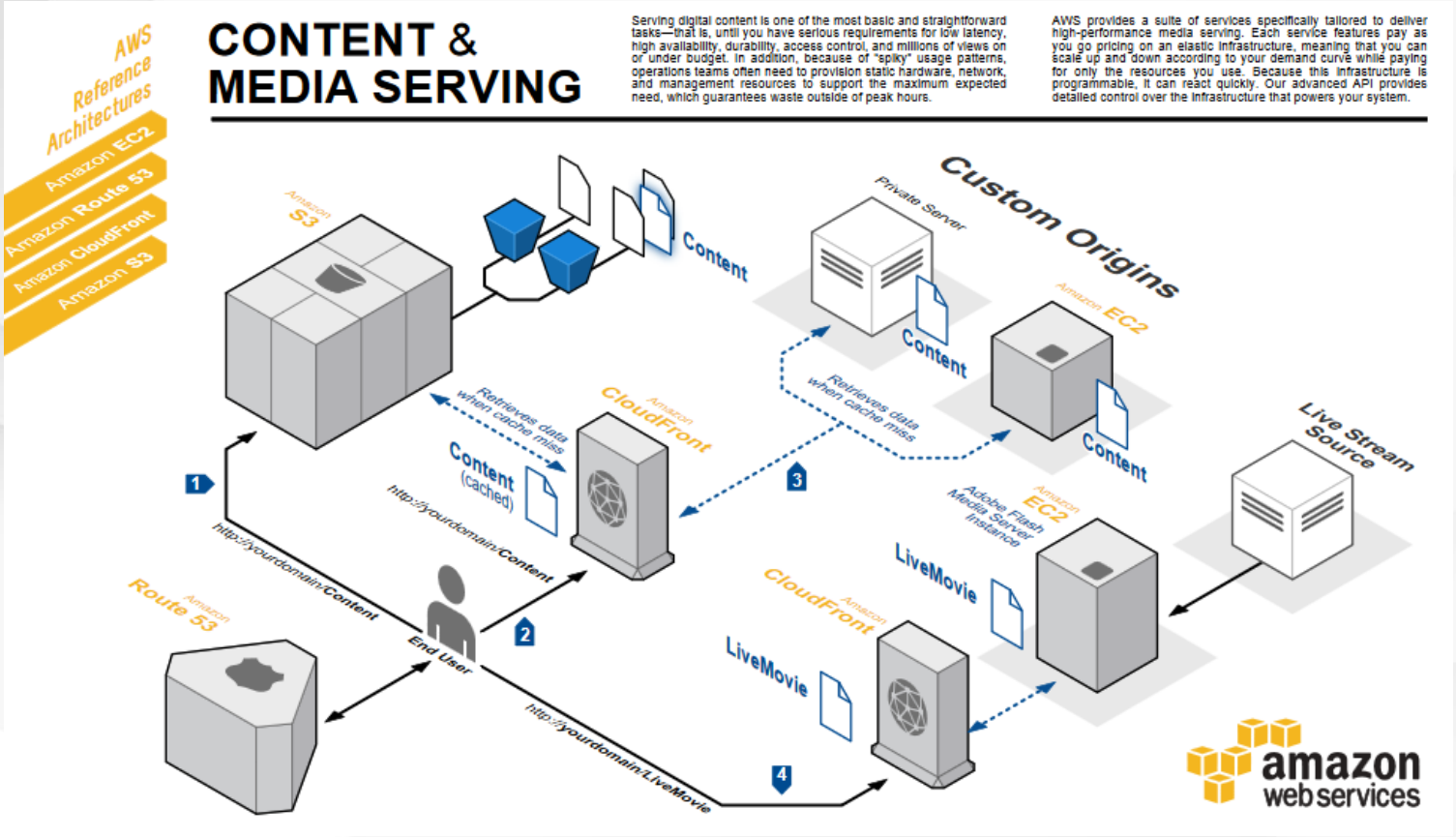
- Cloud Native App의 동작 확인을 위해 로그를 활용하며, 로그 라우터를 사용하여 **로그 분석 시스템 혹은 보관소로 전달**한다.
- 장기간에 걸쳐 App의 **동작 상태를 열람하거나 저장 관리**한다.

