



Serverless Korea

THE SERVERLESS

사라진 떡국 레시피 되찾기





Serverless Korea

나이를 먹는 방법

아빠, 나이는
어떻게 먹어?

나이? 떡국을
먹으면 된다

떡국 레시피가 어디 있더라…



Serverless Korea

사라진 떡국 레시피

전 세계 서버에서
완전히 사라진 떡국 레시피

Google

떡국 레시피가 없다고?



NAVER



인류가 나이를 먹는 것을
방해하는 세력들



Serverless Korea

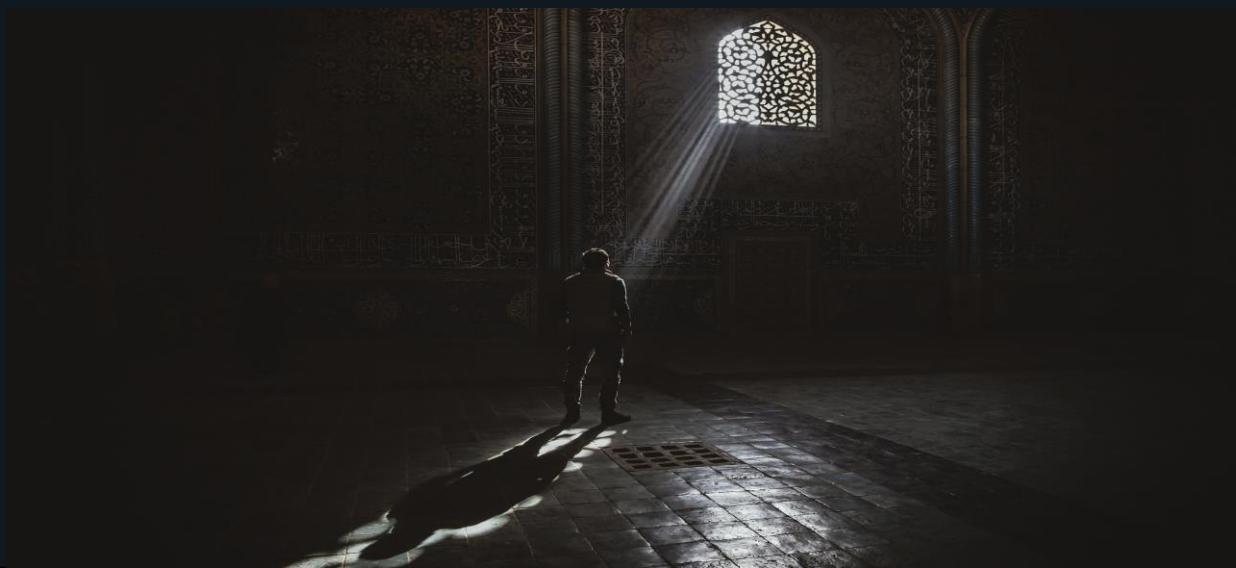
서버없는 레시피 구현

큰 일이야, 방법이 없어

더 이상 인류가 나이를 먹을
방법은 없는가…



마지막 희망, Serverless



목차

01

노아론

떡국, 서비스로 만들다
서비스로 얻은 것
전체적인 구현 방식

02

김유진

떡국 레시피와 플로우 차트
세부적인 구현 방식
로직앱과 파워앱

03

김홍민

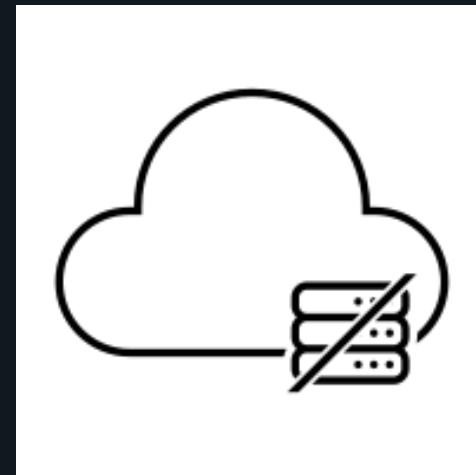
실제 결과
챌린지
활용방안

떡국, 서비스로 만들다



떡국 레시피를 만들다

떡국, 서비스로 만들다



서비스로
떡국 레시피를 만들다

떡국, 서비스로 만들다



서로 사용하는 언어가 다름

떡국, 서비스로 만들다



풀리그랏!

떡국, 서비스로 만들다



애저 평션

떡국, 서비스로 만들다



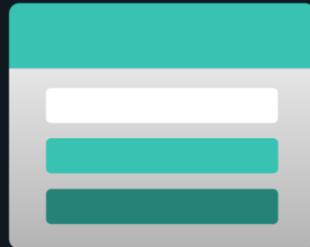
Serverless Korea



애저 평션

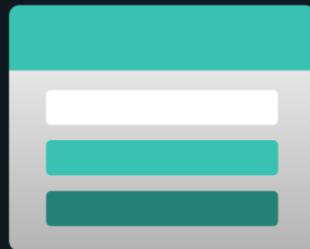
듀여러블 평션

떡국, 서비스로 만들다



블록 스토리지

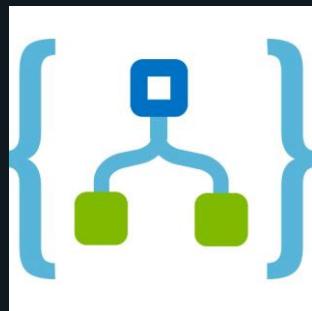
떡국, 서비스로 만들다



블롭 스토리지

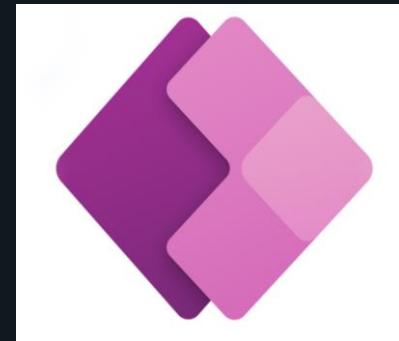
떡국이미지 저장

떡국, 서비스로 만들다



로직 앱

떡국, 서비스로 만들다



파워 앱



파워 오토메이트

클라이언트 구현, 로직앱과 연결

떡국, 서비스로 만들다



파워 앱

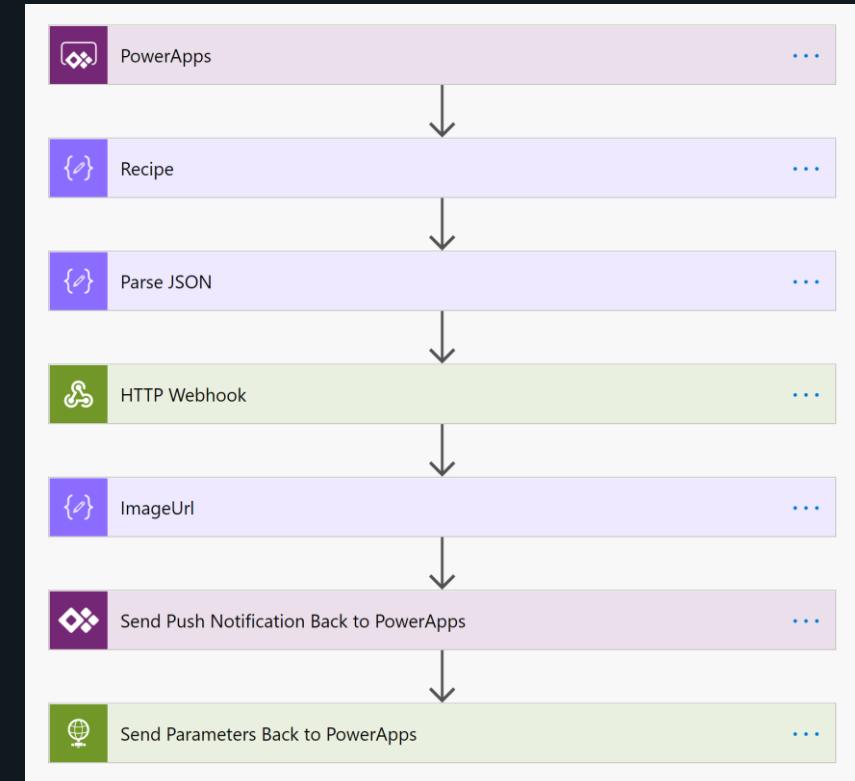
Time to soak in mins
Time to slice in mins
Time to stir-fry in mins
Time to boil in mins
roharon@studentambassadors.com
<input checked="" type="checkbox"/> I bought sliced garae-tteok!
<input checked="" type="checkbox"/> I want to add pepper!

Cook for me!

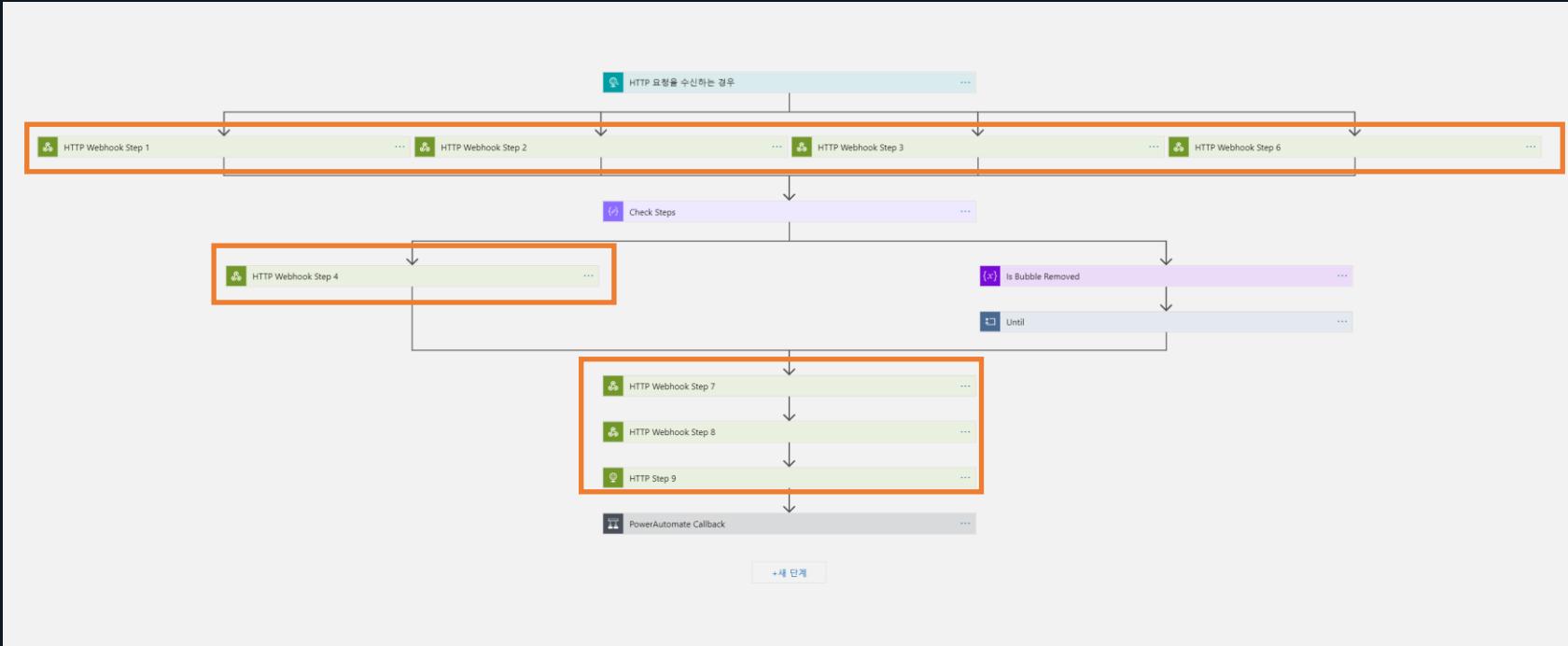
떡국, 서비스로 만들다



파워 오토메이트



서버리스로 얻은 것



레시피를 위한 9가지의 함수를 이용해
로직앱 구성

서버리스로 얻은 것



서버리스를 사용하지 않았다면?

서버리스로 얻은 것

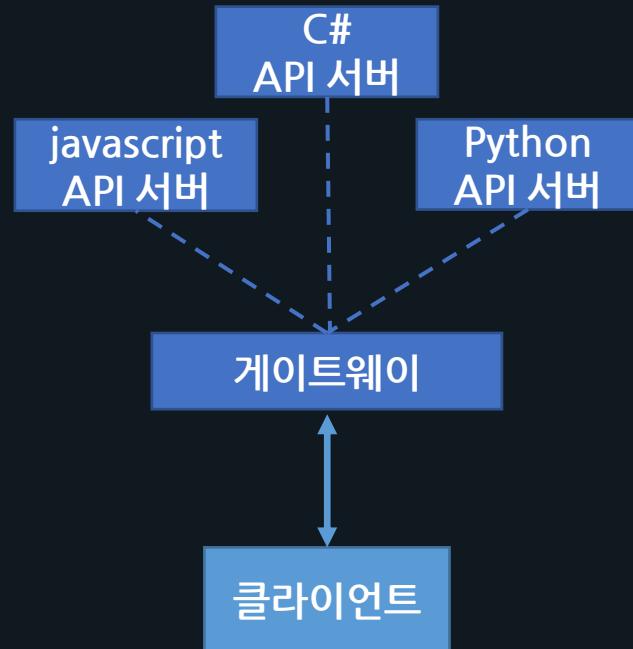


Serverless Korea

서버리스를 사용하지 않았다면?



