



AWS 실습

- AWS Lambda

Sung-Dong Kim,
School of Computer Engineering,
Hansung University

서버 없는 “hello world” 실행 (1)

- 1단계: Lambda 콘솔로 이동
- 2단계: 함수 생성 → Blue print 사용
 - 블루프린트: hello-world-python 검색
 - [구성]

서버 없는 “hello world” 실행 (2)

- 3단계: Lambda 함수 구성과 생성
 - 기본 정보
 - 함수 이름
 - 실행 역할: create new role from template(s)
 - 역할 이름: lambda_basic_execution
 - [함수 생성]
 - Lambda 함수 핸들러: 이벤트 처리 함수, AWS Lambda는 이벤트 데이터를 핸들러에게 전달

서버 없는 “hello world” 실행 (3)

- 3단계: Lambda 함수 구성과 생성
 - runtime: Python (Java, Node.js, C#, Go, python 등 가능)
 - 핸들러
 - 이벤트 처리 함수
 - AWS Lambda가 코드 실행을 시작하는 곳
 - AWS Lambda는 이벤트 데이터를 핸들러에게 전달
 - 메모리, 타임아웃, VPC: 그대로 둬

서버 없는 “hello world” 실행 (4)

- 4단계: Lambda 함수 호출 및 결과 확인
 - [테스트 이벤트 구성]
 - 이벤트 템플릿: Hello World
 - 이벤트 이름: HellowWorldEvent
 - key1의 value1을 “hello world”로 변경
 - [생성]
 - [테스트] → 실행 결과의 “세부 정보”

서버 없는 “hello world” 실행 (5)

- 5단계: 지표 감시
 - 자동으로 Lambda function 모니터
 - CloudWatch를 통해 지표 보고
 - # of requests, latency per request, ...
 - [test] 버튼을 여러번 클릭 → “Monitoring” 탭

Configuration

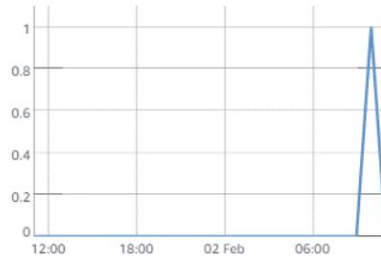
Monitoring

CloudWatch metrics at a glance (aggregated per hour)

[View traces in X-Ray](#)

Invocation count

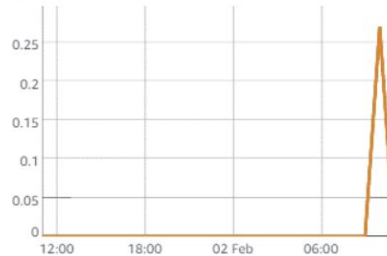
Last 24 hours ▼

[Jump to Metrics](#)[Jump to Logs](#)

Count

Invocation duration

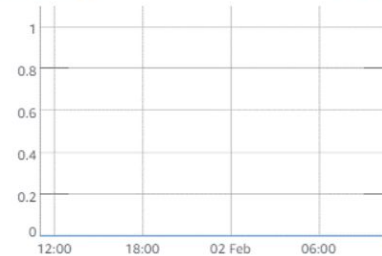
Last 24 hours ▼

[Jump to Metrics](#)[Jump to Logs](#)

Max Milliseconds
Avg Milliseconds
Min Milliseconds

Invocation errors

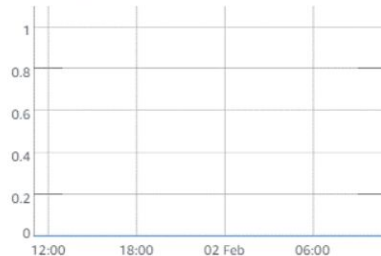
Last 24 hours ▼

[Jump to Metrics](#)[Jump to Logs](#)

Count

Throttled invocations

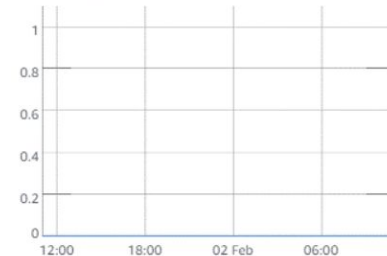
Last 24 hours ▼

[Jump to Metrics](#)[Jump to Logs](#)

Count

Iterator age

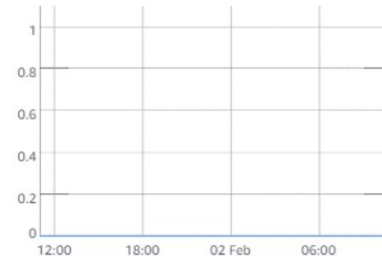
Last 24 hours ▼

[Jump to Metrics](#)[Jump to Logs](#)

Max Milliseconds

DLQ errors

Last 24 hours ▼

[Jump to Metrics](#)[Jump to Logs](#)

Count

삭제

- 함수 삭제
- 로그 그룹 삭제 - CloudWatch -> 로그 -> 로그 그룹
- 역할 삭제 - IAM -> 액세스 관리 -> 역할

source

- <https://aws.amazon.com/getting-started/tutorials/run-serverless-code/>
- <https://aws.amazon.com/ko/getting-started/tutorials/run-serverless-code/>

What to do

- palindrome을 검사하는 isPalindrome 함수 생성
 - runtime: Node.js
- API gateway 생성
 - 함수를 trigger → HTTP request를 이용하여 string 인자 전달
 - REST API를 구성

Lambda 함수 만들기 (1)

- Lambda \rightarrow 함수 \rightarrow 함수 생성
- 새로 작성

함수 이름

함수의 용도를 설명하는 ()

isPalindrome

공백 없이 문자, 숫자, 하이픈

런타임 정보

함수를 작성하는 데 사용할

Node.js 16.x

아키텍처 정보

함수 코드에 대해 원하는

☒ x86_64


☐ arm64

▼ 기본 실행 역할 변경

실행 역할

함수에 대한 권한을 정의하는 역할을 선택합니다. 새

- ☐ 기본 Lambda 권한을 가진 새 역할 생성
- ☐ 기존 역할 사용
- ☒ AWS 정책 템플릿에서 새 역할 생성

 역할을 생성하는 데 몇 분 정도 걸릴 수

역할 이름

새 역할의 이름을 입력합니다.

execute_palindrome

공백 없이 문자, 숫자, 하이픈 또는 밑줄만 사용합니다.

정책 템플릿 - 선택 사항 정보

정책 템플릿을 하나 이상 선택합니다.