2. AWS 모범 설계 사례

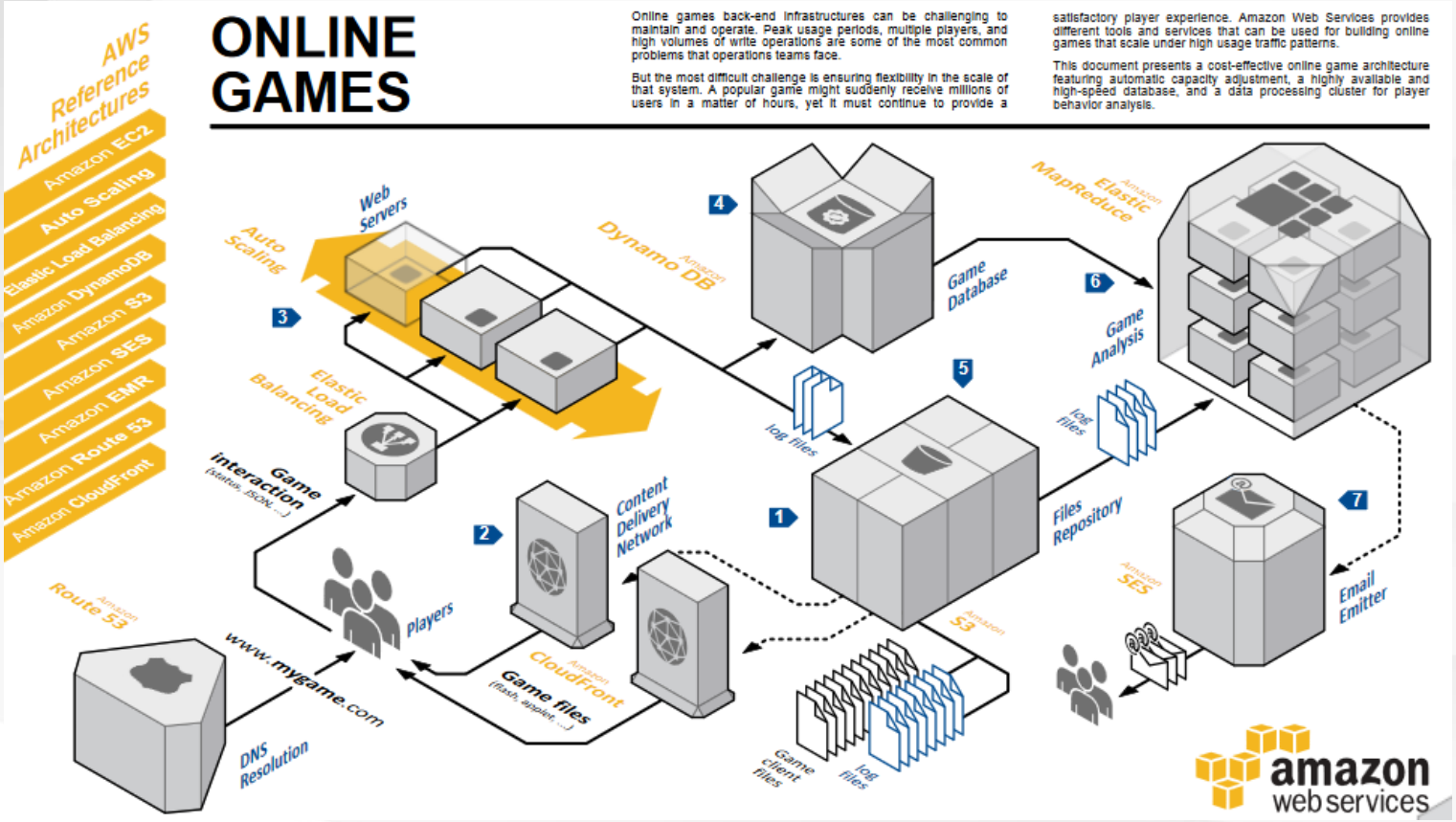
웹 어플리케이션


AWS Reference Architectures
Amazon Route 53
Amazon CloudFront
Amazon S3
Elastic Load Balancing
Amazon EC2
Auto Scaling
Amazon RDS
WEB APPLICATION HOSTING
Highly available and scalable web hosting can be complex and expensive. Dense peak periods and wild swings in traffic patterns result in low utilization of expensive hardware. Amazon Web Services provides the reliable, scalable, secure, and high-performance infrastructure required for web applications while enabling an elastic, scale-out and scale-down infrastructure to match IT costs in real time as customer traffic fluctuates.
Amazon S3
Resources and Static Content
DNS Resolution
Amazon Route 53
Amazon CloudFront
Content Delivery Network
Amazon EC2
Web Servers
Auto Scaling
Elastic Load Balancing
Application Servers
Load Balancer
Amazon RDS Master
Synchronous Replication
Amazon RDS Multi-AZ Standby
Database Servers
amazon web services

미디어 서비스


AWS Reference Architectures
Amazon EC2
Amazon Route 53
Amazon CloudFront
Amazon S3
CONTENT & MEDIA SERVING
Serving digital content is one of the most basic and straightforward tasks—that is, until you have serious requirements for low latency, high availability, durability, access control, and millions of views on or under budget. In addition, because of "spiky" usage patterns, operations teams often need to provision static hardware, network, and management resources to support the maximum expected need, which guarantees waste outside of peak hours.
AWS provides a suite of services specifically tailored to deliver high-performance media serving. Each service features pay as you go pricing on an elastic infrastructure, meaning that you can scale up and down according to your demand curve while paying for only the resources you use. Because this infrastructure is programmable, it can react quickly. Our advanced API provides detailed control over the infrastructure that powers your system.