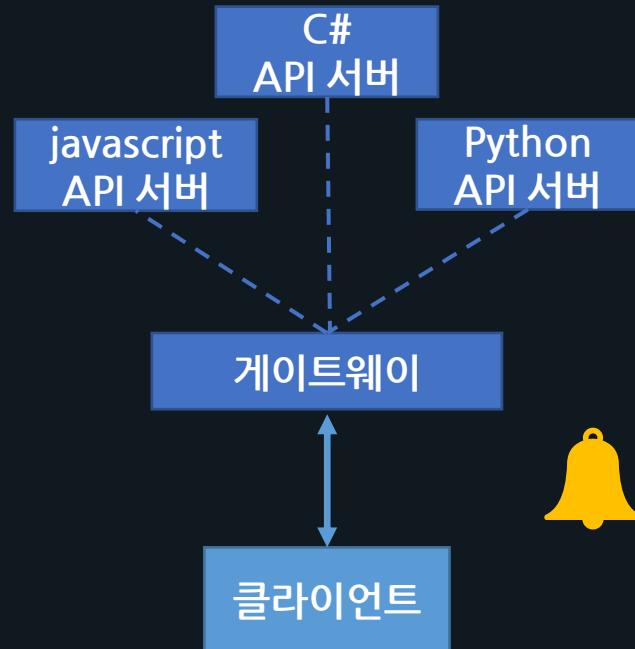
서버리스로 얻은 것



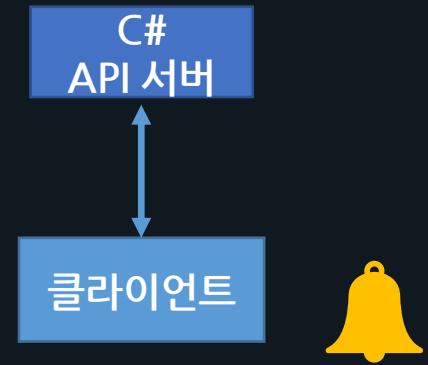
서버리스를 사용하지 않았다면?



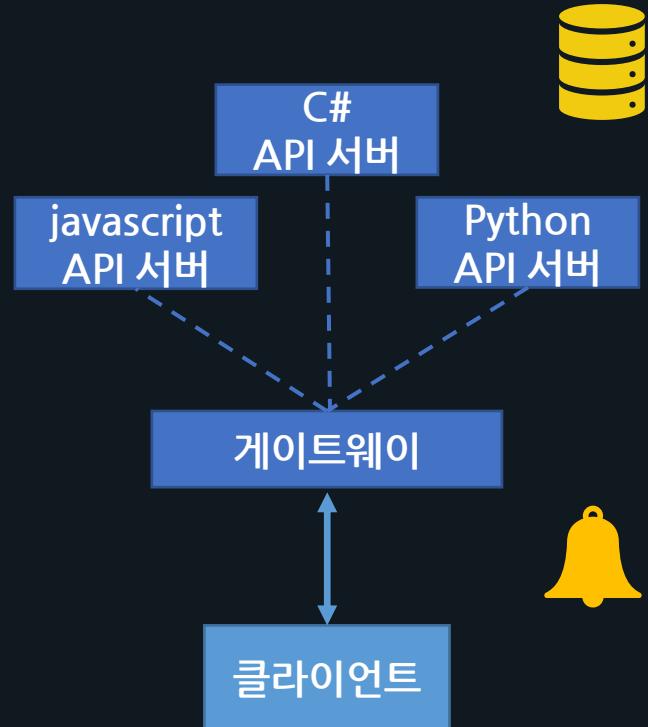
서버리스로 얻은 것



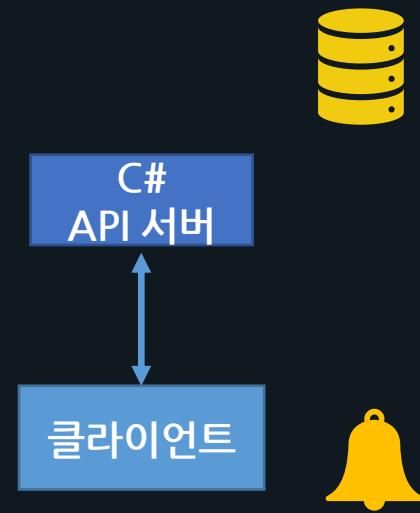
서버리스를 사용하지 않았다면?



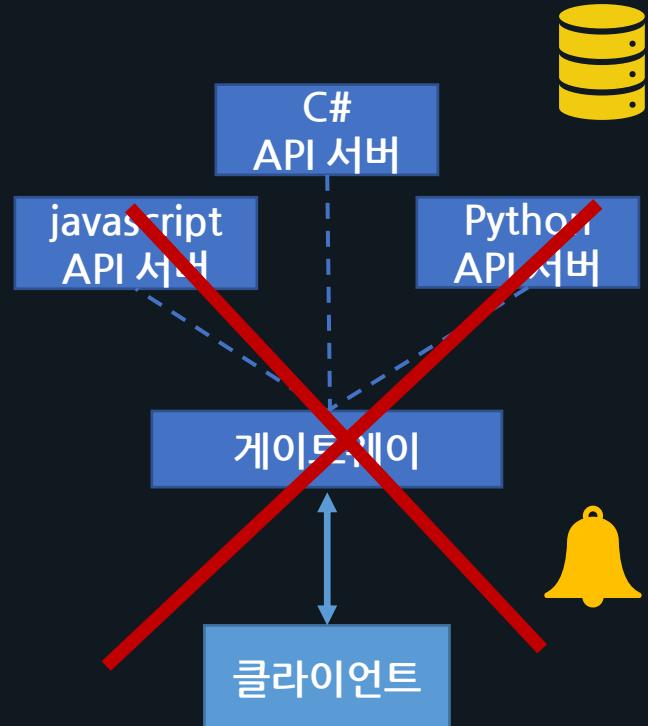
서버리스로 얻은 것



서비스를 사용하지 않았다면?



서버리스로 얻은 것



사용한다. 서버리스!



서버리스로 얻은 것



비용적인 측면

서비스 컴퓨팅 사용

검색

Storage Accounts	블록 Blob Storage, 범용 V2, LRS 중복성, 핫 액세스...	선풀: ₩0.00	월간: ₩61.07
Azure Function	소비 계층, 128MB 메모리, 100밀리초 실행 시간, 10...	선풀: ₩0.00	월간: ₩0.00
Azure Function	소비 계층, 128MB 메모리, 100밀리초 실행 시간, 10...	선풀: ₩0.00	월간: ₩0.00
Azure Function	소비 계층, 128MB 메모리, 100밀리초 실행 시간, 10...	선풀: ₩0.00	월간: ₩0.00
Azure Function	소비 계층, 128MB 메모리, 100밀리초 실행 시간, 10...	선풀: ₩0.00	월간: ₩0.00
Azure Function	소비 계층, 128MB 메모리, 100밀리초 실행 시간, 10...	선풀: ₩0.00	월간: ₩0.00
Azure Function	소비 계층, 128MB 메모리, 100밀리초 실행 시간, 10...	선풀: ₩0.00	월간: ₩0.00
Azure Function	소비 계층, 128MB 메모리, 100밀리초 실행 시간, 10...	선풀: ₩0.00	월간: ₩0.00
Azure Function	소비 계층, 128MB 메모리, 100밀리초 실행 시간, 10...	선풀: ₩0.00	월간: ₩0.00
Azure Function	소비 계층, 128MB 메모리, 100밀리초 실행 시간, 10...	선풀: ₩0.00	월간: ₩0.00
Azure Logic Apps	10,000개 작업 실행 x 1일, 100개 표준 커넥터 실행 ...	선풀: ₩0.00	월간: ₩634.86

지원

지원: 포함 항목 ₩0.00

포함 항목

프로그램 및 제품

라이선스 프로그램:

Microsoft Online Services 계약

개발/테스트 가격 표시

예상 선불 비용

월별 예상 비용 ₩0.00 ₩16,476.12

내보내기 저장 다른 이름으로 저장 공유

가상 머신 사용

검색

가상 머신	1 A0(1 vCPU, 0.75GB RAM) x 1 월; Linux – Ubuntu;...	선풀: ₩0.00	월간: ₩16,476.12
-------	--	-----------	----------------

지원

지원: 포함 항목 ₩0.00

포함 항목

프로그램 및 제품

라이선스 프로그램:

Microsoft Online Services 계약

개발/테스트 가격 표시

예상 선불 비용

월별 예상 비용 ₩0.00 ₩16,476.12

내보내기 저장 다른 이름으로 저장 공유

한국 원(₩)

서버리스로 얻은 것



비용적인 측면

서비스 컴퓨팅 사용

The screenshot shows the Azure portal's bill summary page. It lists usage details for various services:

구분	설명	선불	월간
Azure Function	소비 계층, 128MB 메모리, 100밀리초 실행 시간, 10...	₩0.00	₩0.00
	소비 계층, 128MB 메모리, 100밀리초 실행 시간, 10...	₩0.00	₩0.00
	소비 계층, 128MB 메모리, 100밀리초 실행 시간, 10...	₩0.00	₩0.00
	소비 계층, 128MB 메모리, 100밀리초 실행 시간, 10...	₩0.00	₩0.00
	소비 계층, 128MB 메모리, 100밀리초 실행 시간, 10...	₩0.00	₩0.00
	소비 계층, 128MB 메모리, 100밀리초 실행 시간, 10...	₩0.00	₩0.00
	소비 계층, 128MB 메모리, 100밀리초 실행 시간, 10...	₩0.00	₩0.00
	소비 계층, 128MB 메모리, 100밀리초 실행 시간, 10...	₩0.00	₩0.00
	소비 계층, 128MB 메모리, 100밀리초 실행 시간, 10...	₩0.00	₩0.00
Azure Logic Apps	10,000개 작업 실행 x 1일, 100개 표준 커넥터 실행 ...	₩0.00	₩634.86

Below the table, there are dropdown menus for currency selection (including " 포함 항목" and "지원"), a "프로그램 및 제품" section showing "₩0.00" and "₩695.93", and a "예상 선불 비용" section with a dropdown menu set to "한국 원(₩)".

Azure Function X 9
스토리지 계정 X 1
로직 앱 X 1

서버리스로 얻은 것

비용적인 측면

가상 머신 사용



The screenshot shows a bill from a cloud provider. At the top, it says '가상 머신' (Virtual Machine) with a value of 1 A0(1 vCPU, 0.75GB RAM) x 1 월, Linux - Ubuntu... and a total monthly cost of ₩16,476.12. Below this, there's a section for '지원' (Support) which is free. Under '프로그램 및 제품' (Programs and Products), it lists 'Microsoft Online Services 계약' (Microsoft Online Services Contract) and '개발/테스트 가격 표시' (Development/Test Price Display). A large callout box highlights the monthly cost of ₩16,476.12.

가상 머신
(가장 저렴한 A0 플랜)

서버리스로 얻은 것



비용적인 측면

서비스 컴퓨팅 사용

A screenshot of the Azure portal showing the cost breakdown for various services. The total monthly bill is ₩634.86. The breakdown includes:

- Storage Accounts: ₩0.00
- Azure Function: ₩61.07
- Azure Logic Apps: ₩634.86

The currency dropdown at the bottom shows ₩695.93.

Azure Function X 9
스토리지 계정 X 1
로직 앱 X 1

가상 머신 사용

A screenshot of the Azure portal showing the cost breakdown for a virtual machine. The total monthly bill is ₩16,476.12. The breakdown includes:

- 가상 머신: ₩16,476.12

The currency dropdown at the bottom shows ₩16,476.12.

가상 머신
(가장 저렴한 A0 플랜)

서버리스로 얻은 것



비용적인 측면

서비스 컴퓨팅 사용

A screenshot of the Azure portal showing a list of services and their costs. The list includes Storage Accounts, Azure Functions, and Azure Logic Apps. Most items show a cost of ₩0.00, except for one Azure Function which shows ₩61.07. A yellow callout bubble points to this entry with the text "많은 비용 절감".

