

프론트엔드 개발자의 시선으로 다시 설계한 웹 클라우드 인프라

AWS를 처음 접한 프론트엔드 개발자가
AWS 인프라를 재설계하며 겪은 판단과 변화의 기록

소개



송하람

이지스엔터프라이즈(주)

<https://github.com/haramsong>

<https://blog.hrsong.com/>

<https://www.linkedin.com/in/haramsong/>

목차

1. AWS 접하게 된 계기
2. 프론트엔드 인프라 변환 과정
3. 자동화의 시작
4. 느낀 점 & 마무리

AWS를 접하게 된 계기

배경

- 회사 내 AWS 인프라를 함께 관리할 사람이 필요
- 프론트엔드 개발자로 근무하며 AWS를 처음 접함
- 운영보다 ‘구조 이해’가 먼저

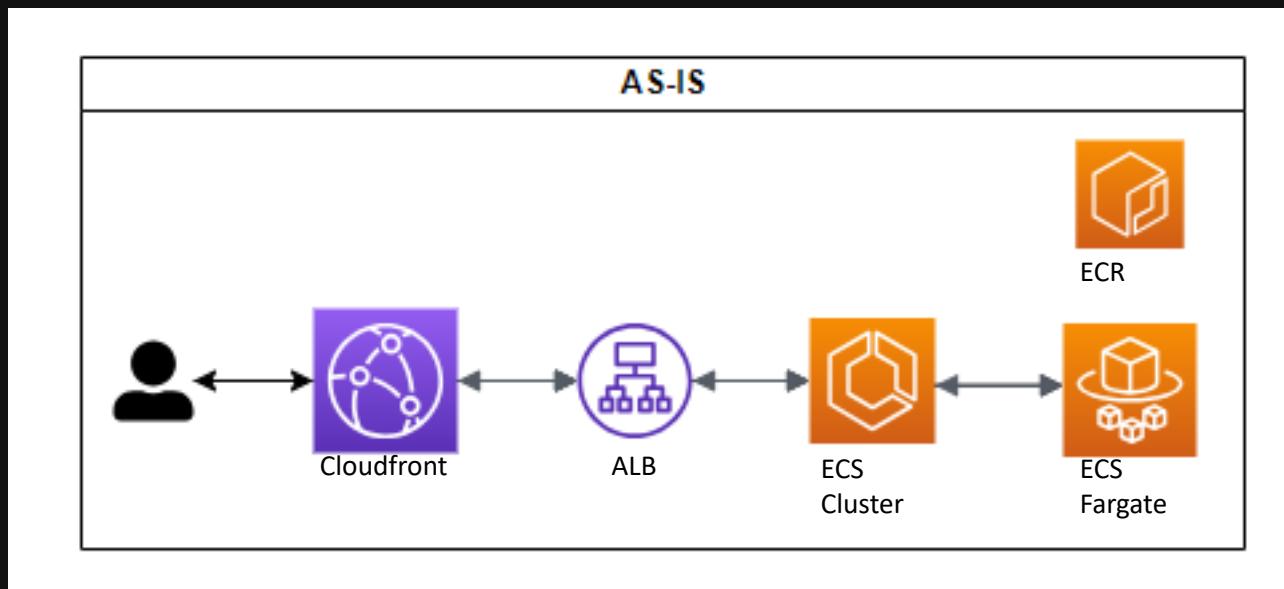
AWS를 공부하며 든 의문

- 프론트엔드 인프라도 AWS에 존재
- SSR을 사용하지 않는 Nuxt 서비스
- 컨테이너에서 서빙하는 정적 리소스

프론트엔드 인프라 변환 과정

AS-IS 아키텍처

- 모든 요청을 거치는 서버 아키텍처
- 정적 파일도 ECS + Fargate에서 제공
- 배포, 비용, 운영 모두 복잡



프론트엔드 관점에서 본 문제

- 정적 서비스에 과한 서버 구조
- 긴 배포시간 + 많은 실패 지점
- 비용 대비 낮은 효율

새로운 관점

- Nuxt는 이미 SSG 형태로 동작 중
- 정적 웹 사이트로 배포 가능
- 구조를 단순화할 수 있겠다는 확신

TO-BE 아키텍처 설계

- 정적 리소스는 CDN을 통해 제공
- 백엔드 API는 별도 ECS 서비스 유지
- 역할이 명확해진 구조

정적 웹 사이트 배포 시도

- Jenkins 파이프라인 수정(ECS -> S3)
- 퍼블릭 엑세스 허용
- 버킷 정책 허용
- 정적 웹 사이트 호스팅 활성화
- 버킷 웹사이트 엔드포인트 연결

```
{  
    "Version": "2008-10-17",  
    "Statement": [  
        {  
            "Sid": "AllowGetObjectsFromBucket",  
            "Effect": "Allow",  
            "Principal": "*",  
            "Action": "s3:GetObject",  
            "Resource": "arn:aws:s3:::<s3-bucket-name>/*"  
        }  
    ]  
}
```