Custom Origins
Private Server
Content
Amazon S3
Content
Amazon EC2
Content
Retrieves data when cache miss
Amazon CloudFront
Content (cached)
http://yourdomain/Content
Retrieves data when cache miss
Live Stream Source
Adobe Flash Media Server Instance
Amazon EC2
LiveMovie
Amazon CloudFront
LiveMovie
Amazon Route 53
End User
http://yourdomain/Content
http://yourdomain/LiveMovie
1
2
3
4
amazon web services

게임 서비스


AWS Reference Architectures
Amazon EC2
Auto Scaling
Elastic Load Balancing
Amazon DynamoDB
Amazon S3
Amazon SES
Amazon EMR
Amazon Route 53
Amazon CloudFront
ONLINE GAMES
Online games back-end infrastructures can be challenging to maintain and operate. Peak usage periods, multiple players, and high volumes of write operations are some of the most common problems that operations teams face.
But the most difficult challenge is ensuring flexibility in the scale of that system. A popular game might suddenly receive millions of users in a matter of hours, yet it must continue to provide a satisfactory player experience. Amazon Web Services provides different tools and services that can be used for building online games that scale under high usage traffic patterns.
This document presents a cost-effective online game architecture featuring automatic capacity adjustment, a highly available and high-speed database, and a data processing cluster for player behavior analysis.
Web Servers
Auto Scaling
Elastic Load Balancing
Game interaction
(status, JSON, ...)
Amazon Route 53
www.mygame.com
DNS Resolution
players
Content Delivery Network
Amazon CloudFront
Game files
(flash, applet, ...)
Amazon DynamoDB
Game Database
log files
Amazon S3
Game client files
log files
Files Repository
Amazon Elastic MapReduce
Game Analysis
log files
Amazon SES
Email Emitter
1
2
3
4
5
6
7
amazon web services

3. AWS 비용 관리

비용 최적화

불필요한 비용을 제거하고 경제적으로 리소스를 운영하여
비즈니스 목표를 달성하는 요소이다.