서비스	설명	선불: ₩0.00	월간: ₩61.07
Storage Accounts	블록 Blob Storage, 범용 V2, LRS 중복성, 핫 액세스...	₩0.00	₩61.07
Azure Function	소비 계층, 128MB 메모리, 100밀리초 실행 시간, 10...	₩0.00	₩0.00
Azure Function	소비 계층, 128MB 메모리, 100밀리초 실행 시간, 10...	₩0.00	₩0.00
Azure Function	소비 계층, 128MB 메모리, 100밀리초 실행 시간, 10...	₩0.00	₩0.00
Azure Function	소비 계층, 128MB 메모리, 100밀리초 실행 시간, 10...	₩0.00	₩0.00
Azure Function	소비 계층, 128MB 메모리, 100밀리초 실행 시간, 10...	₩0.00	₩0.00
Azure Function	소비 계층, 128MB 메모리, 100밀리초 실행 시간, 10...	₩0.00	₩0.00
Azure Function	소비 계층, 128MB 메모리, 100밀리초 실행 시간, 10...	₩0.00	₩0.00
Azure Function	소비 계층, 128MB 메모리, 100밀리초 실행 시간, 10...	₩0.00	₩0.00
Azure Function	소비 계층, 128MB 메모리, 100밀리초 실행 시간, 10...	₩0.00	₩0.00
Azure Logic Apps	10,000개 작업 실행 x 1월, 100개 표준 커넥터 실행 ...	₩0.00	₩634.40

Azure Function X 9
스토리지 계정 X 1
로직 앱 X 1

가상 머신 사용

A screenshot of the Azure portal showing a list of virtual machines. One entry for "A0(1 vCPU, 0.75GB RAM) x 1 월; Linux – Ubuntu..." shows a cost of ₩16,476.12. A yellow callout bubble points to this entry with the text "가상 머신 (가장 저렴한 A0 플랜)".

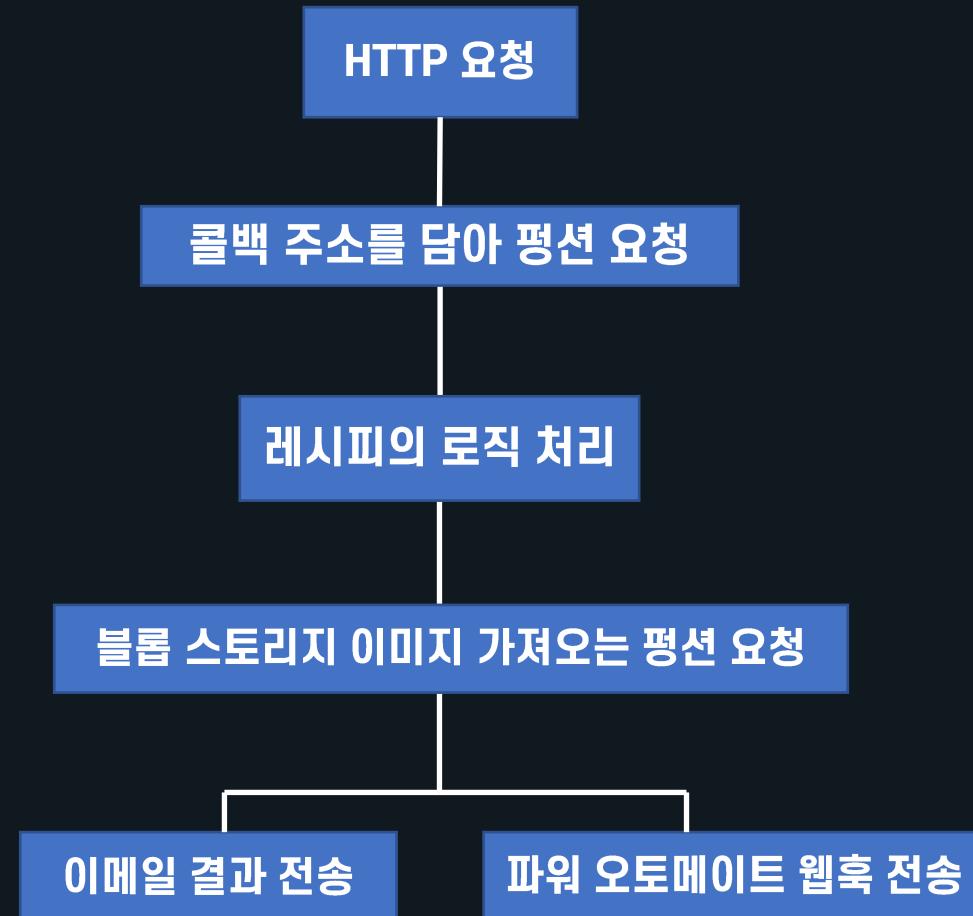
서비스	설명	선불: ₩0.00	월간: ₩16,476.12
가상 머신	1 A0(1 vCPU, 0.75GB RAM) x 1 월; Linux – Ubuntu...	₩0.00	₩16,476.12

많은 비용 절감

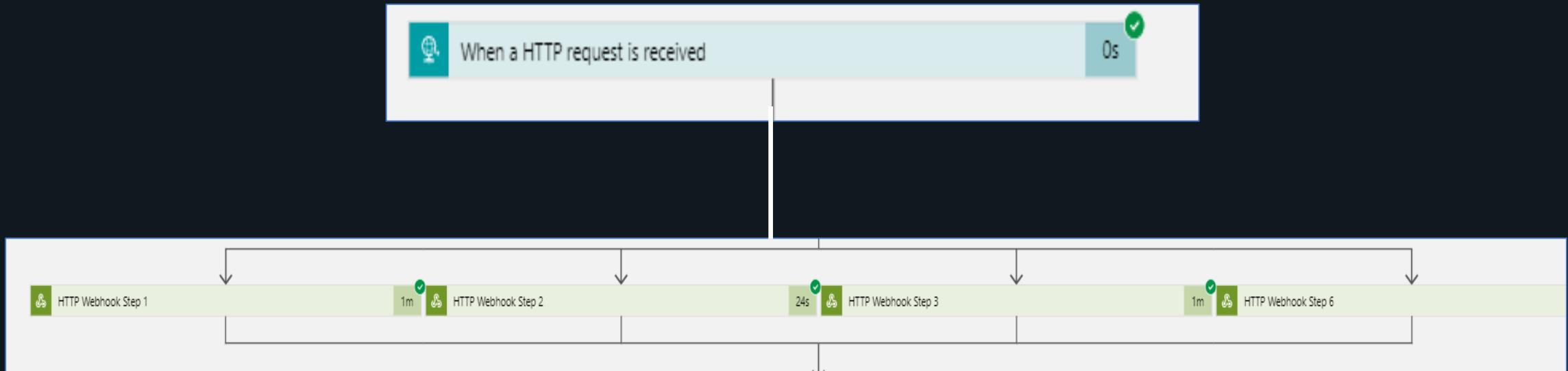
A0 플랜
1 vCPU
0.75 GB RAM

가상 머신
(가장 저렴한 A0 플랜)