정적 웹 사이트 호스팅 편집 [정보](#)

정적 웹 사이트 호스팅
이 버킷을 사용하여 웹 사이트를 호스팅하거나 요청을 리디렉션합니다. [자세히 알아보기](#)

비활성화
 활성화

호스팅 유형
 정적 웹 사이트 호스팅
버킷 엔드포인트를 웹 주소로 사용합니다. [자세히 알아보기](#)
 객체에 대한 요청 리디렉션
요청을 다른 버킷 또는 도메인으로 리디렉션합니다. [자세히 알아보기](#)

ⓘ 고객이 웹 사이트 엔드포인트의 콘텐츠에 액세스할 수 있게 하려면 모든 콘텐츠를 공개적으로 읽기 가능하도록 설정해야 합니다. 이렇게 하려면, 버킷에 대한 S3 퍼블릭 액세스 차단 설정을 편집하면 됩니다. 자세한 내용은 [Amazon S3 퍼블릭 액세스 차단 사용](#)을 참조하십시오.

인덱스 문서
웹 사이트의 홈 페이지 또는 기본 페이지를 지정합니다.
index.html

오류 문서 - 선택 사항
오류가 발생하면 반환됩니다.
404.html

리디렉션 규칙 - 선택 사항
JSON으로 작성된 리디렉션 규칙은 특정 콘텐츠에 대한 웹 페이지 요청을 자동으로 리디렉션합니다. [자세히 알아보기](#)

드러난 한계

- Public하게 공개된 버킷
- AWS S3 엔드포인트 주소를 그대로 사용
- S3 자체에 GET 요청 + 데이터 전송 요금 발생

Amazon S3, 여러 HTTP 오류 코드에 대한 과금 중단

게시된 날짜: 2024년 8월 19일

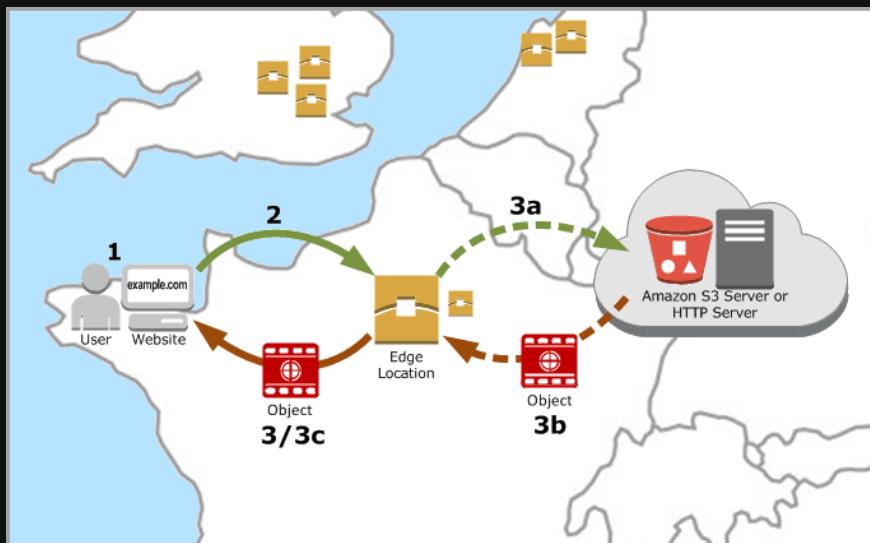
고객이 요청하지 않은 무단 요청이 무료로 처리되는 Amazon S3의 변경 사항이 완료되었습니다. 이번 변경으로 버킷 소유자는 개별 AWS 계정 또는 AWS 조직 외부에서 시작된 경우 HTTP 403(Access Denied) 오류 응답을 반환하는 요청에 대해 요청 또는 대역폭 요금이 발생하지 않습니다. 무료로 제공되는 오류 코드의 전체 목록을 보려면 [Amazon S3 오류 응답에 대한 요금 청구](#)를 참조하세요. 이 청구 변경으로 인해 고객 애플리케이션을 변경할 필요는 없으며, 변경 내용은 모든 S3 버킷에 적용됩니다.