- **비용 투명화** : 시스템 비용을 손쉽게 확인하고 서비스별로 부과할 수 있으며 결과적으로 리소스를 최적화하고 비용을 절감한다.
- **소유 비용 절감** : 관리형 서비스를 통해 유지관리 운영 부담을 없애고 비용 효율성을 높일 수 있다.
- **사용량 기반의 운영 비용** : 투자비 없이 사용한 리소스에 대해서만 비용을 지불하여 가격을 절감 시킬 수 있다.
- **규모의 경제** : 직접 구축하고 관리하는 것보다 클라우드에서는 가변비용(Variable cost)을 낮출 수 있으므로 저렴한 비용 운영을 할 수 있다.

컴퓨팅 자원 구매 옵션

- On-Demand : 약정 없이 쓴 만큼만 지불하는 구조이며 비즈니스를 예측하기 어려운 신규 서비스 혹은 중단이 발생하면 안 되는 서비스에 적합하다.

- Reserved : 1년 혹은 3년으로 약정을 하는 비용 구조이며 40~70% 할인을 받는다. 항상 사용 중이며 장기간 활용할 리소스에 적합하다.

- Spot : 남은 자원에 대한 경매 방식으로 온디멘드 대비 80~90% 저렴하다. 단기적으로 수요가 많은 서비스에 적합하다.

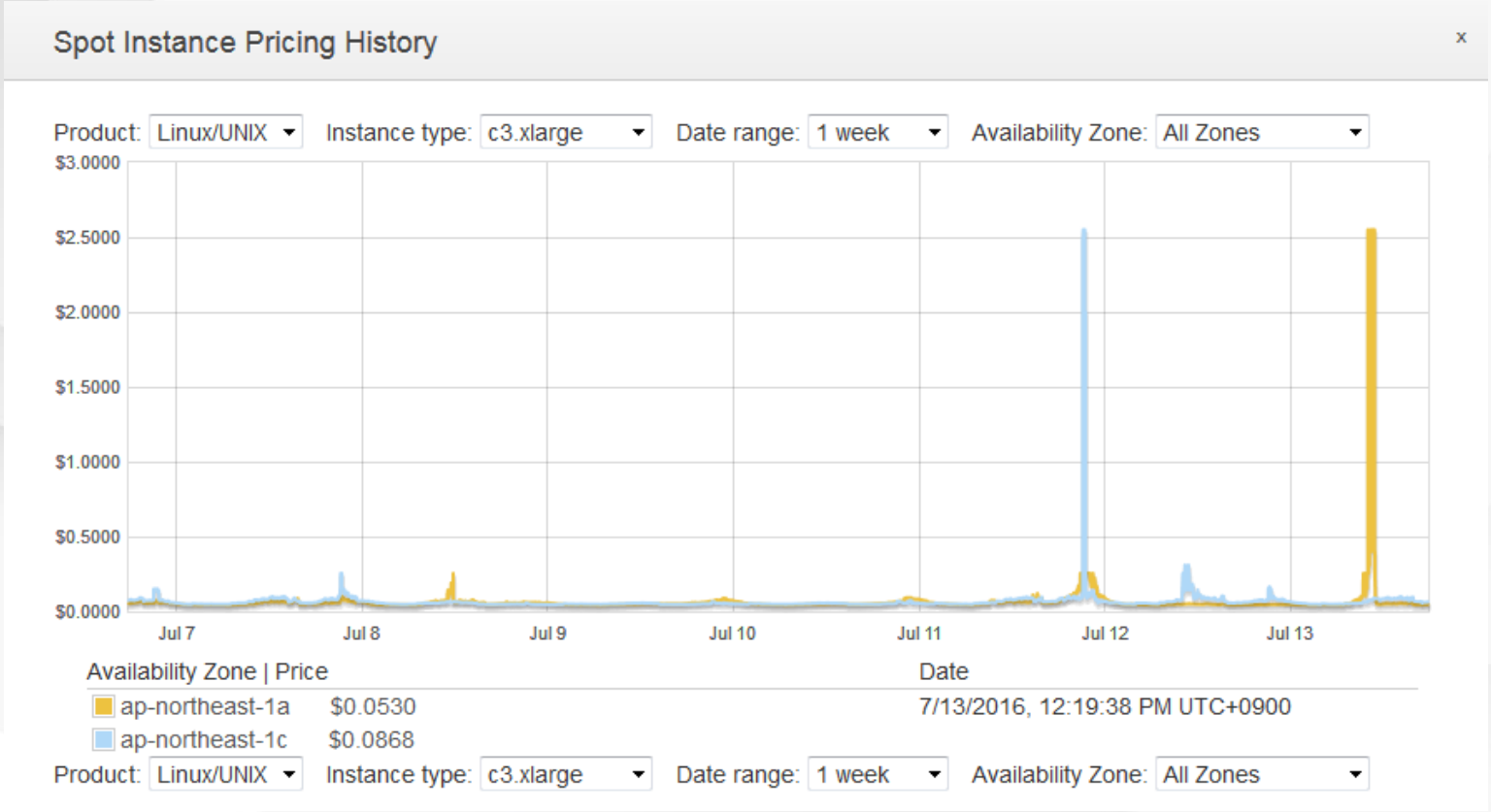
- Dedicated : 특정 고객을 위한 물리적인 리소스 자원이며 규제 및 법적인 문제가 민감한 시스템에 적합하다.