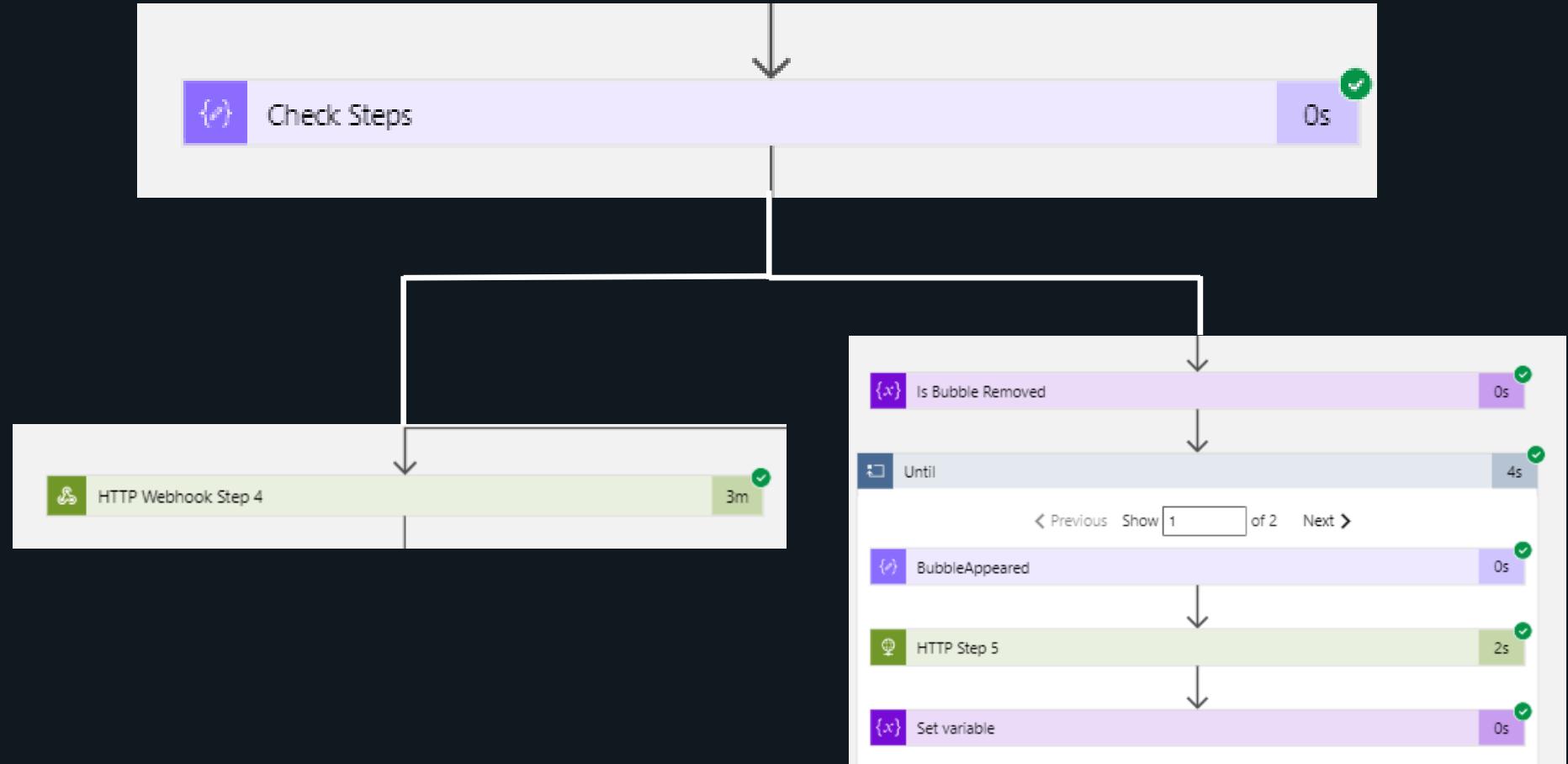
전체적인 구현 방식



전체적인 구현 방식

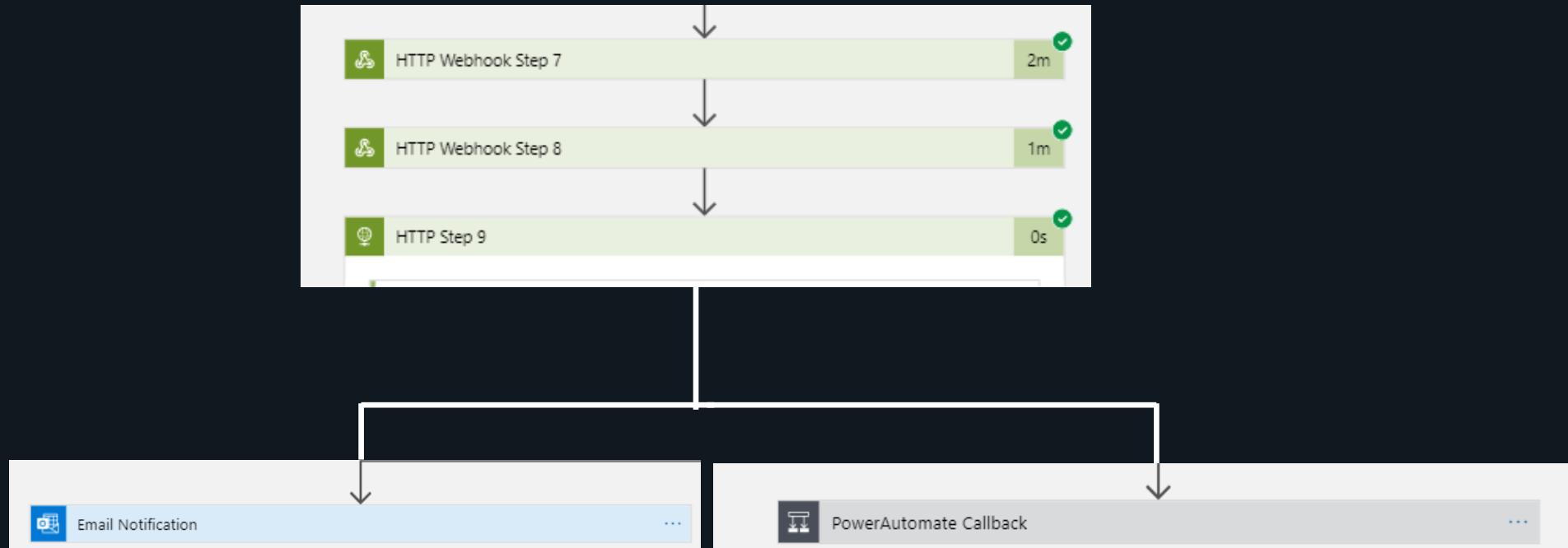


전체적인 구현 방식



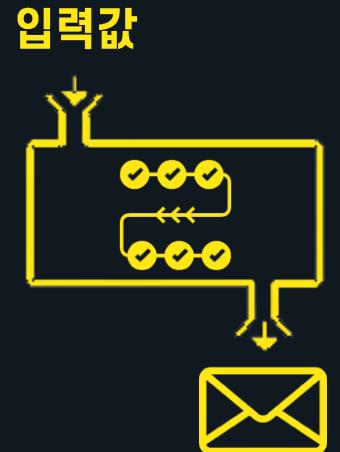


전체적인 구현 방식



SW 요구사항 분석

기능



제약조건



Azure

25

SW 요구사항 분석



Serverless Korea

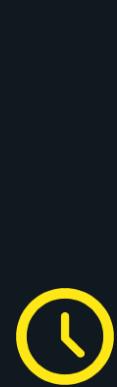
목표

요구하는 기능들을
애저 서비스 서비스를 이용하여
25일 내로 구현

시스템 명세



Serverless Korea



입력값

- 떡 불리기
- 파 썰기
- 계란을 풀고 휘젓기
- 재료를 끓이기



- 떡이 썰렸는지
- 후추 토픽 여부

- 이메일 주소



처리내용

떡국 레시피



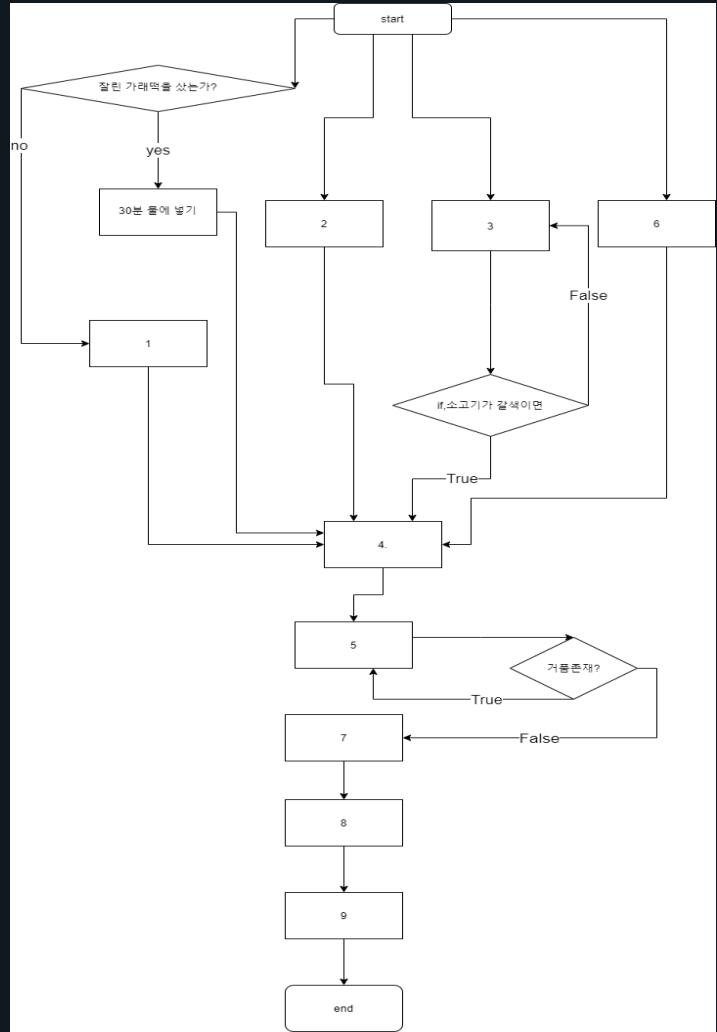
출력



⌚ 파워 오토메이트 만들기 #23 by justinyoo was closed on 23 Dec 2020
⌚ 파워 앱 만들기 #13 by justinyoo was closed on 23 Dec 2020
⌚ 레시피 스텝 9: 후추 뿌려 음식 내기 #12 by justinyoo was closed on 29 Nov 2020
⌚ 레시피 스텝 8: 계란 풀어 휘젓기 + 채 썬 파 넣기 #11 by justinyoo was closed on 24 Nov 2020
⌚ 레시피 스텝 7: 양념 추가 - 마늘, 간장, 소금 #10 by justinyoo was closed on 30 Nov 2020
⌚ 레시피 스텝 6: 계란 풀기 #9 by justinyoo was closed on 24 Nov 2020
⌚ 레시피 스텝 5: 끓는 동안 거품 제거 #8 by justinyoo was closed on 30 Nov 2020
⌚ 레시피 스텝 4: 재료 물에 넣고 끓이기 #7 by justinyoo was closed on 24 Nov 2020
⌚ 레시피 스텝 3: 소고기 볶기 #6 by justinyoo was closed on 22 Nov 2020
⌚ 레시피 스텝 2: 파 썰기 #5 by justinyoo was closed on 30 Nov 2020
⌚ 레시피 스텝 1: 가래떡 썰기 + 물에 불리기 #4 by justinyoo was closed on 22 Nov 2020
⌚ 전체 아키텍처 구성 #3 by justinyoo was closed on 30 Nov 2020
⌚ 순서도 작성 #1 by justinyoo was closed on 18 Nov 2020