이 변경 사항은 [2024년 5월 13일에 처음 발표되었으며](#), 몇 주 내에 대부분의 S3 API에 적용되었습니다. 이제 이 변경 사항은 AWS GovCloud(미국) 리전 및 AWS 중국 리전을 비롯한 모든 AWS 리전의 S3 API에 모두 적용됩니다. 자세히 알아보려면 S3 사용 설명서의 Amazon S3 오류 응답 및 [오류 응답에 대한 요금 청구](#)를 참조하세요.

리스크는 줄었으나 그래도 보안 신경은 필수!

보안과 라우팅 설계

- Cloudfront를 사용 결정
- S3 퍼블릭 접근 차단
- CloudFront OAC로 오리진 보호



https://docs.aws.amazon.com/ko_kr/AmazonCloudFront/latest/DeveloperGuide/HowCloudFrontWorks.html

OAC란?

기존 OAI와 다른 S3 오리진을 보호하는 기능

- 보안 - 배포의 보안 태세를 강화
- 포괄적인 HTTP 메서드 지원
- SSE-KMS로 암호화된 S3 객체 다운로드 및 업로드를 지원
- 모든 AWS 리전에서 S3 접근

OAC란?

CloudFront > Distributions > create

Create distribution

Origin

Origin domain
Choose an AWS origin, or enter your origin's domain name.

Origin path - optional Info
Enter a URL path to append to the origin domain name for origin requests.

Name
Enter a name for this origin.

Origin access Info

- Public**
Bucket must allow public access.
- Origin access control settings (recommended)**
Bucket can restrict access to only CloudFront.
- Legacy access identities**
Use a CloudFront origin access identity (OAI) to access the S3 bucket.

Origin access control
Select an existing origin access control (recommended) or create a new configuration.

Bucket policy
Policy must allow access to CloudFront IAM service principal role.

- I will manually update the policy

⚠ You must update the S3 bucket policy
CloudFront will provide you with the policy statement after creating the distribution

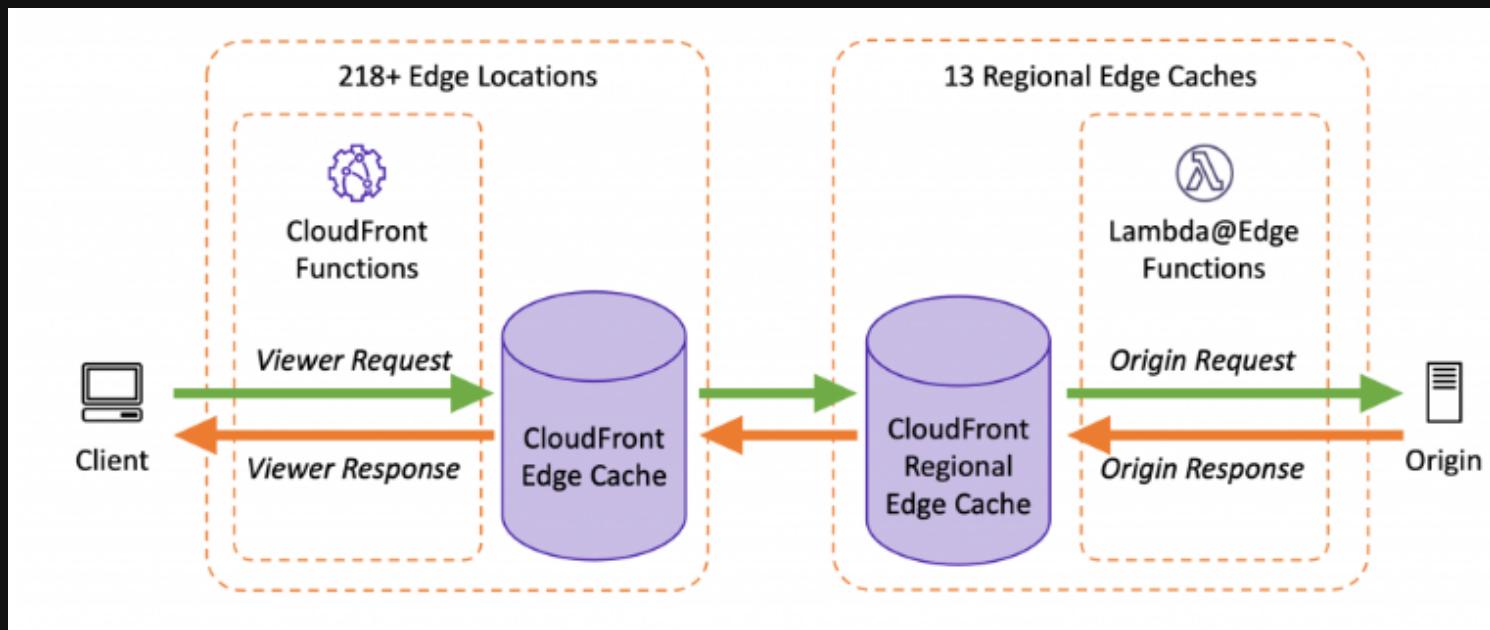
<https://aws.amazon.com/ko/blogs/korea/amazon-cloudfront-introduces-origin-access-control-oac/>