RI 가격 절감

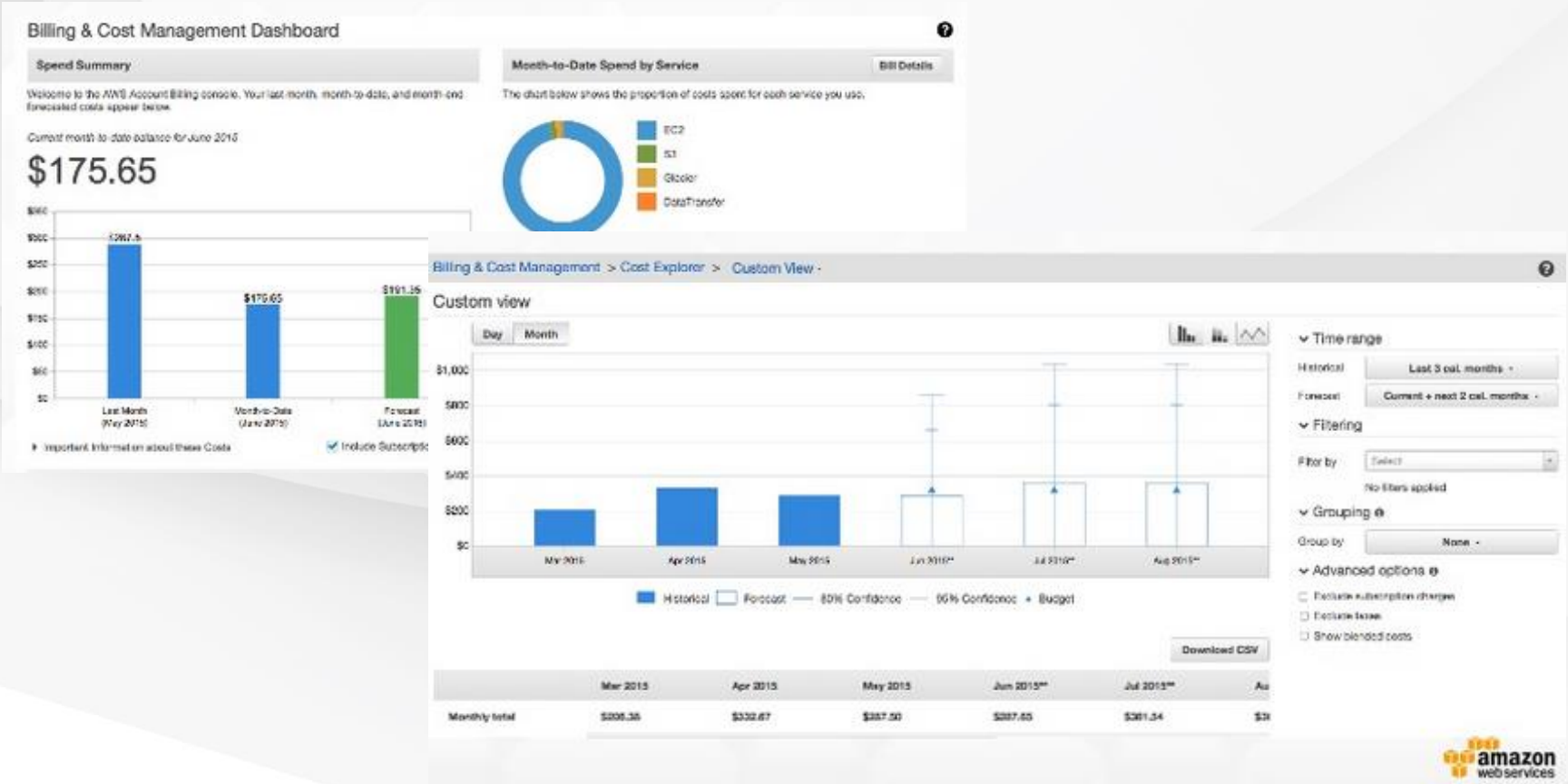
http://calculator.s3.amazonaws.com/index.html