그림 1. 프로젝트 이슈들

프로그램 설계

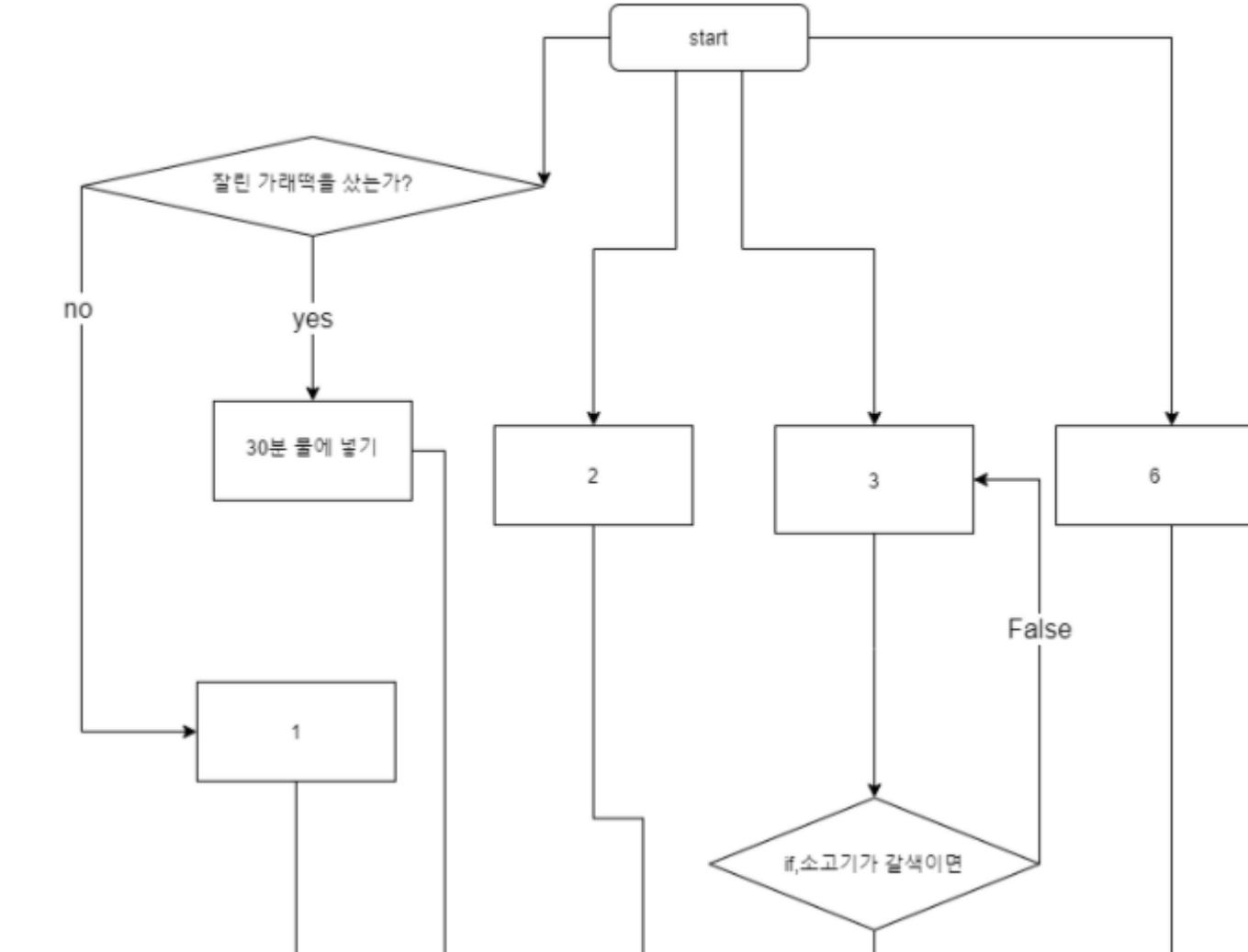


순서도

떡국 레시피의 절차에 따라 프로그램의 알고리즘을 작성

그림 1. 떡국 레시피 자동 프로세스 순서도

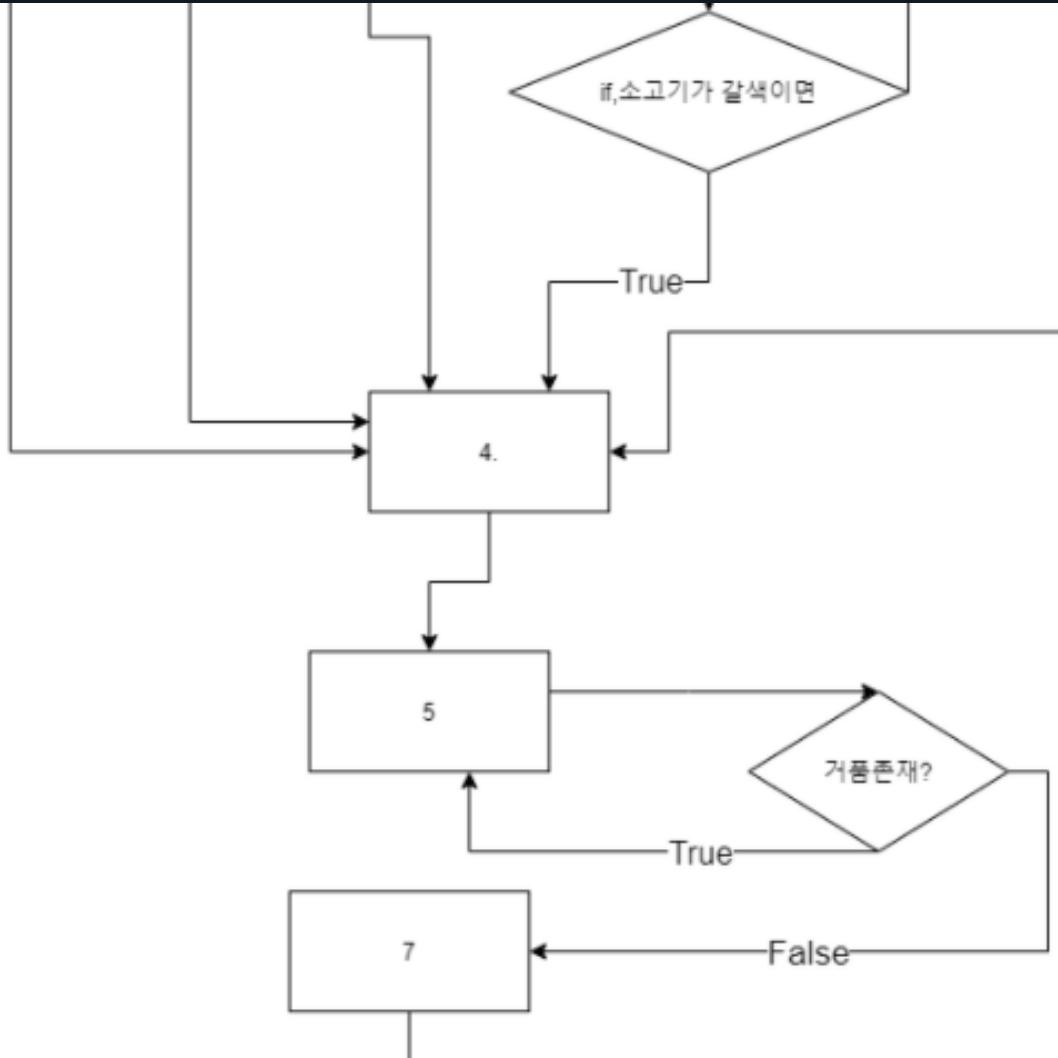
순서도 – 재료손질



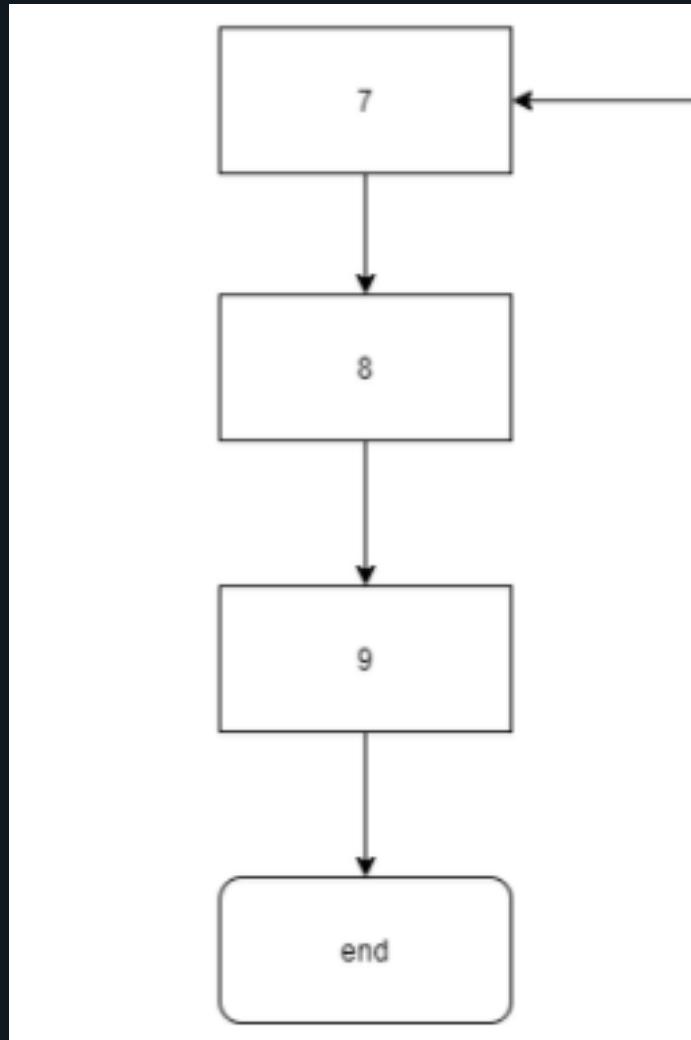


Serverless Korea

순서도 - 조리



순서도 - 마무리



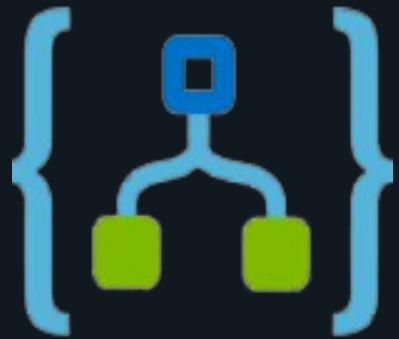
펑션앱



펑션앱

- 오케스트레이터 평션(Orchestrator Function)
- 액티비티 평션(Activity Function)
- 클라이언트 평션(Client Function)

로직앱



로직앱

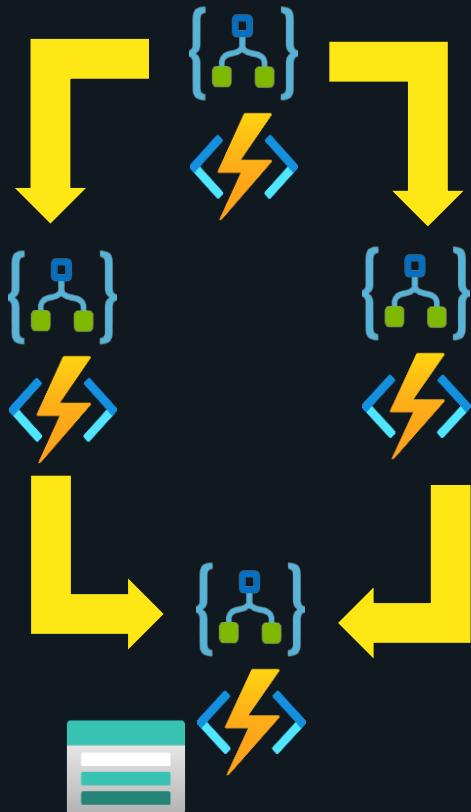
- 이벤트 발생, 트리거로 작동
- 작업 흐름(Work Flow)

웹훅 URL



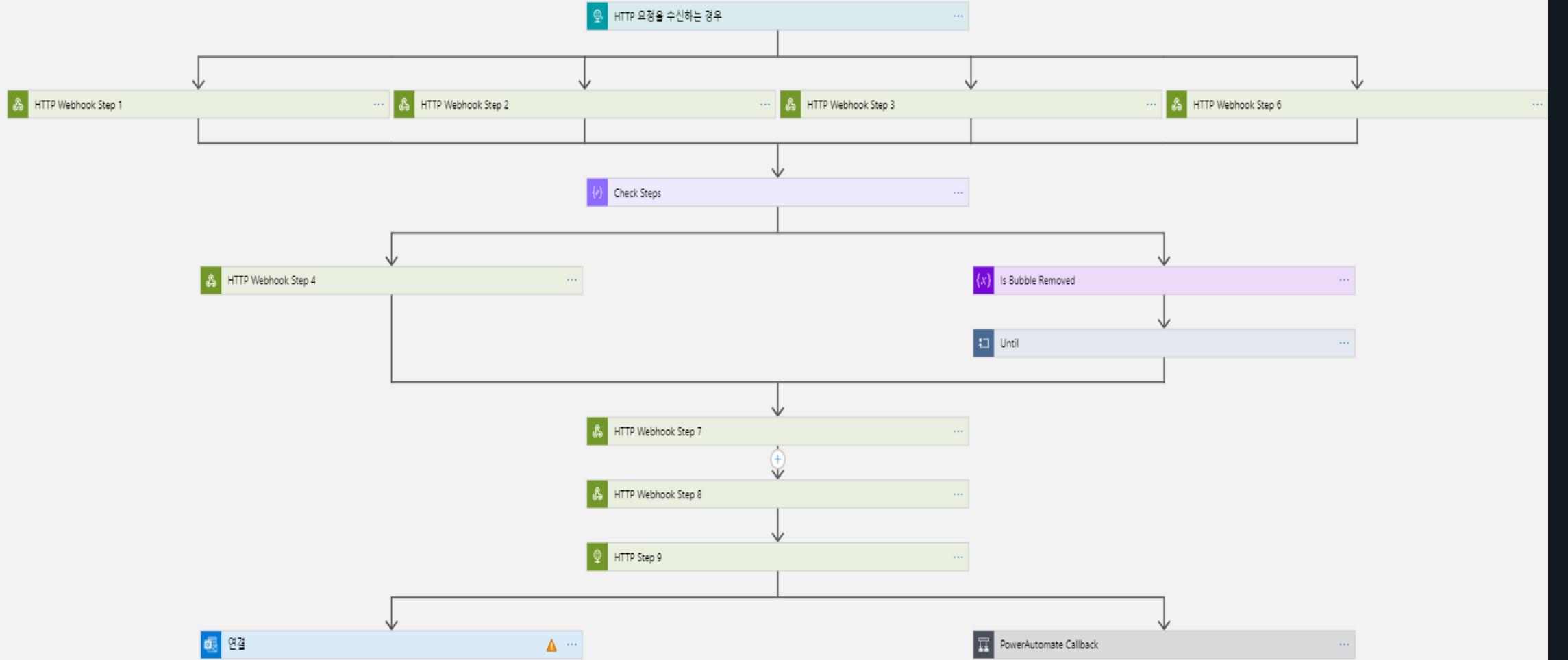
- 듀얼러블 평선의 시간 제한 문제(2분)
- 웹훅 URL을 이용

로직앱 작업 흐름 연결



- 알고리즘에 따른 로직앱 간의 유기적 연결
- 마지막 로직앱에서 블롭 저장소의 이미지 호출

로직앱 구성



로직앱 입력 값



Serverless Korea

The image displays three separate configurations for an 'HTTP Webhook Step' in a logic app, labeled Step 1, Step 2, and Step 3. Each step has a green header bar with the title and a white body containing configuration fields.

HTTP Webhook Step 1:

- 구독 - 메서드:** POST
- 구독 - URI:** [@] step01Endpoint
- 구독 - 본문:**

```
{
    "boughtSlicedGaraetteok": "@{boughtSlicedGaraetteok}",
    "callbackUrl": "fx listCallbackUrl()",
    "timeToSoakInMinutes": "@{timeToSoakInMinutes}"
}
```
- 구독 취소 - 메서드:** (empty)
- 구독 취소 - URI:** (empty)
- 구독 취소 - 본문:** (empty)
- Add new parameter:** (button)

HTTP Webhook Step 2:

- 구독 - 메서드:** POST
- 구독 - URI:** [@] step02Endpoint
- 구독 - 본문:**

```
{
    "callbackUrl": "fx listCallbackUrl()",
    "timeToSliceInMinutes": "@{timeToSliceInMinutes}"
}
```
- 구독 취소 - 메서드:** (empty)
- 구독 취소 - URI:** (empty)
- 구독 취소 - 본문:** (empty)
- Add new parameter:** (button)

HTTP Webhook Step 3:

- 구독 - 메서드:** POST
- 구독 - URI:** [@] step03Endpoint
- 구독 - 본문:**

```
{
    "callbackUrl": "fx listCallbackUrl()",
    "timeToStirFryInMinutes": "@{timeToStirFryInMinutes}"
}
```
- 구독 취소 - 메서드:** (empty)
- 구독 취소 - URI:** (empty)
- 구독 취소 - 본문:** (empty)
- Add new parameter:** (button)

파워앱 / 파워 오토메이트



파워앱 파워 오토메이트

- 파워앱에서 파워 오토메이트를 호출
- 파워 오토메이트에서 로직앱을 호출
- 최종적으로 파워 오토메이트에서 파워앱으로 노티

파워 앱



Serverless Korea

Time to soak in mins

Time to slice in mins

Time to stir-fry in mins

Time to boil in mins

youjin@studentambassadors.com

I bought sliced garae-tteok!

I want to add pepper!

Cook for me!

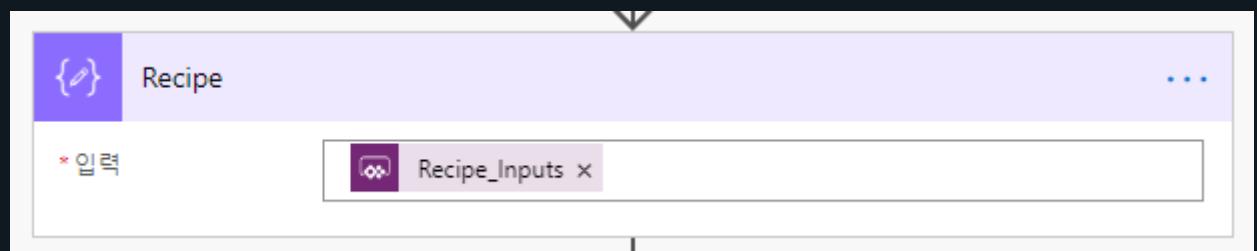
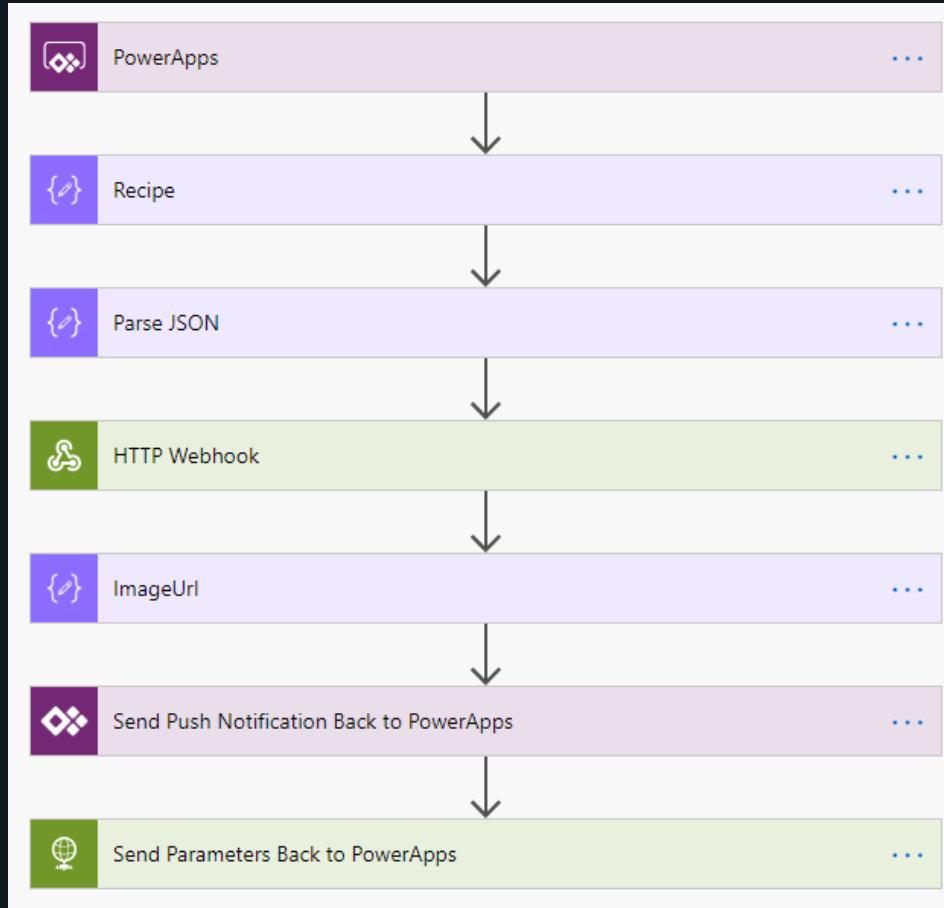