OAC란?

```
1  {
2      "Version": "2008-10-17",
3      "Id": "PolicyForCloudFrontPrivateContent",
4      "Statement": [
5          {
6              "Sid": "AllowCloudFrontServicePrincipal",
7              "Effect": "Allow",
8              "Principal": {
9                  "Service": "cloudfront.amazonaws.com"
10             },
11             "Action": "s3:GetObject",
12             "Resource": "arn:aws:s3:::<s3-bucket-name>/*",
13             "Condition": {
14                 "StringEquals": {
15                     "AWS:SourceArn": "arn:aws:cloudfront:<account>:distribution/<cloudfront-distribution-id>"
16                 }
17             }
18         }
19     ]
20 }
```

왜 CloudFront Function인가

- 403/404 리다이렉트 비용 제거
- 엣지에서 빠른 경로 판단
- 응답 시간 개선



CloudFront Function 예시

```
var level = 0; // subdirectory level where index.html is located.
var regexExpr = /^\/.*(\.\.\w+$)/;
Complexity is 5 Everything is cool!
function handler(event) {
    var request = event.request;
    var olduri = request.uri;

    if (isRoute(olduri)) { // if is a route request. i.e: /route1
        var defaultPath = '';

        var parts = olduri
            .replace(/^\//, '') // remove leading '/'
            .replace(/\$/g, '') // remove trailing '/' if any
            .split('/'); // split uri into array of parts. i.e: ['route1', 'my-picture.png']

        var nparts = parts.length;

        // determine the limit as either level or nparts, whichever is lower
        var limit = (level < nparts) ? level : nparts;

        // build the default path. i.e: /route1
        for (var i = 0; i < limit; i++) {
            defaultPath += '/' + parts[i];
        }

        var newuri = defaultPath + '/index.html';

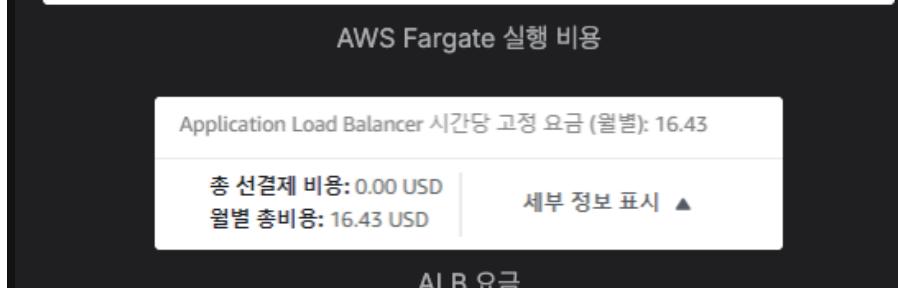
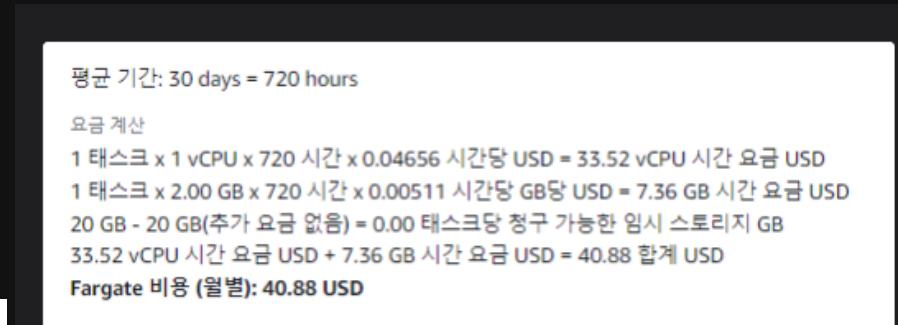
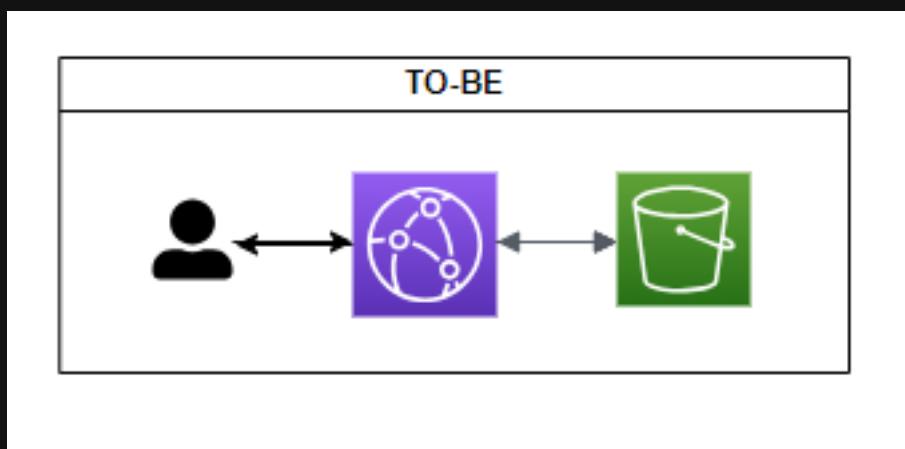
        request.uri = newuri;
        console.log('Request for [' + olduri + '], rewritten to [' + newuri + ']');
    }

    return request;
}

// Returns true if uri is a route. i.e: /route1
// Returns false if uri is a file. i.e: /route1/index.html
function isRoute(uri) {
    return !regexExpr.test(uri);
}
```