단순 마이크로서비스 권한 ✕
DynamoDB

Lambda 함수 만들기 (2)

- 함수 코드 작성
 - Lambda function은 module로 export 하는 함수 = handler
 - 다음 페이지 코드 작성 → [save] → [Deploy]

```
exports.handler = async (event, context, callback) => {  
  const string = 'abcd';  
  const reverse = string.split("").reverse().join("");  
  const isPalindrome = (string == reverse);  
  const result = isPalindrome ? `${string} is a palindrome` : `${string}  
is not a palindrome`;  
  
  return result;  
};
```

Lambda 함수 만들기 (3)

- 테스트 → 테스트 이벤트 구성
 - 이벤트 템플릿: Hello World
 - 이벤트 이름: palindrome-test
 - [생성]
 - [테스트]

Tools Window

Test

Deploy



index.js



Execution result: X



▼ Execution results

Status:

Test Event Name

palindrome-test

Response

"abcd is not a palindrome"

Function Logs

START RequestId: 5825c1f6-92e3-4a6d-b091-c734f6cc64a6 Vers:

END RequestId: 5825c1f6-92e3-4a6d-b091-c734f6cc64a6

REPORT RequestId: 5825c1f6-92e3-4a6d-b091-c734f6cc64a6 Du

Request ID

5825c1f6-92e3-4a6d-b091-c734f6cc64a6

Lambda 함수 만들기 (4)

- 함수를 테스트 이벤트에 반응하도록 수정
 - key1의 value1, value2, value3 변경
 - [save]
 - [test]

```
exports.handler = async (event, context, callback) => {  
  let isPalindrome, rev, key, result="";  
  for (key in event) {  
    rev = event[key].split("").reverse().join("");  
    isPalindrome = (event[key] == rev);  
    result += isPalindrome ? `${event[key]} is a Palindrome : ` :  
    `${event[key]} is not a Palindrome : `;  
  }  
  callback(null, result);  
};
```

API Gateway 만들기 (1)

- API Gateway → API 생성

프로토콜 선택

REST API 또는 WebSocket API의 생성 여부를 선택합니다.

☒ REST ☐ WebSocket

새 API 생성

Amazon API Gateway에서 API는 HTTPS 엔드포인트를 통해 호출할 수 있는 리소스와 메서드의 컬렉션을 참조합니다.

☒ 새 API ☐ 기존 API에서 복제 ☐ Swagger에서 가져오기 ☐ 예제 API

이름 및 설명

API에 대해 기억하기 쉬운 이름과 설명을 선택합니다.

API 이름*	<input type="text" value="palindromeAPI2"/>
설명	<input type="text"/>
엔드포인트 유형	<div>지역 </div>

API Gateway 만들기 (2)

- 리소스 → 작업 → 메소드 생성 → GET → v (check)

!/- GET - 설정

새 메서드의 통합 포인트를 선택합니다.

통합 유형 ☒ Lambda 함수 ⓘ

☐ HTTP ⓘ

☐ Mock ⓘ

☐ AWS 서비스 ⓘ

☐ VPC 링크 ⓘ

Lambda 프록시 통합 사용 ☐ ⓘ

Lambda 리전

Lambda 함수

isPalindrome2

기본 제한 시간 사용 ☒ ⓘ

API Gateway 만들기 (3)

- 통합 요청 → 매핑 템플릿
 - 통합 유형: Lambda 함수
 - 정의된 템플릿이 없는 경우 (권장)
 - 매핑 템플릿 추가: application/json → v (check)
 - 템플릿 생성: JSON 형식
 - 저장

```
{  
  "string" :  
  "$input.params('string')"  
}
```

API Gateway 만들기 (4)

- API 배포 (deploy)
 - [작업] → API 배포

API 배포

API를 배포할 스테이지를 선택합니다. 예를 들어, API 테스트 버전이라면 베타라는 이름의 단계에 배포할 수 있습니다.

배포 스테이지

[새 스테이지]

스테이지 이름*

prod

스테이지 설명

배포 설명

취소

배포

API Gateway 만들기 (5)

- URL 호출: Lambda와 HTTP 통신
- 함수 수정: 다음 페이지
- 브라우저에 다음을 입력하여 테스트:

URL?*string=someValue*

```
exports.handler = async (event, context, callback) => {  
  const string = event.string;  
  const reverse = string.split("").reverse().join("");  
  const isPalindrome = (string == reverse);  
  const result = isPalindrome ? `${string} is a palindrome` :  
    `${string} is not a palindrome`;  
  
  callback(null, result);  
};
```


source



- <https://medium.com/free-code-camp/going-serverless-how-to-run-your-first-aws-lambda-function-in-the-cloud-d866a9b51536>



Azure Introduction

Sung-Dong Kim,
School of Computer Engineering,
Hansung University

Azure란? (1)

- cloud computing platform by Microsoft, Feb. 2010
- 완전한 클라우드 컴퓨팅 플랫폼: 응용 프로그램 개발, 테스트, 배포, 관리를 위한 클라우드 서비스를 통합
- 좋아하는 도구, 프레임워크를 사용하여 글로벌 네트워크에서 application을 자유롭게 빌드, 관리, 배포할 수 있음
- <https://www.microsoft.com/ko-kr/videoplayer/embed/RWlZ>

Q6

Azure란? (2)

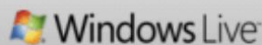
- 기존 응용 프로그램을 호스트
- 새 응용 프로그램 개발 간소화
- on-premise 응용 프로그램 향상
- AI 및 기계 학습 서비스 제공 - 시각, 청각, 음성
- 저장소 솔루션 제공
- development, data storage, service hosting, service management, ...



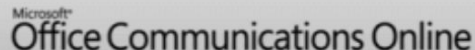
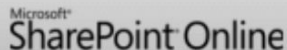
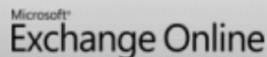
Microsoft Azure

Global • Trusted • Hybrid

Application Services



Software Services



Platform Services



Infrastructure Services

Azure로 무엇을 할 수 있나?