- 정수값 입력 필드 (시간)
- 문자열 입력 필드 (이메일)
- 불리언 체크박스 (불리언)
- 실행 버튼



Serverless Korea

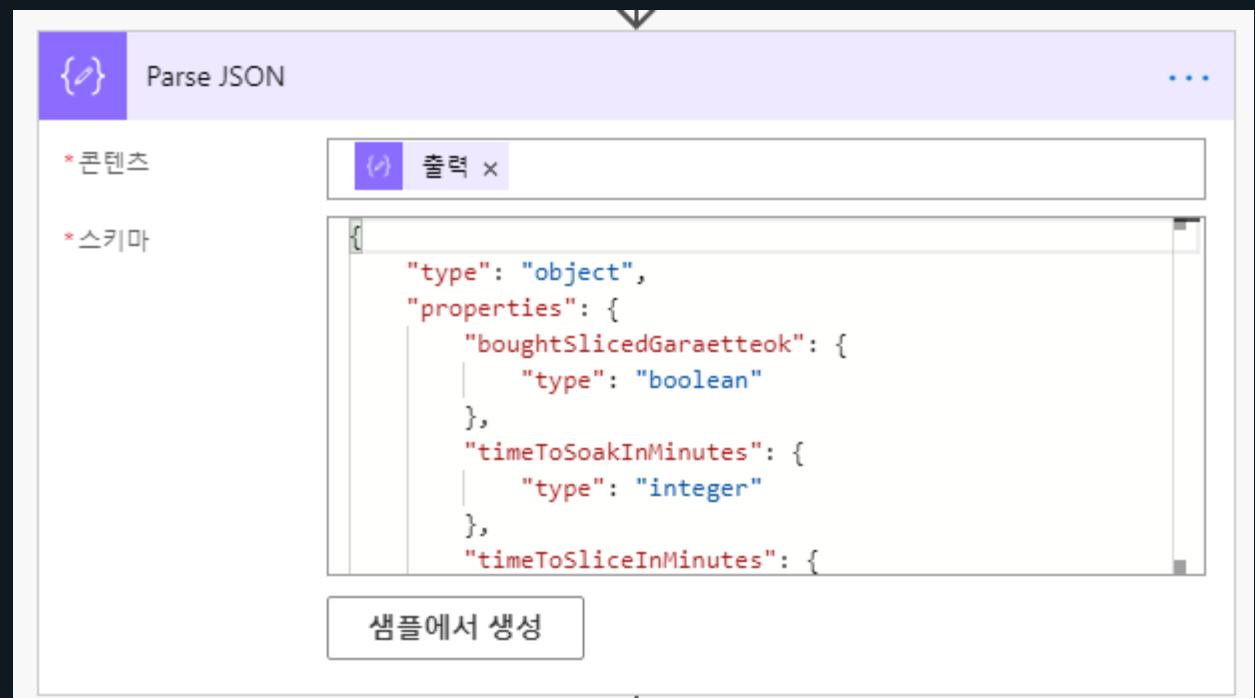
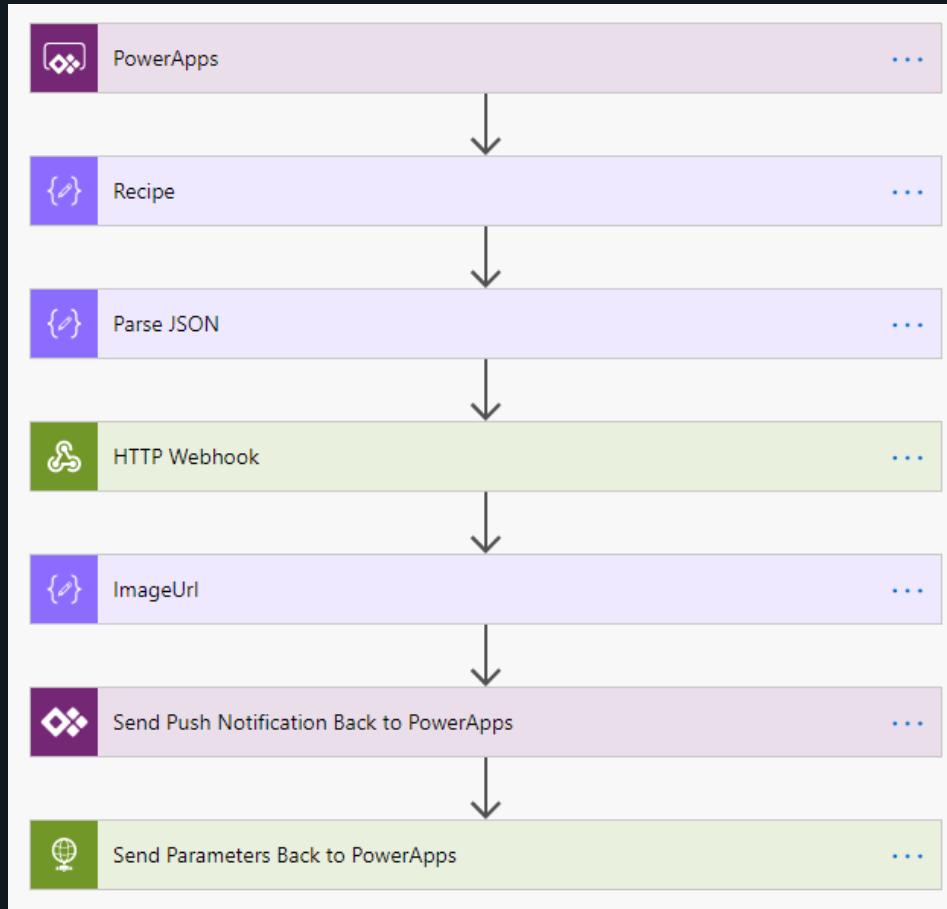
파워 오토메이트



파워오토메이트



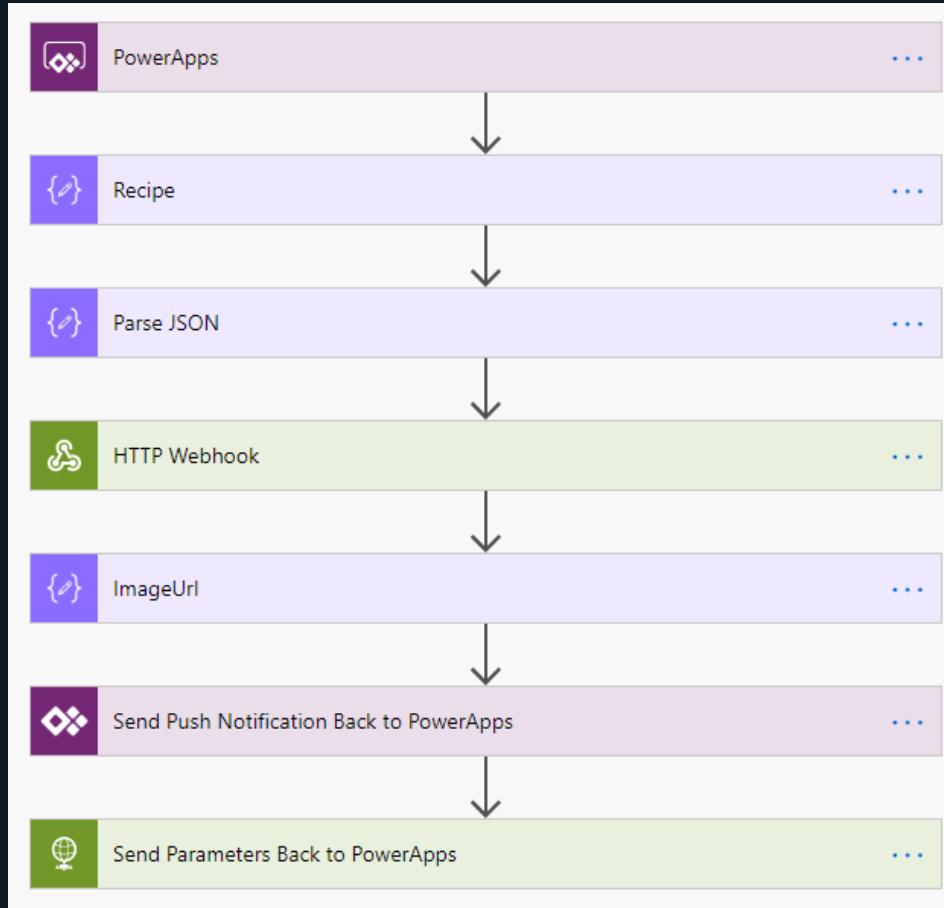
Serverless Korea



파워 오토메이트



Serverless Korea



HTTP Webhook

* 구독 - 메서드: POST

* 구독 - URI:
https://prod-71.eastus.logic.azure.com:443/workflows/037eeae7361f4ff8a7d12e783b0df01e/triggers/manual/paths/invoke?api-version=2016-10-01&sp=%2Ftriggers%2Fmanual%2Funrun&sv=1.0&sig=2B81E126-AiQ5MTZ36WKFP8Ax1TDPa0n6HHRI0khfJ4

구독 - 본문:

```
{
  "boughtSlicedGaraetteok": true,
  "timeToSoakInMinutes": 10,
  "timeToSliceInMinutes": 5,
  "timeToStirFryInMinutes": 3,
  "timeToBoilInMinutes": 2,
  "pepper": false,
  "email": "user@example.com",
  "callbackUrl": "listCallbackUrl()"
}
```

구독 취소 - 메서드:

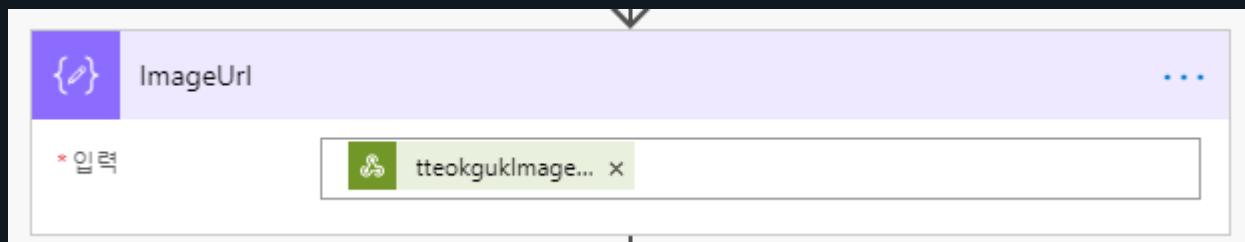
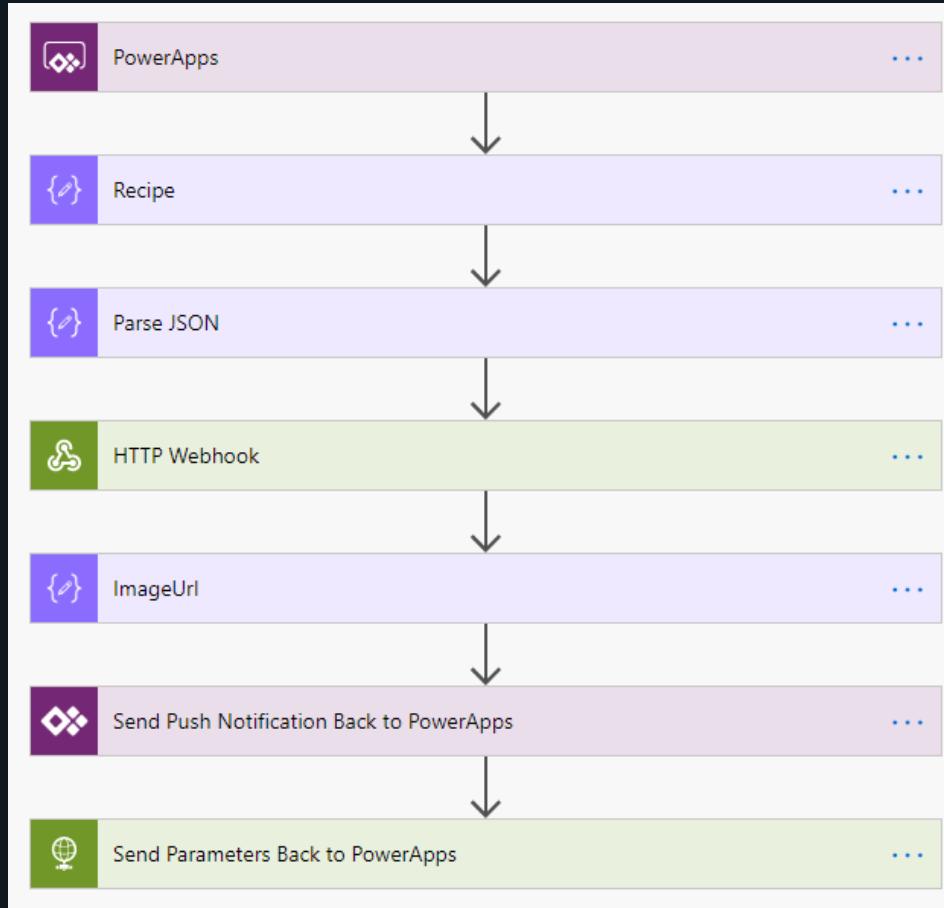
구독 취소 - URI:

구독 취소 - 본문:

고급 옵션 표시 ▾

This screenshot shows the configuration details for the 'HTTP Webhook' step in the Logic App. It specifies the subscription method as 'POST' and the subscription URI as a complex Azure Logic App URL. The payload is set to a JSON object containing various cooking parameters. There are also sections for unsubscribing and advanced options.