TO-BE 아키텍처

- Cloudfront + S3 조합으로 단순화
- ECS, ALB 미사용으로 인한 비용 절감
- 배포 시간 감소



자동화의 시작

인프라 개선 후 생긴 불편함

- 매번 새 설정마다 반복적인 작업
- Access Key 사용 불가로 인한 콘솔 수기 설정 불편
- 잘못된 설정 발생 가능(휴먼 에러)

Terraform + Jenkins을 활용한 자동화

- Terraform으로 인프라 선언
- Jenkins에서 build & generate
- 기존 수기 프로세스를 버튼 한번 클릭으로 변경

Terraform 코드 작성

```
1 // s3.tf
2 resource "aws_s3_bucket" "staging_website" {
3   bucket = var.bucket_name
4
5   tags = {
6     Name      = var.bucket_name
7     Environment = "Staging"
8   }
9 }
10
11 resource "aws_s3_bucket_public_access_block" "staging_website_acl" {
12   bucket = aws_s3_bucket.staging_website.id
13
14   block_public_acls      = true
15   block_public_policy    = true
16   ignore_public_acls    = true
17   restrict_public_buckets = true
18 }
19
20 resource "aws_s3_bucket_policy" "staging_website_policy" {
21   bucket = aws_s3_bucket.staging_website.id
22
23   policy = jsonencode({
24     Version = "2012-10-17",
25     Statement = [
26       {
27         Effect = "Allow",
28         Principal = {
29           Service = "cloudfront.amazonaws.com"
30         },
31         Action  = "s3:GetObject",
32         Resource = "arn:aws:s3:::${var.bucket_name}/*",
33         Condition = {
34           StringEquals = {
35             "AWS:SourceArn" = "arn:aws:cloudfront::${data.aws_caller_identity.current_arn}"
36           }
37         }
38       }
39     ]
40   })
41
42   depends_on = [aws_s3_bucket_public_access_block.staging_website_acl]
43 }
```

The screenshot shows the AWS S3 console interface. At the top, there are two tabs: '범용 버킷' (General Bucket) and '디렉터리 버킷' (Directory Bucket). The '범용 버킷' tab is selected. Below the tabs, it says '버킷은 S3에 저장되는 데이터의 컨테이너입니다.' (A bucket is a container for data stored in S3). There is a search bar labeled '이름으로 버킷 찾기' (Search for bucket by name) and a dropdown menu showing the bucket name. On the right, it says 'AWS 리전' (AWS Region) and '아시아 태평양(서울) ap'.