- 가상 머신, 기존 애플리케이션 실행, 지능형 봇, 합성 현실,... → 100여 개의 서비스
- AI, 기계학습 서비스
- 동적으로 확장되는 storage solution



Microsoft Azure

관리 / 모니터링 (Management & Monitoring)

관리 (Administration) Azure Portal	모니터링 (Monitoring) 로그 분석	스케줄링 (Job Scheduling) 공정관리자 (Scheduler)	배포관리 (Deployment Orchestration) 자동화 (Automation)	CLI (Command Line Interface) Azure PowerShell Azure CLI (Windows, Mac, Linux)
--	-----------------------------------	---	--	--

인지 서비스 (Cognitive Services)

언어 (Language)	영상 (Vision)	음성 (Speech)	지식 (Knowledge)
언어인식	얼굴인식	음성전달	추천
텍스트인식	감정인식	발표자인식	검색

보안 / 신원 (Security & Identity)

디렉토리 (Directory Service) Azure Active Directory	보안 (Security) 보안 센터	암호화 (Encryption) 주요 자격 증명 모음	2 팩터 인증 (Multi-Factor Authentication) Multi-Factor Authentication	개발 도구 (Developer Tools) Visual Studio Team Services
---	-------------------------------	--	---	---

분석 / IoT (Analytics & IoT)

분석 (Analytics) 스트림 분석 데이터 레이크	빅데이터 (Big Data) HDInsight	IoT (Internet of Things) IoT Hub
머신러닝 (Machine Learning) 기계학습	시각화 (Visualization) Power BI Embedded	데이터스트리밍/관리 이벤트 허브 데이터 팩토리

어플리케이션 서비스 (Application Service)

웹 앱 (Web Applications) 웹 앱 클라우드서비스	모바일 (Mobile Applications) 모바일 앱	모바일 분석 (Mobile Analytics) Mobile Engagement	앱 테스트 (App Testing) DevTest Labs	앱 스트리밍 (App Streaming) Remote App
메시징 (Messaging) 큐 서비스버스	알림 (Notification) 알림허브	워크플로우 (Workflow) Logic Apps	API 관리 (API Management) API Management API 앱	백그라운드 프로세스 (Backend Process Logic) Functions

미디어 & CDN

CDN (Contents Delivery Network) CDN	미디어 (Transcoding) 미디어서비스
---	------------------------------------

데이터 / 데이터베이스 (Data as a Service)

RDB (Relational Database) SQL Database	NoSQL (NoSQL database) CosmosDB	DW (Data Warehouse) SQL Data Warehouse	캐쉬 (Caching) Redis Cache	검색 (Managed Search) Search	테이블저장소 (Table storage) 테이블저장소
--	---	--	------------------------------------	--------------------------------------	---

컴퓨팅 (Compute)

VM (Virtual Machine) 가상컴퓨터	가변 VM (Auto Scale) VM Scale Set	컨테이너 (Container) 컨테이너
마이크로서비스 (Microservice-based Application) 서비스 패브릭	병렬/배치작업 (Job-Based Application) 배치	재해복구 (Disaster Recovery) Site Recovery

네트워크 (Networking)

VPN (Virtual Private Network) 가상네트워크 VPN 게이트웨이	전용망 (Dedicated Network) Express 경로
부하분산 (Load Balancing) 부하분산장치 트래픽관리자	DNS (Domain Name Service) DNS

저장소 (Storage)

파일 저장소 (File Storage) File	오브젝트 저장소 (Object Storage) Blob	Disk (Virtual Server Disk Infra) Standard Disk Premium Disk
아카이빙/백업 (Archiving & Backup) 백업	하이브리드저장소 (Hybrid Storage) 스토어싱플	데이터 이관 (Data Transport) Import/Export

Core Azure Infrastructure Services



Compute

Virtual machines

Availability sets

VM Scale Sets



Storage

Disks
(Standard, premium)

Blob storage
(Hot, cool)

Files



Networking

Virtual networks

VPN, ExpressRoute

Load Balancer

DNS, Traffic Manager

Application 호스팅 (1)

- 인프라 세부 정보에 대한 걱정 없이 application을 실행할 수 있는 다양한 클라우드 기반 컴퓨팅 제공 사항을 지원
- 응용 프로그램 호스트 방법 결정
 - 전체 인프라를 VM으로 관리
 - Azure에서 제공하는 플랫폼 관리 기능 사용
 - 코드 실행만 호스트하는 서버를 사용하지 않는 프레임워크

Application 호스팅 (2)

- IaaS
 - 호스팅에 대한 모든 권한 제공
 - 코드가 실행되는 VM을 자세히 제어할 경우
- PaaS - 앱을 지원하는 데 필요한 완전히 관리되는 서비스 제공
 - App Service: 대부분의 웹 사이트, 웹 응용프로그램 호스팅
 - Service Fabric: 마이크로 서비스 아키텍처의 경우

서버를 사용하지 않는 호스팅- 코드만 작성: **Functions**

Virtual Machines/
Virtual Networks



Service Fabric



App Service



Functions



Customer-managed
(IaaS)

Platform-managed
(PaaS)

Code-only
(serverless)

Azure Virtual Machines (1)

- Windows / Linux VM 배포 지원
- 사용자가 컴퓨터 구성을 완전히 제어
- 사용자가 모든 서버 SW 설치, 구성, 유지 관리, OS 패치 담당

Azure Virtual Machines (2)

- VM 특징
 - size
 - region
 - network: 가상 네트워크, 네트워크 보안 그룹
 - resource groups: 논리적 컨테이너

Azure Virtual Machines (3)

- 언제?
 - application 인프라를 완전히 제어할 때
 - on-premise application workload를 변경하지 않고 Azure에 마이그레이션 할 때

Azure App Service (1)

- 웹 기반 프로젝트를 가장 빠른 경로로 게시할 때 이용
- 인프라를 관리할 필요 없이 선택한 프로그래밍 언어로 웹 응용 프로그램을 빌드하고 호스팅 할 수 있음
- 웹앱, 모바일 앱 백엔드, API 앱 개발
- DevOps를 염두에 두고 설계됨: GitHub, Jenkins, Azure DevOps 등 게시 및 연속 통합 배포를 위한 다양한 도구 지원

Azure App Service (2)

- 웹 응용 프로그램, REST API 및 모바일 백 엔드를 호스팅 하는 서비스
- .NET, .NET Core, Java, Ruby, Node.js, PHP, Python
- 보안, 부하 분산, 자동 크기 조정 등 Azure의 기능을 응용 프로그램에 추가

Azure App Service (3)

- 언제?
 - 기존 웹 애플리케이션을 Azure로 마이그레이션 할 때
 - 앱에 대해 완전히 관리되는 호스팅 플랫폼이 필요할 때
 - 앱에서 모바일 클라이언트를 지원할 때
- source -

<https://docs.microsoft.com/ko-kr/azure/app-service/>

Azure Functions (1)

- 코드를 실행하기 위해 전체 프로그램 또는 인프라를 빌드, 관리하지 않아도 됨
- 이벤트 or 일정에 대한 응답으로 실행할 때
- 서버를 사용하지 않는 스타일 제공 → 필요한 코드만 작성
- HTTP 요청, Webhook, 클라우드 서비스 event 또는 일정에 따라 코드 실행이 트리거 됨

C#, F#, Node.js, Python, PHP, ...