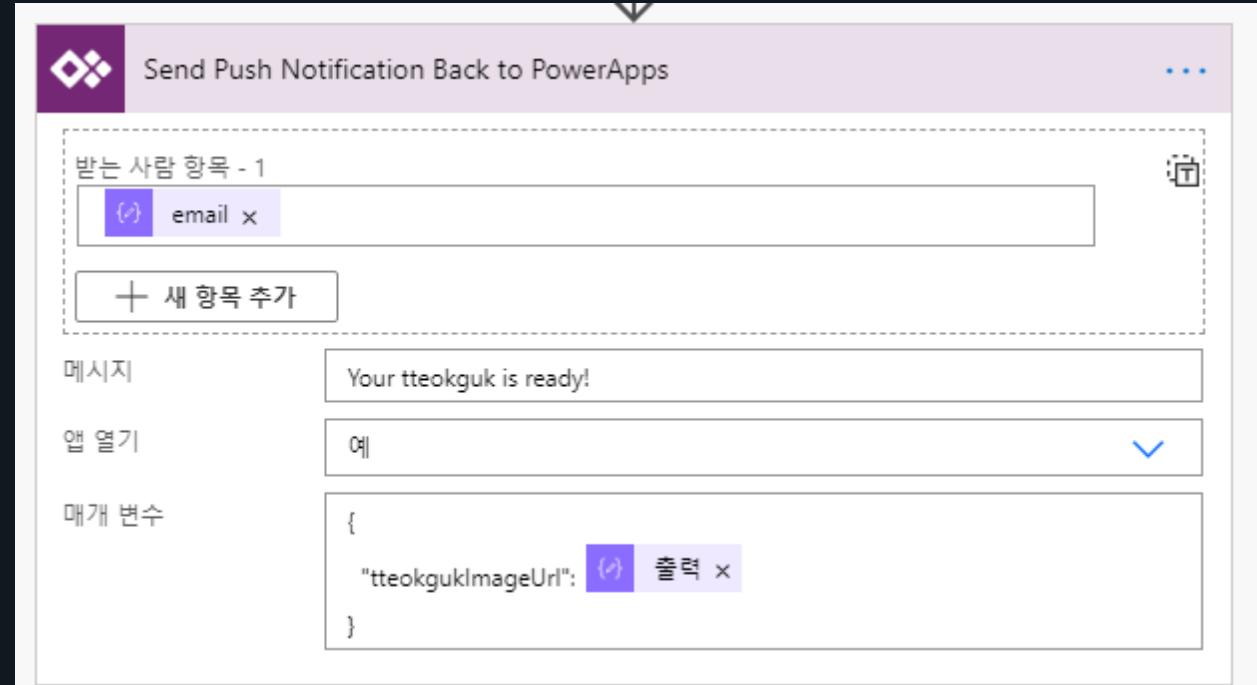
파워 오토메이트



파워 오토메이트



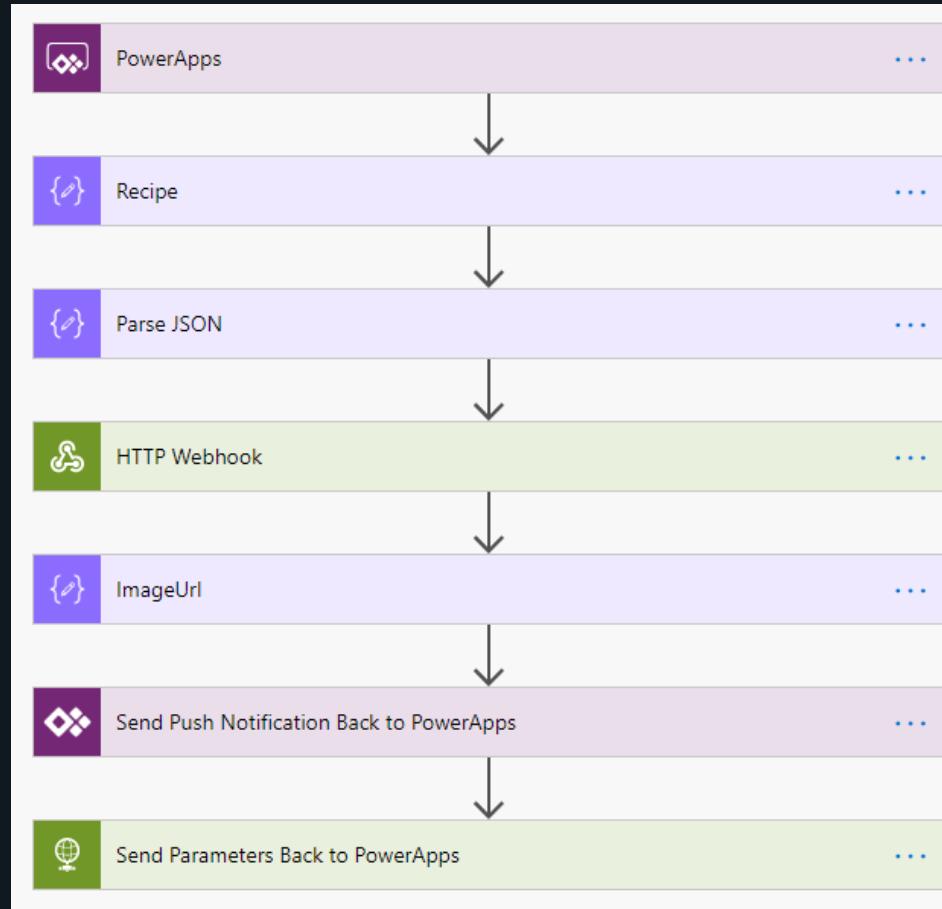
Serverless Korea



파워 오토메이트



Serverless Korea



Send Parameters Back to PowerApps

* 메서드: POST

* URI: <https://apps.powerapps.com/play/e01244c7-cb52-4f74-8ee3-9e61c91072de?tenantId=84c31ca0-ac3b-4eae-ad11-519d80233e6f&tteokgukImageUrl=>

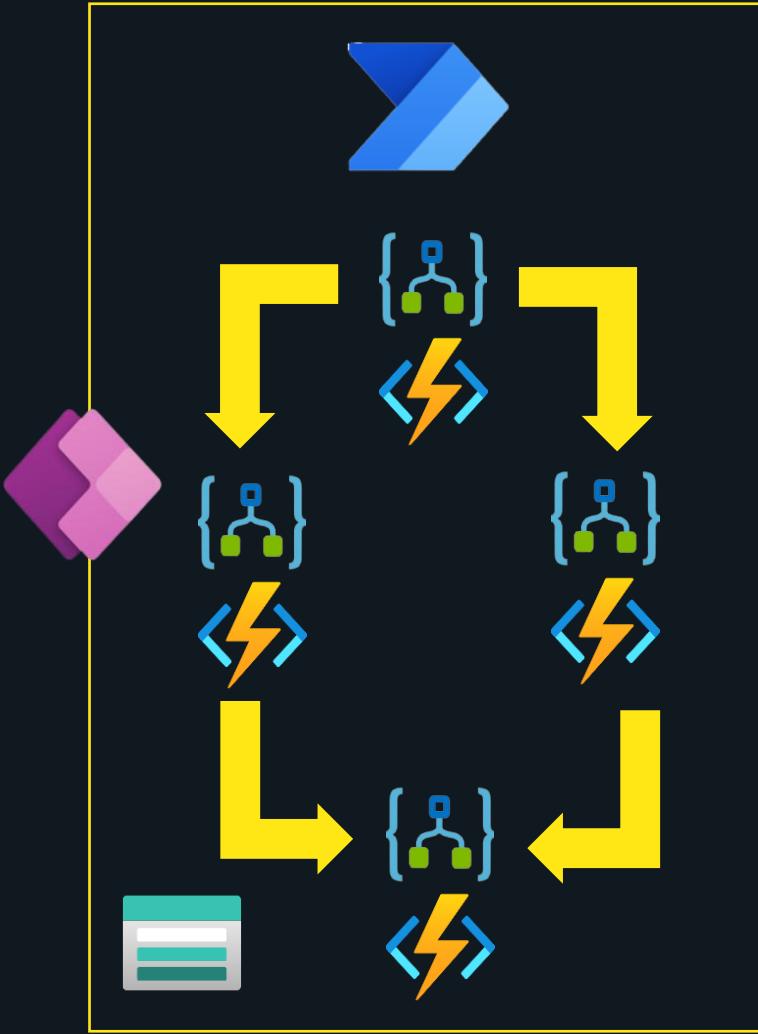
(x) 출력

헤더, 쿼리, 본문, 쿠키, 요청 콘텐츠 입력, HTTP 쿠키 입력

고급 옵션 표시 ▾

This screenshot shows the configuration for the 'Send Parameters Back to PowerApps' step. It is set to use a POST method and send parameters to a specific URI. The 'Output' field is currently empty. The step also includes sections for Headers, Query, Body, Cookies, Request Content Input, and HTTP Cookie Input. A 'Show Advanced Options' button is visible at the bottom.

시스템 구조 설계



파이썬의 클라이언트 평선

```
async def main(req: func.HttpRequest, starter: str) -> func.HttpResponse:  
    client = df.DurableOrchestrationClient(starter)  
  
    instance_id = await client.start_new(req.route_params["functionName"], None, client_input=req.get_json())  
  
    return client.create_check_status_response(req, instance_id)
```

- 오케스트레이터 평선 트리거
- 임의의 인스턴스 ID 생성
- 상태값 반환

파이썬의 오케스트레이터 평션

```
while not is_fried:  
    time = context.current_utc_datetime  
    if is_first:  
        # First time to fry  
        delay_time = time + timedelta(minutes=TIME_TO_STIR)  
        is_first = False  
  
    else:  
        # Not fried yet  
        delay_time = time + timedelta(minutes=1)  
  
    yield context.create_timer(delay_time)  
    result = yield context.call_activity('Fry', CALLBACK_URL)  
  
    is_fried = result
```

- 현재 시간 값 + timedelta = 타이머 시간
- 작업 흐름에 따른 조건 분기
- 액티비티 평션 호출 및 결과 값을 변수에 할당

파이썬의 액티비티 평선

```
def main(req: func.HttpRequest) -> dict:
    bool_data = [True, False]
    pick_data = choice(bool_data)

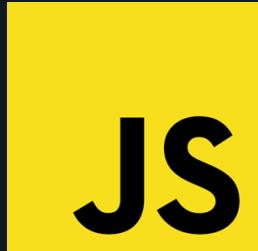
    if pick_data:
        # Must execute when pick_data is True
        logging.info(req)

        requests.post(
            req,
            headers={"Content-Type": "application/json"},
            data=json.dumps({
                "completed": pick_data
            })
        )

    return pick_data
```

- 임의로 정해지는 불리언 값
- 조건이 충족되면 웹훅 URL 반환

폴리글랏(Polyglot)



```
module.exports = async function (context, req) {  
    const client = df.getClient(context);
```



```
async def main(req: func.HttpRequest, starter: str) -> func.HttpResponse:  
    client = df.DurableOrchestrationClient(starter)
```



```
public static class Step2 {  
    private static HttpClient httpClient = new HttpClient();
```