퍼블릭 액세스 차단(버킷 설정)

퍼블릭 액세스는 ACL(액세스 제어 목록), 버킷 정책, 액세스 지정 정책 또는 모두를 통해 버킷 및 객체에 부여됩니다. 모든 S3 버킷 및 퍼블릭 액세스 차단을 활성화하도록 권장하지만, 이 설정을 적용하기 전에 퍼블릭 액세스가 없어도 애플리케이션이 올바르게 작동하는지 확인하세요.

모든 퍼블릭 액세스 차단

▶ 이 버킷의 개별 퍼블릭 액세스 차단 설정

버킷 정책

JSON으로 작성된 버킷 정책은 버킷에 저장된 객체에 대한 액세스 권한을 제공합니다. 버킷 정책은 다른 계정이 소유하는 버킷에 대해 퍼블릭 액세스 차단 설정이 활성화되어 있어야 퍼블릭 액세스가 차단됩니다.

① 이 버킷에 대해 퍼블릭 액세스 차단 설정이 활성화되어 있어 때문에 퍼블릭 액세스가 차단됩니다.
활성화된 설정을 확인하려면 이 버킷의 퍼블릭 액세스 차단 설정을 확인하세요. [Amazon S3 퍼블릭 액세스 차단 설정](#)

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "Service": "cloudfront.amazonaws.com"
      },
      "Action": "s3:GetObject",
      "Resource": "arn:aws:s3:::staging_website/*",
      "Condition": {
        "StringEquals": {
          "AWS:SourceArn": "arn:aws:cloudfront::${data.aws_caller_identity.current_arn}"
        }
      }
    }
  ]
}
```

Jenkins stage - Plan

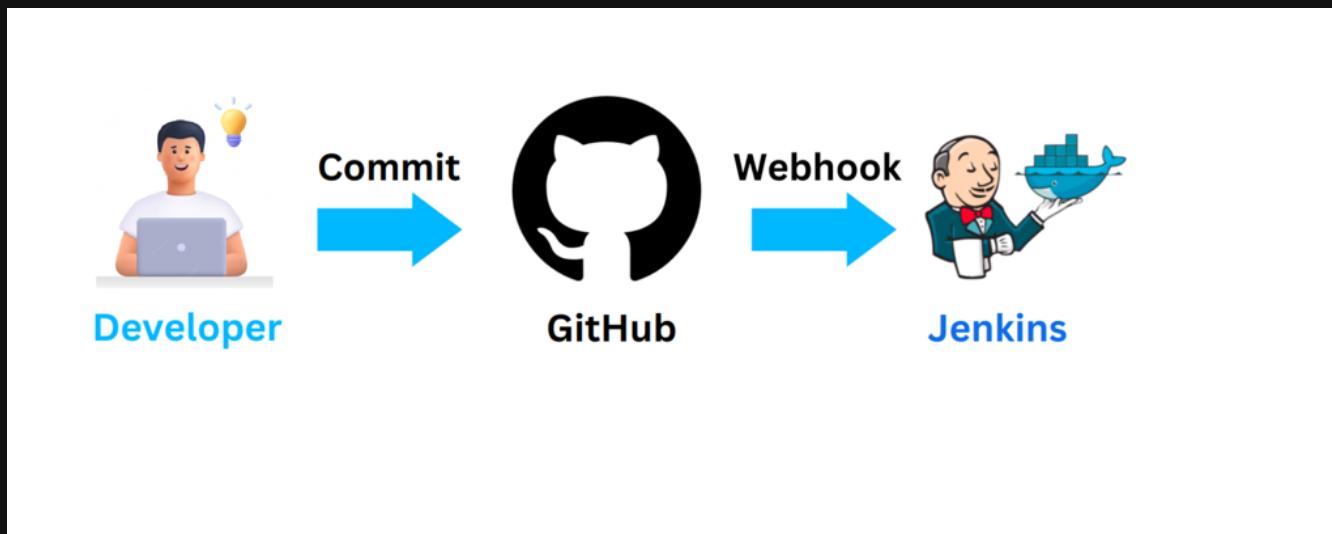
```
1 stage('Check Infra') {
2     when {
3         expression {
4             def cleanInfraChecked = CLEAN_INFRA.equals("false")
5             if (!cleanInfraChecked) {
6                 echo "CLEAN_INFRA value is checked..."
7             }
8             return cleanInfraChecked
9         }
10    }
11
12    steps{
13        dir(BUCKET_NAME) {
14            script{
15                echo "Terraform plan::: Start"
16                sh '''
17                    terraform plan \
18                        --var="provider_region=${AWS_DEFAULT_REGION}" \
19                        --var="bucket_name=${BUCKET_NAME}" \
20                        --var="acm_id=${ACM_ID}" \
21                        --var="custom_cache_policy_id=${CUSTOM_CACHE_POLICY_ID}" \
22                        --var="cloudfront_function_name=${CLOUDFRONT_FUNCTION_NAME}" \
23                        ...
24                echo "Terraform plan::: Finished"
25            }
26        }
27    }
28 }
```

Jenkins stage - Deploy