Azure Functions (2)

- 언제?
 - 웹 기반 이벤트 또는 일정에 따라 트리거되는 코드가 있을 때
 - 완전히 호스트된 프로젝트의 오버헤드가 필요하지 않을 때

Azure Service Fabric

- 마이크로 서비스를 관리하는 분산된 시스템 플랫폼
- 배포된 응용 프로그램의 프로비전, 배포, 모니터링, 업그레이드/패치 및 삭제 등 관리 기능 제공
- 언제?
 - 응용 프로그램 개발
 - 마이크로 서비스 아키텍처를 사용하도록 기존 애플리케이션을 다시 작성

인증

- 앱 클라이언트 인증을 위한 여러 가지 방법 제공
- Azure AD (Active Directory): MS 다중 테넌트, 클라우드 기반 ID 및 액세스 관리 서비스 → SSO (Single Sign-On) 추가
- App Service 인증: Azure AD, social ID 공급자 모두를 사용하여 인증 사용할 때

모니터링

- Visual Studio Application Insights: 웹 응용 프로그램 모니터링
- Azure Monitor: Azure 인프라, 리소스 모니터링

Azure 지역

- 전 세계 여러 데이터 센터에 application을 배포할 수 있음
- 다중 지역 앱
 - 최대 가용성 제공
 - 둘 이상의 데이터 센터에 application을 호스팅
 - Azure Traffic Manager: 다중 지역 지원

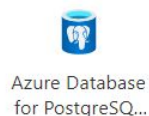
응용 프로그램 및 프로젝트 관리 방법

- Azure Portal: portal.azure.com
- Azure CLI
- Azure PowerShell
- REST API: Azure 리소스와 응용 프로그램을 프로그래밍 방식으로 프로비저닝, 관리
- API: Azure SDK를 이용해 응용 프로그램에서 리소스를 프로그래밍 방식으로 관리

Azure Portal

- 웹 기반 통합 콘솔
- 간단한 웹앱, 복잡한 클라우드 배포 등 모든 것을 구축, 관리, 모니터링
- 사용자 지정 대시보드 - 리소스 관리
- 필요한 옵션을 구성하여 최적의 환경 구축

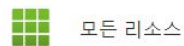
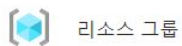
Azure 서비스



최신 리소스

이름	형식	마지막으로 본 날짜
KVision	리소스 그룹	4시간 전
종량제	구독	9개월 전

탐색



Azure Marketplace



- Azure에서 실행되도록 최적화된 솔루션과 서비스의 검색, 체험, 구매, 프로비전 제공

Microsoft Azure

Create a resource

Dashboard

All services

FAVORITES

Resource groups

All resources

Recent

App Services

SQL databases

Virtual machines (classic)

Virtual machines

Cloud services (classic)

Subscriptions

Azure Active Directory

Monitor

Security Center

Cost Management + Billing

Help + support

Advisor

Dashboard > Marketplace > Everything

Marketplace

My Saved List 1

Everything

Compute

Networking

Storage

Web

Mobile

Containers

Databases

Analytics

AI + Machine Learning

Internet of Things

Integration

Security

Identity

Developer tools

Management Tools

Software as a service (SaaS)

Blockchain

Search resources, services, and docs

Everything

Search Everything

Pricing

All


Operating System

All

Publisher


All

What's new




Azure Data Box

Microsoft




Red Hat Enterprise Linux

Red Hat




Azure Databricks

Microsoft




Citrix Virtual Apps Essentials

Citrix



Barracuda WAF-as-a-Service

Barracuda Networks, I




Confidential Compute VM

Microsoft Azure Com


More

Users interested in Template deployment also viewed




Storage account - blob, file, table, queue

Microsoft



Template deployment


Microsoft



CentOS-based 7.5


Rogue Wave Software (formerly OpenLogic)

Recommended for you




SQL Database

Microsoft




Ubuntu Server 16.04 LTS

Canonical




Azure Cosmos DB

Microsoft




Container Service

Microsoft



Application Insights

Microsoft



Azure Database for MySQL

Microsoft

Web

More

Source



- <https://docs.microsoft.com/ko-kr/azure/guides/developer/azure-developer-guide>
- <https://docs.microsoft.com/ko-kr/learn/azure/>