amazon web services

Spot Instance 사례



비용 관리 보드


Billing & Cost Management Dashboard
Spend Summary
Month-to-Date Spend by Service
Bill Details
Welcome to the AWS Account Billing console. Your last month, month-to-date, and month-end forecasted costs appear below.
The chart below shows the proportion of costs spent for each service you use.
Current month-to-date balance for June 2015
$175.65
EC2
S3
Glacier
DataTransfer
$179.65
$191.35
Last Month (May 2015)
Month-to-Date (June 2015)
Forecast (June 2015)
Important Information about these Costs
Billing & Cost Management > Cost Explorer > Custom View -
Custom view
Day
Month
$1,000
$800
$600
$400
$200
$0
Mar 2015
Apr 2015
May 2015
Historical
Forecast
80% Confidence
95% Confidence
Budget
Download CSV
Time range
Historical
Last 3 cal. months
Forecast
Current + next 2 cal. months
Filtering
Filter by
No filters applied
Grouping
Group by
None
Advanced options
Exclude subscription charges
Exclude taxes
Show blended costs
Mar 2015
Apr 2015
May 2015
Monthly total
$206.35
$332.67
$357.50
$357.65
$361.34

학습정리

지금까지 [설계 방안]에 대해서 살펴보았습니다.

Cloud Native app의 설계 요소

Cloud Native app을 설계할 때 코드베이스, 종속성, 설정, 백엔드 서비스, 배포, 릴리즈, 실행, 프로세스, 포트바인딩, 일회성, dev/prod 일치, 로그의 요소들을 적용시켜 서비스의 수준을 높인다.

AWS 모범 설계 사례

웹 어플리케이션, 미디어 서비스, 온라인 게임 서비스의 기본적인 AWS 서비스를 활용한 설계의 지침을 통해 기존의 개념을 보강하고 탄력성과 확장성 같은 클라우드 특징을 충분히 활용하여 비즈니스의 가치를 높인다.

AWS 비용 관리

빌링 Console을 통한 비용 분석 및 다양한 컴퓨트 자원의 과금 모델들을 활용하여 효율적으로 비용을 관리한다.