```
steps{
    dir(BUCKET_NAME) {
        script{
            echo "Deploy Infra::: Start"
            sh '''
                terraform apply \
                    --var="provider_region=${AWS_DEFAULT_REGION}" \
                    --var="bucket_name=${BUCKET_NAME}" \
                    --var="acm_id=${ACM_ID}" \
                    --var="custom_cache_policy_id=${CUSTOM_CACHE_POLICY_ID}" \
                    --var="cloudfront_function_name=${CLOUDFRONT_FUNCTION_NAME}" \
                    --auto-approve
                ...
            echo "Deploy Infra::: Finished"
        }
    }
}
```

인프라에서 그치지 않은 자동화

- PRO이 병합되면 개발계에 매번 수기 배포
- Webhook을 활용한 자동 배포 프로세스 구현
- Jenkins + Slack 연동으로 편리한 배포 확인



Github Webhook

Webhooks / Manage webhook

We'll send a POST request to the URL below with details of any subscribed events. You can also specify which data format you'd like to receive (JSON, x-www-form-urlencoded, etc). More information can be found in [our developer documentation](#).

Payload URL *

`https://your-jenkins-ip/generic-webhook-trigger/invoke?token=`

Content type

application/json

Secret

(empty)

SSL verification

By default, we verify SSL certificates when delivering payloads.

Enable SSL verification [Disable \(not recommended\)](#)

Which events would you like to trigger this webhook?

- Just the push event.
- Send me everything.
- Let me select individual events.

Active

We will deliver event details when this hook is triggered.

[Update webhook](#)

[Delete webhook](#)

Jenkins Webhook Trigger

Generic Webhook Trigger ?
Is triggered by HTTP requests to http://JENKINS_URL/generic-webhook-trigger/invoke
There are example configurations in [the Git repository](#).
You can fiddle with JSONPath [here](#). You may also want to checkout the syntax [here](#).
You can fiddle with XPath [here](#). You may also want to checkout the syntax [here](#).
You can fiddle with regular expressions [here](#). You may also want to checkout the syntax [here](#).
If your job is **not parameterized**, then the resolved variables will just be contributed to the build. If your job is **parameterized**, and you resolve variables that have the same name as those parameters, then the plugin will populate the parameters when triggering job. That means you can, for example, use the parameters in combination with an SCM plugin, like GIT Plugin, to pick a branch.

Post content parameters

Variable
Name of variable

Expression
 \$.ref
 JSONPath
 XPath

Expression to evaluate in PO

Value filter

Optional. Anything in the evaluated expression can be used as a variable in the configuration.

Optional filter

Expression

Regular expression to test on the evaluated text specified below. The regexp syntax is documented [here](#).

Text

Text to test for the given expression. You can use any combination of the variables you configured above.

This is an optional feature. If specified, this job will only trigger when given expression matches given text.

Slack 알림 추가