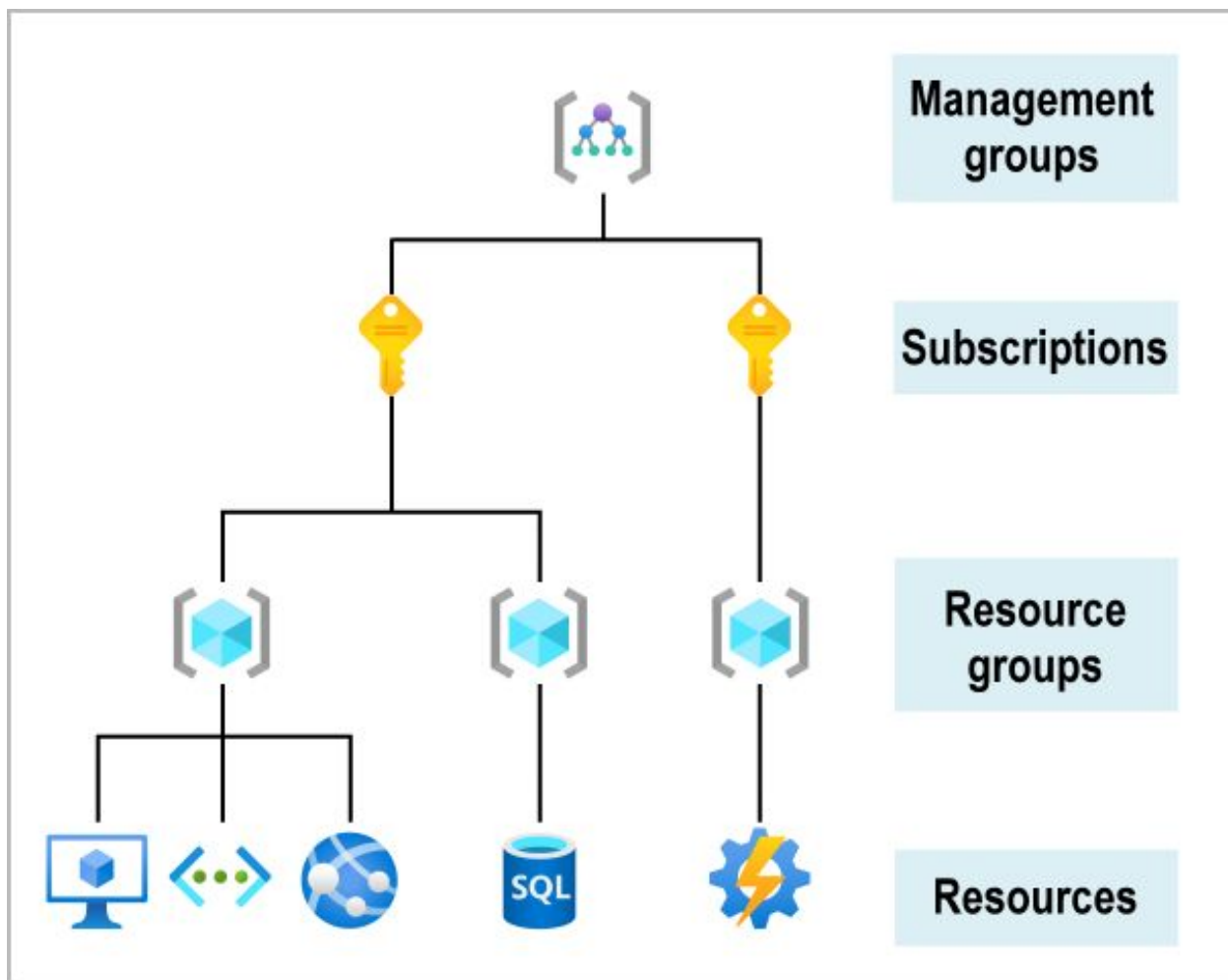
Azure

- Getting Started

Sung-Dong Kim,
School of Computer Engineering,
Hansung University

Azure 구독, 관리 그룹, 리소스 및 지역

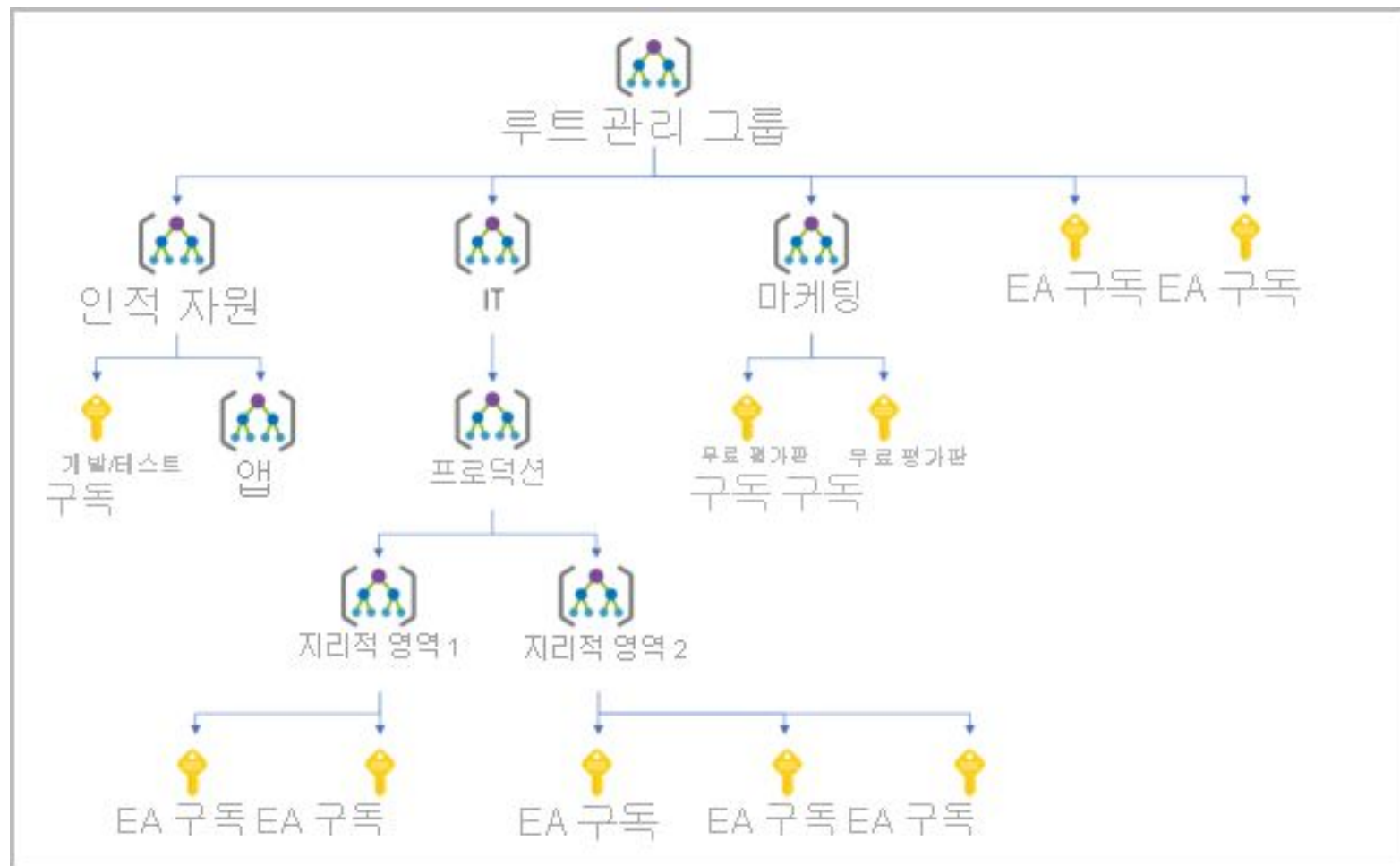
- Azure의 리소스에 대한 구조: management groups - subscriptions - resource groups - resources
- resource: 가상 머신, 스토리지, SQL database, ... → 사용자가 만든 서비스 instance
- resource group: resource가 배포, 관리되는 논리적 컨테이너
- subscription: 만들고 사용가능한 리소스 양에 대한 제한/할당량
→ 팀/프로젝트별 비용 관리
- 관리 그룹: 여러 구독에 대한 액세스, 정책 등을 관리



Azure 구독

- Azure 제품/서비스에 대한 인증되고 권한이 부여된 액세스 제공
- 리소스 프로비저닝
- Azure AD(active directory)에 있는 ID인 Azure 계정과 연결된 Azure 서비스의 논리적 단위

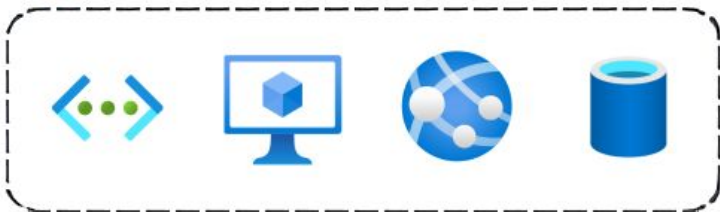




Azure 리소스 그룹 (1)

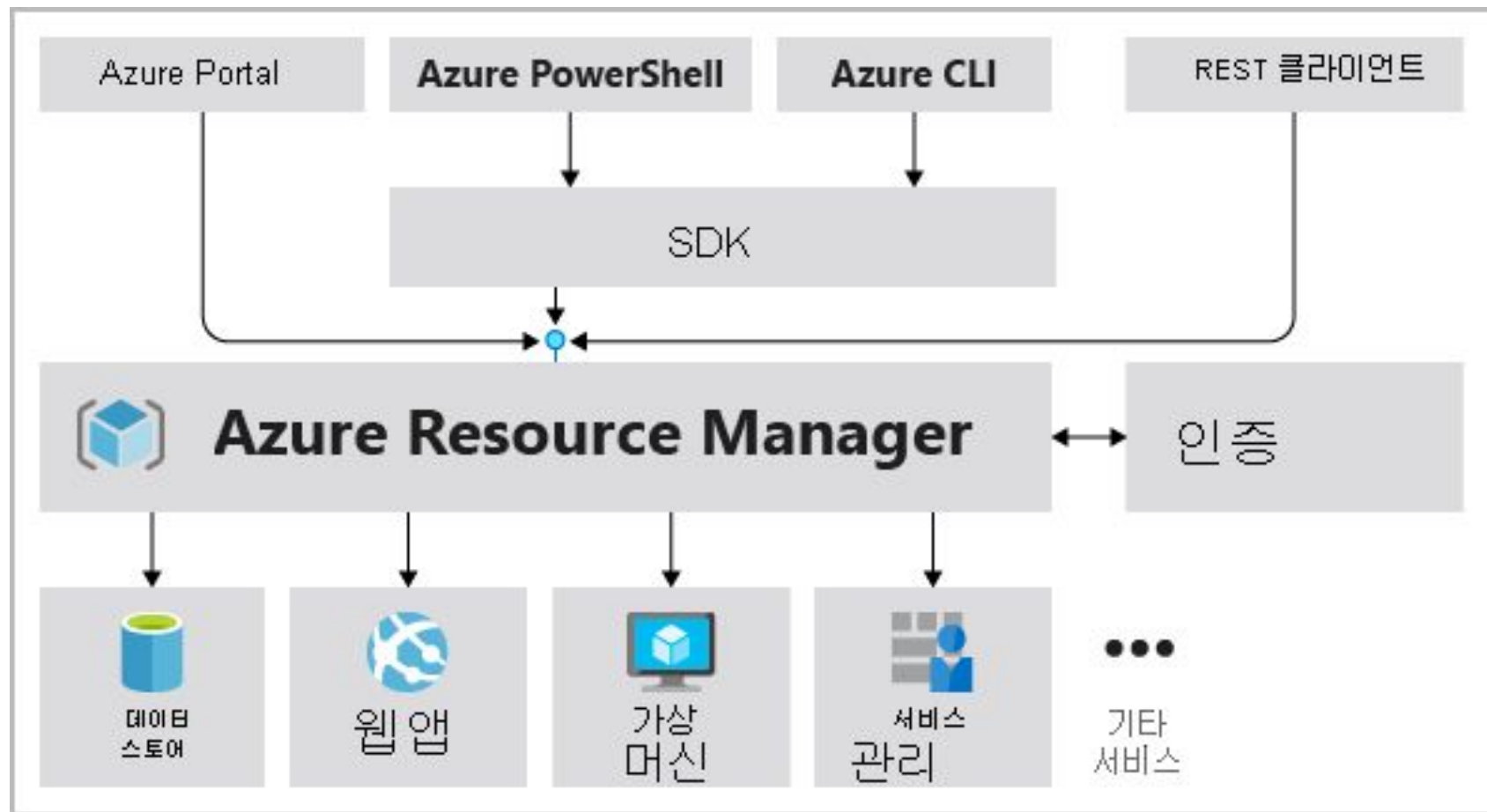
- 리소스를 보관하는 컨테이너: 리소스를 간편하게 구성, 관리
- Azure 플랫폼의 기본 요소
- 모든 리소스는 리소스 그룹에 있어야 함
- 하나의 리소스는 하나의 resource group의 멤버여야 함
- 논리적 그룹화가 중요

 리소스 그룹



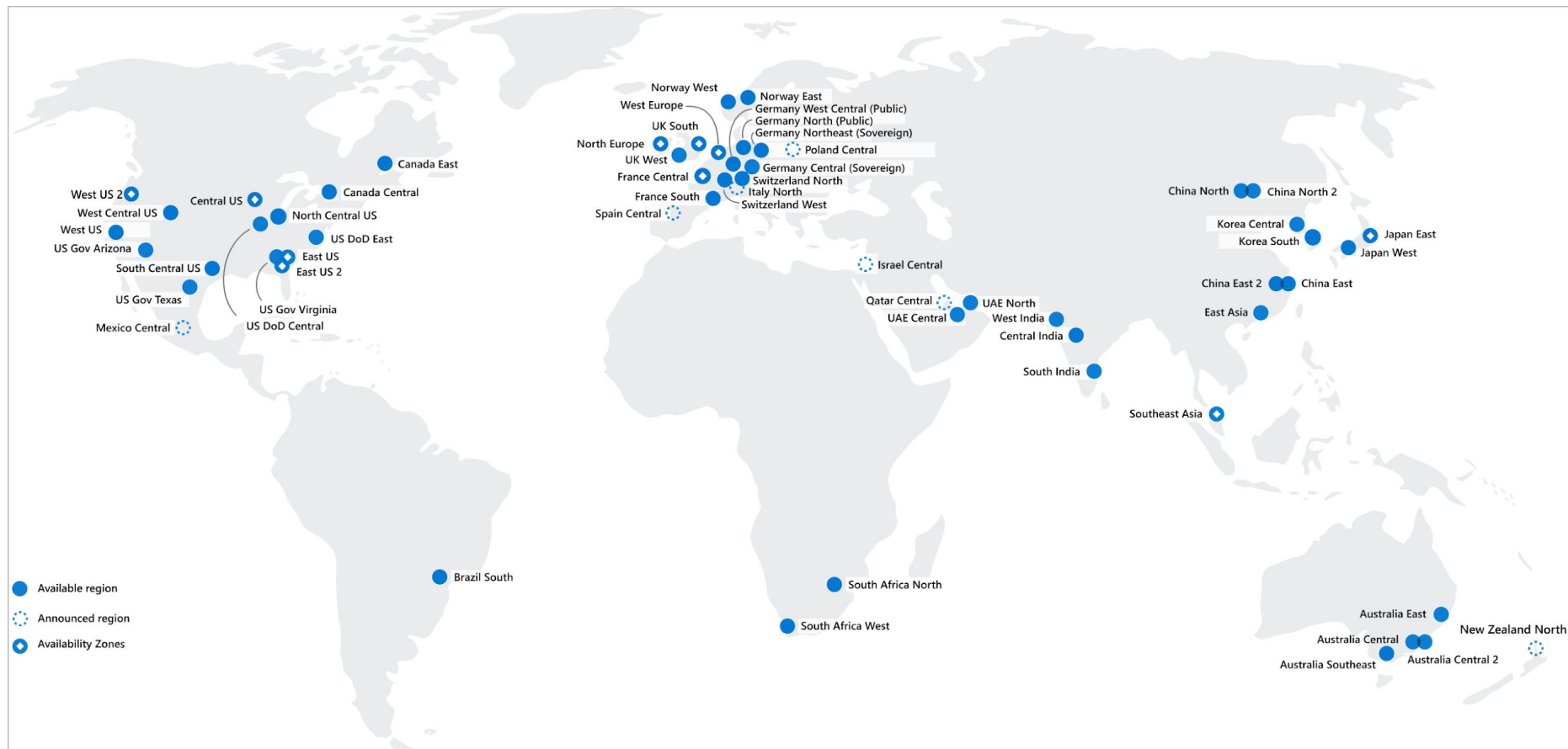
Azure 리소스 그룹 (2)