```
stage('Send Slack') {
    steps {
        script {
            def customBuildUrl = env.BUILD_URL.replaceAll("^http://[^/]+", CUSTOM_DOMAIN)
            def buildUser = env.BUILD_USER ?: "자동 실행"
            def message = """
                @here
                ● *빌드 시작*
                - 📁 Project: *전자공문 개발계 알파서버*
                - 🏢 Job: *${env.JOB_NAME}*
                - 🏠 Build: *#${env.BUILD_NUMBER}*
                - 🌱 Branch: *${params.GIT_BRANCH ?: 'N/A'}*
                - 🚀 Triggered by: *${buildUser}*
                - ⏳ Started at: *${new Date().format("yyyy-MM-dd HH:mm:ss")}*
                - 🔗 <${customBuildUrl}|빌드 확인>
                - 🎵 Recent Commits:
                """".stripIndent() + "\n" +
                """` ${env.LAST_COMMITS}`"""
                slackSend channel: 'redacted', message: message
        }
    }
}
```

Slack 알림 확인

jenkins 웹 오후 4:46

@here

빌드 시작

- Project: 전자공문 개발계
- Job: [REDACTED]
- Build: #39
- Branch: main
- Triggered by: 자동 실행
- Started at: 2025-06-04 16:46:31
- ⚙️ 빌드 확인
- 🔄 Recent Commits:

b337601 - hyunz: 🎉 기관, 계정 목록 No 7자단위부터 ... 되게 수정, 페이지네이션 10단위로 디풀트 고정 (17초 전)

08b8e81 - seungheon1029: 🔥 모달 담드 영역 클릭시 종료 안되게 클릭 이벤트 삭제 (2시간 전)

7d2590f - sanguneo: :bug: 전체목록 수와 페이지 목록수가 같거나 작으면 체크로직 건너뛰던 부분 제거 (8시간 전)

4:46 @here

✓ 빌드 성공 🎉

- Project: 전자공문 개발계
- Job: [REDACTED]
- Build: #39
- Branch: main
- Triggered by: 자동 실행
- 소요 시간: 26 sec
- ⚙️ 빌드 확인
- 🌎 배포 확인

느낀 점 & 마무리

재설계 이후 변화

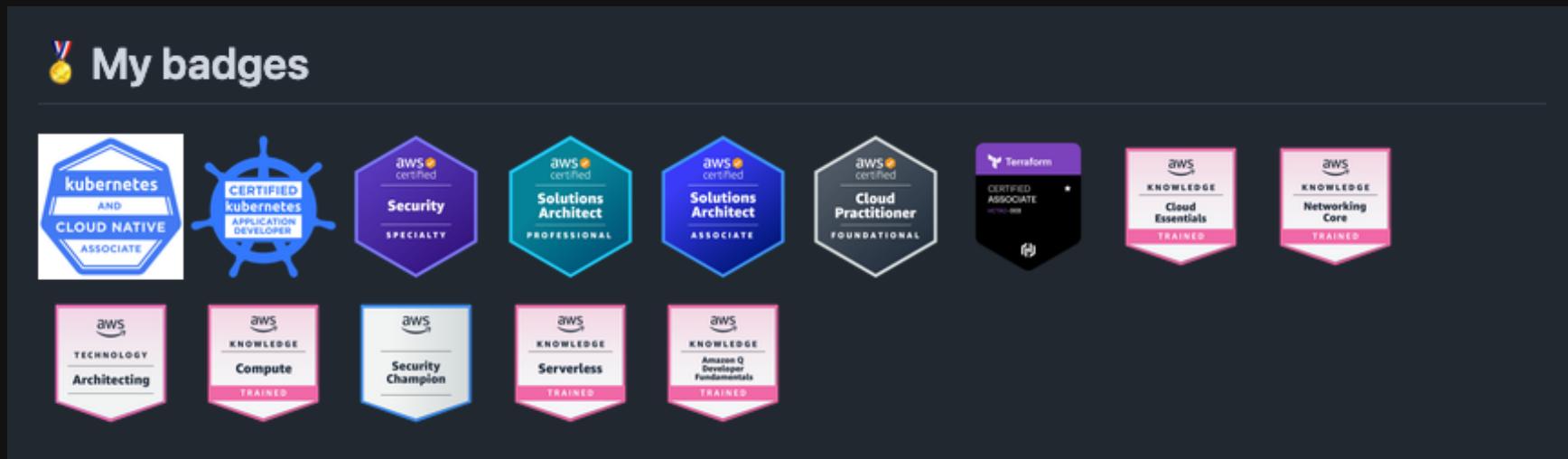
- 아키텍처 구조 단순화
- 배포 시간 단축
- 월 수백 달러 비용 절감
- 수기 업무 자동화로 인한 개발 생산성 향상

개발자로서 배운 점

- 인프라는 특정 직군의 전유물 X
- 프론트엔드 요구사항 -> 설계 기준
- 아는 만큼 구조는 단순

인간으로서 배운 점

- 지식 습득에만 그치는 게 아니라 자신의 가치를 증명하는 과정이 필요
- 개인의 변화에 그치지 않은 타인들의 변화
- 한번 얻은 신뢰는 다른 기회를 잡을 수 있는 환경이 주어짐



감사합니다