- 수명 주기: resource group을 삭제하면 모든 resource가 삭제됨
- Azure Resource Manager
 - 배포 및 관리 서비스
 - 리소스 생성, 업데이트, 삭제 등의 관리 기능
 - 액세스 제어, 잠금, 태그 등의 관리 기능

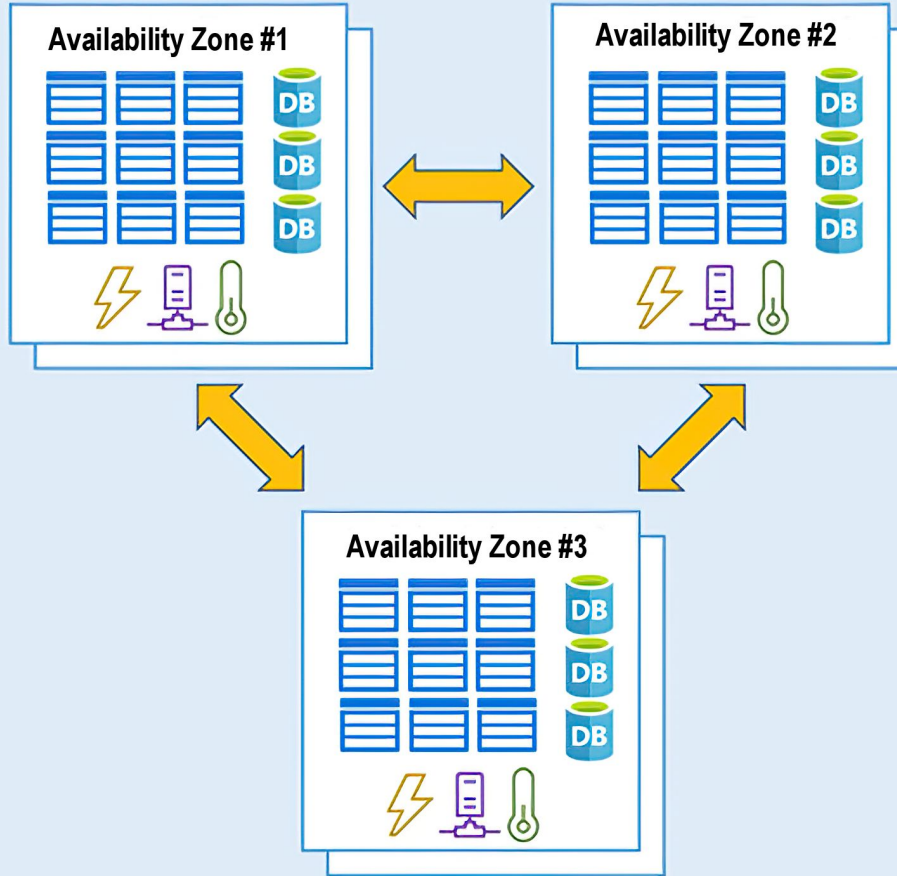


Azure 지역/가용성 영역

- 중복된 하드웨어 환경 구축 → 중복성 설정 → 높은 가용성
- 가용성 영역
 - Azure 지역 내에서 물리적으로 분리된 데이터 센터
 - 독립된 전원, 냉각 및 네트워크를 갖춘 하나 이상의 데이터 센터
 - 고속 프라이빗 광 네트워크를 통해 연결됨



Azure Region

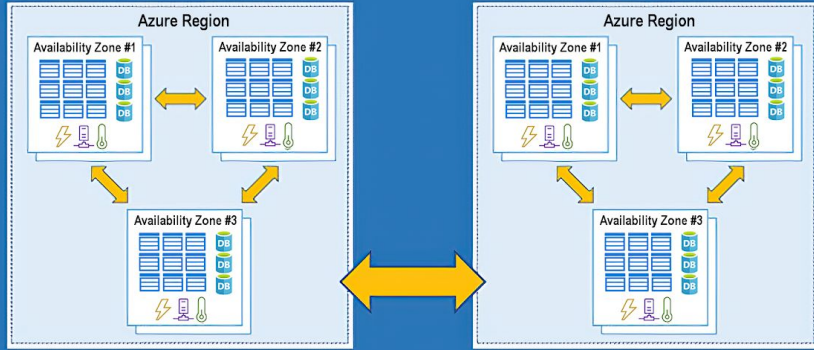


Azure 지역 쌍

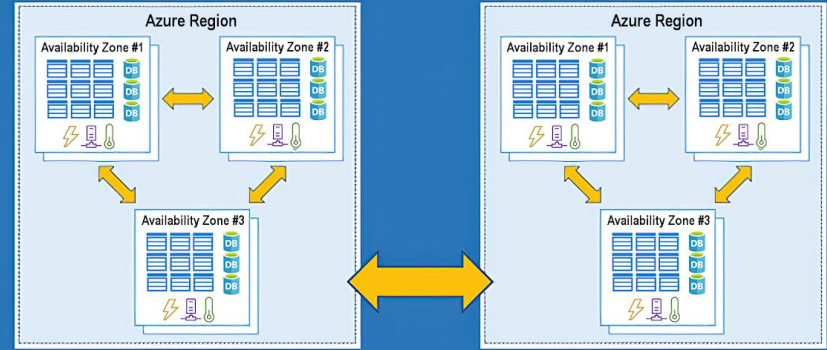
- 각 지역은 300 mile 이상 떨어져 있는 동일한 지리적 위치 내의 다른 Azure 지역과 항상 쌍을 이룸
- 미국 동부 - 미국 서부
- 동남 아시아 - 동아시아
- 일부 서비스는 자동 지역 중복 스토리지 제공

Geography

Region Pair



Region Pair



Azure 서비스 가입

- azure.microsoft.com/free



AWS Lambda

Sung-Dong Kim,
School of Computer Engineering,
Hansung University

Serverless

- ▶ 관리할 서버가 없다.
- ▶ 서버 인프라를 구축 운영하지 않고도 코드 배포만으로 backend application을 구축할 수 있는 architecture
- ▶ 서비스 자체에 집중 가능

Lambda

- ▶ 서버 실행 없이 코드를 실행시켜주는 computing service
- ▶ Code로 이루어진 function 실행
- ▶ function = lambda function → 특정 event가 발생하면 실행됨

Examples

- ▶ HTTP Server 만들기: API gateway가 호출되었을 때, lambda를 실행하도록 연결
- ▶ S3에 image upload시 thumbnail을 만들기
- ▶ 특정 event에 lambda function 연결 → lambda function 실행 후, 특정 resource를 실행하게 함 = 원하는 시점에 application code를 실행할 수 있음

Lambda 기반 HTTP API 서버 만들기

- ▶ AWS console → lambda
- ▶ 함수 생성
- ▶ 이름: hello-lambda, 환경: node.js
- ▶ 역할: resource 접근 권한을 역할로 묶음 → lambda, resource에 부여
- ▶ Index.js, exports handler = async () { ... } ➔ 진입점
- ▶ Deploy → Test
 - ▶ Event를 프린트: console.log(event)

Lambda 기반 HTTP API 서버 만들기

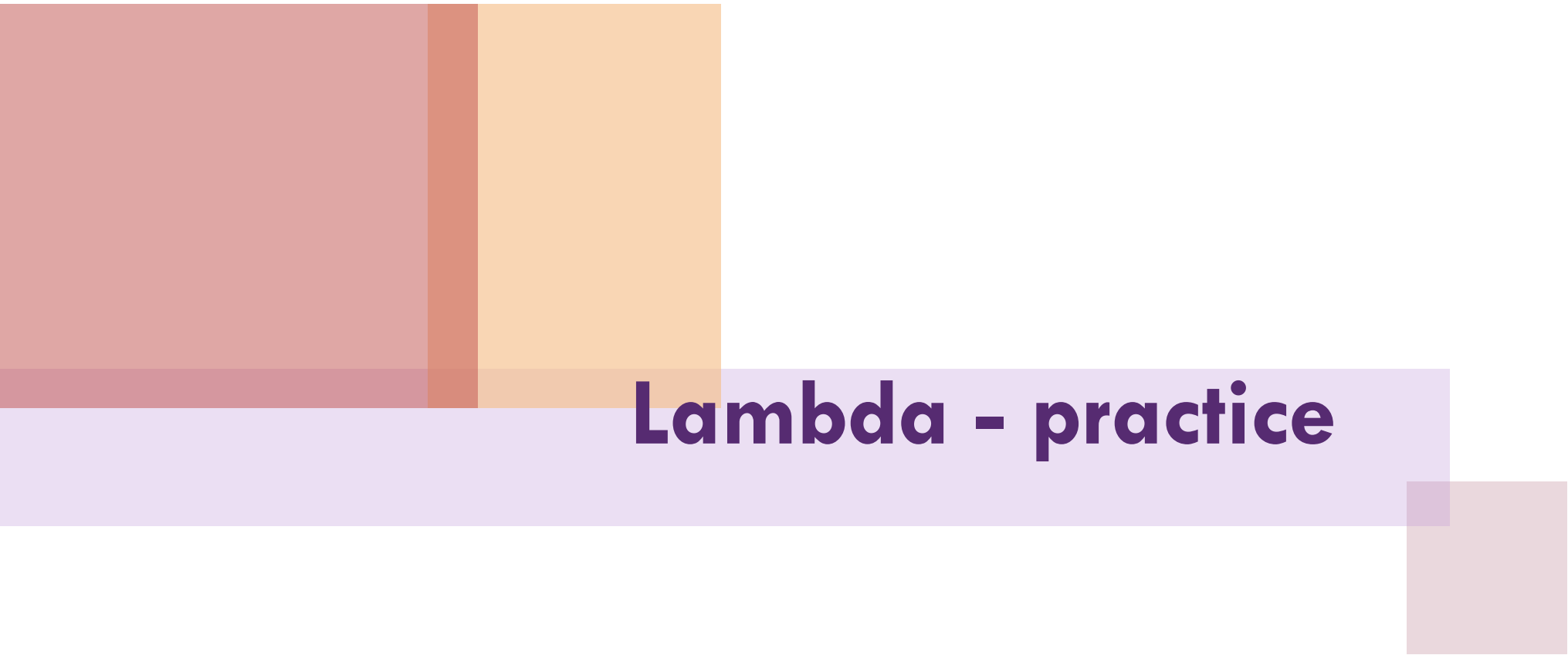
- ▶ Debugging
 - ▶ [test] tab → parameter 설정 → 호출 → [log] 확인
- ▶ Local source → zip으로 압축 → 업로드
- ▶ Lambda 함수에 API gateway 연결
 - ▶ 트리거 추가: resource를 실행 후 lambda function 실행
 - ▶ API gateway → API 생성 → HTTP API → 보안 (open)
 - ▶ [구성] tab : trigger 확인
 - ▶ 세부정보: API endpoint → browser

장/단점

- ▶ 서버 고민 없음
- ▶ 자동 auto scaling → 개발 생산성 (빠르게 개발)
- ▶ 사용한 만큼 비용 (millisecond 단위)
- ▶ Running time 제한: 최대 15분

사례

- ▶ 직방
- ▶ 당근마켓
- ▶ Jobkorea: lambda 검색



Lambda - practice

Sung-Dong Kim,
School of Computer Engineering,
Hansung University

Contents

- ▶ AWS Lambda 함수 만들기
- ▶ 매분마다 함수를 트리거 하는 Amazon CloudWatch 이벤트 생성
- ▶ 함수가 IAM 역할을 사용하여 실행중인 EC2 instance를 중지 시킴



1. Lambda 함수 생성

- ▶ Lambda → 함수 생성
- ▶ 새로 작성 (author from scratch)
- ▶ 이름: myStopinator
- ▶ runtime: python 3.8
- ▶ 기본 실행 역할 변경
 - ▶ Amazon CloudWatch Logs에 로그를 업로드 할 수 있는 권한

2. 트리거 구성

- ▶ [+ 트리거 추가]: EventBridge(CloudWatch Events)
- ▶ 규칙 생성
 - ▶ 이름: `everyMinute`
 - ▶ 예약 표현식: `rate(1 minute)`

3. Lambda 함수 구성

▶ 코드

```
import boto3
region = 'us-east-1'
instances = ['instance-id']
ec2 = boto3.client('ec2', region_name=region)

def lambda_handler(event, context):
    ec2.stop_instances(InstanceIds=instances)
    print('stopped your instances: ' + str(instances))
```

4. Lambda 함수 작동 확인

- ▶ 인스턴스가 중지되었나?