실제 결과

Time to soak in mins

Time to slice in mins

Time to stir-fry in mins

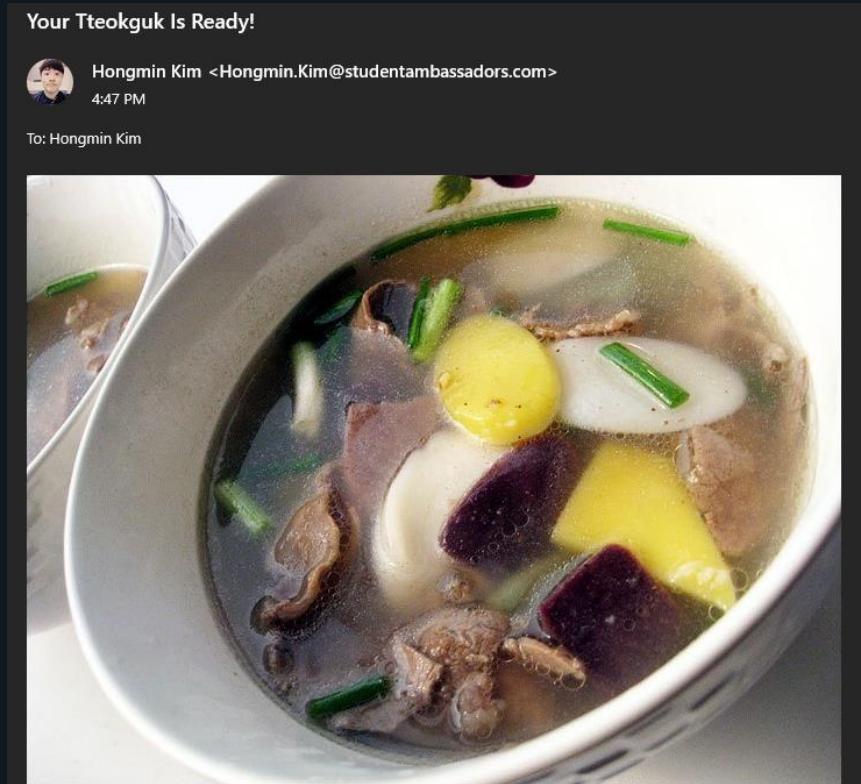
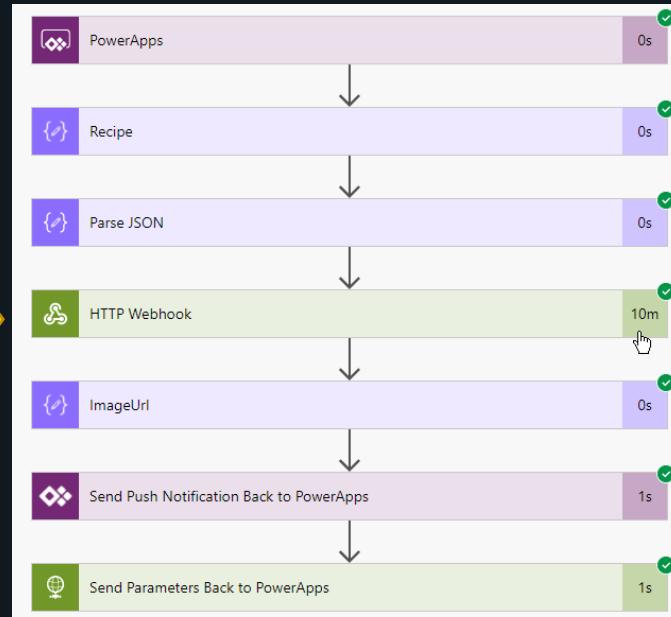
Time to boil in mins

Hongmin.Kim@studentambassadors.com

I bought sliced garae-tteok!

I want to add pepper!

Cook for me!



파워 앱

파워 오토메이트 -> 로직 앱

결과



챌린지 - TMI



TMI (Too Many Items)

애저 평션 vs 애저 로직 앱

애저 평션

- 애저 앱 서비스를 통해 호스팅
- 이벤트 발생 시 코드가 동작
- 바인딩을 통해 다른 서비스 등과 통합
- 코드를 통해 작성
- 다양한 배포 방법

로직 앱

- 애저 포탈에서 서비스
- 이벤트 발생 시 해당 워크플로우가 동작
- 커넥터를 통해 다른 서비스 등과 통합
- 디자이너를 통해 작성
- ARM 템플릿을 통해 배포

활용방안 (로직 앱 + 애저 평션)

- 1) 로직 앱을 통해 비즈니스 프로세스 설계
- 2) 애저 평션을 통해 각 흐름의 코드 구현
- 3) 로직 앱을 통해 2에 대한 오케스트레이션
- 4) 애저 포탈을 통해 배포

데모

대시보드 > 새로 만들기 >

Marketplace

프라이빗 Marketplace(미리 보기) X

Favorites

최근에 만들어짐

서비스 공급자

범주

시작

AI + 기계 학습

분석

블록체인

컴퓨팅

컨테이너

데이터베이스

개발자 도구

DevOps

ID

트렌드

로직 앱 X

가격 책정 : 모두 X

운영 체제 : 모두 X

Publisher Type : 모두 X

Offer Type : 모두 X

Publisher name : All X

모든 결과 표시

타일 보기 ▼

 노트 앱
Microsoft Azure Service
코딩 작업 없이 자동으로 클라우드 내 데이터를 액세스 및 사용하세요.
[만들기](#) ♡

대시보드 > ServerlessDemo1 >

논리 앱 디자이너

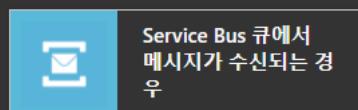


통합 솔루션 구축이 그 어느 때보다 쉬워졌습니다.
Logic Apps은 엔터프라이즈 통합 공간의 속도와 확장성을 향상시킵니다. 디자이너의 사용 편의성, 사용 가능한 트리거 및 작업의 다양성, 강력한 관리 도구로 인해 API를 그 어느 때보다 간편하게 중앙 집중화할 수 있습니다. 비즈니스가 디지털화됨에 따라 Logic Apps을 통해 레거시 및 최첨단 시스템을 함께 연결할 수 있습니다.

- 비즈니스 프로세스 및 워크플로를 시작적으로 만들기
- SaaS 및 엔터프라이즈 애플리케이션과 통합
- 온-프레미스 및 클라우드 애플리케이션에서 값을 잠금 해제

일반적인 트리거로 시작

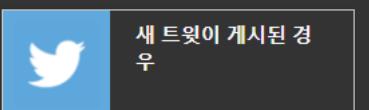
가장 일반적으로 사용되는 트리거 중 하나를 선택한 다음, 다양한 커넥터 컬렉션을 사용하여 수많은 작업을 오케스트레이션합니다.



Service Bus 큐에서
메시지가 수신되는 경
우



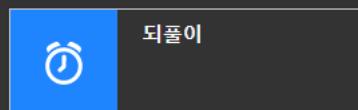
HTTP 요청을 수신하
는 경우



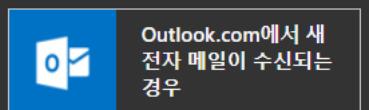
새 트윗이 게시된 경
우



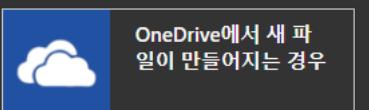
Event Grid 리소스 이
벤트가 발생하는 경우



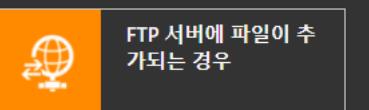
되풀이



Outlook.com에서 새
전자 메일이 수신되는
경우



OneDrive에서 새 파일이
만들어지는 경우



FTP 서버에 파일이 추
가되는 경우

대시보드 > ServerlessDemo1

ServerlessDemo1 | 논리 앱 디자이너

논리 앱

검색(Ctrl+ /) ⌘ 저장 ✖ 삭제 ⌘ 실행 ⌘ 디자이너 ⌘ 코드 보기 ⌘ 매개 변수 ⌘ 템플릿 ⌘ 커넥터 ? 도움말 ⓘ 정보

개요

활동 로그

액세스 제어(IAM)

태그

문제 진단 및 해결

개발 도구

논리 앱 디자이너

</> 논리 앱 코드 보기

버전

API 연결

빠른 시작 가이드

설정

워크플로 설정

권한 부여

액세스 키

Identity

속성

잡금

When a new email arrives

폴더: Inbox

중요도: Normal

얼마나 자주 항목을 확인하시겠습니까?

1 분

Add new parameter

Hongmin.Kim@studentambassadors.com에 연결되었습니다. 연결을 변경합니다.

+새 단계



Serverless Korea

대시보드 > ServerlessDemo1

ServerlessDemo1 | 논리 앱 디자이너

논리 앱

검색(Ctrl+ /) 저장 삭제 실행 디자이너 코드 보기 매개 변수 템플릿 커넥터 도움말 정보

기요 활동 로그 액세스 제어(IAM) 태그 문제 진단 및 해결

개발 도구 논리 앱 디자이너 </> 논리 앱 코드 보기 버전 API 연결 빠른 시작 가이드

설정 워크플로 설정 권한 부여 액세스 키 Identity 속성 잠금 모니터링 경고 메트릭

When a new email arrives

작업 선택

커넥터 및 작업 검색

사용자용 모두 기본 제공 표준 엔터프라이즈 사용자 지정

컨트롤 HTTP 인라인 코드 Azure Functions 날짜 시간 데이터 작업 변수

트리거 작업

- 응답 요청
- 과거 시간 가져오기 날짜 시간
- 미래 시간 가져오기 날짜 시간
- 시간에 추가 날짜 시간
- 시간에서 빼기

대시보드 > ServerlessDemo1

ServerlessDemo1 | 논리 앱 디자이너

노리 앱

검색(Ctrl+ /)

저장 ✎ 삭제 ⌘ 실행 ⌘ 디자이너 ⌘ 코드 보기 ⌘ 매개 변수 ⌘ 템플릿 ⌘ 커넥터 ? 도움말 ⓘ 정보

개요

활동 로그

액세스 제어(IAM)

태그

문제 진단 및 해결

개발 도구

논리 앱 디자이너

논리 앱 코드 보기

버전

API 연결

빠른 시작 가이드

설정

워크플로 설정

권한 부여

액세스 키

Identity

속성

잠금

When a new email arrives

작업 선택

커넥터 및 작업 검색

serverlessdemo1



Serverless Korea

대시보드 > ServerlessDemo1

ServerlessDemo1 | 논리 앱 디자이너

검색(Ctrl+ /) <> 저장 ✎ 삭제 ▶ 실행 ⚡ 디자이너 </> 코드 보기 [+] 매개 변수 템플릿 커넥터 ? 도움말 ⓘ 정보

개요
활동 로그
액세스 제어(IAM)
태그
문제 진단 및 해결

개발 도구
논리 앱 디자이너 </> 논리 앱 코드 보기
버전
API 연결
빠른 시작 가이드

설정
워크플로 설정
권한 부여
액세스 키
Identity
속성
잠금

When a new email arrives

HttpTrigger1

요청 본문
보낸 사람

Add new parameter

+ 새 단계

100%

A screenshot of the Azure Logic Apps Designer interface. On the left, there's a sidebar with navigation links like 'Serverless Demo1', '검색(Ctrl+ /)', '저장', '삭제', '실행', '디자이너', '코드 보기', '매개 변수', '템플릿', '커넥터', '도움말', and '정보'. Below that is a '개발 도구' section with '논리 앱 디자이너' selected, followed by '논리 앱 코드 보기', '버전', 'API 연결', and '빠른 시작 가이드'. Under '설정', there are links for '워크플로 설정', '권한 부여', '액세스 키', 'Identity', '속성', and '잠금'. The main workspace shows a workflow: 'When a new email arrives' (trigger) connected to 'HttpTrigger1' (action). The 'HttpTrigger1' step has a '요청 본문' input field containing '보낸 사람' (sent to). There's also a 'Add new parameter' button and a '+ 새 단계' (New Step) button at the bottom right of the workspace. The top right corner shows a zoom level of '100%'.



Serverless Korea

대시보드 > ServerlessDemo1

ServerlessDemo1 | 논리 앱 디자이너

논리 앱

검색(Ctrl+ /)

저장 삭제 실행 디자이너 코드 보기 매개 변수 템플릿 커넥터 도움말 정보

개요 활동 로그 액세스 제어(IAM) 태그 문제 진단 및 해결

개발 도구 논리 앱 디자이너

논리 앱 코드 보기 버전 API 연결 빠른 시작 가이드

설정 워크플로 설정 권한 부여 액세스 키 Identity 속성 잠금

When a new email arrives

HttpTrigger1

메시지 게시(V3) (미리 보기)

*팀 DevOps_Team
*채널 General
*메시지 글꼴 12pt B I 편집 모드
본문

Add new parameter

+새 단계

100%

The screenshot shows the Azure Logic Apps Designer interface for the 'ServerlessDemo1' logic app. On the left, there's a navigation sidebar with links like '대시보드', 'ServerlessDemo1', '논리 앱', '검색(Ctrl+ /)', '저장', '삭제', '실행', '디자이너', '코드 보기', '매개 변수', '템플릿', '커넥터', '도움말', and '정보'. The main area displays a workflow with three main components: 'When a new email arrives', 'HttpTrigger1', and '메시지 게시(V3) (미리 보기)'. The 'HttpTrigger1' component has its parameters set to '팀: DevOps_Team', '채널: General', and '메시지: 글꼴 12pt B I 편집 모드 본문'. Below the 'HttpTrigger1' action, there's a note: '연결되었습니다. 연결을 변경합니다.' (The connection has been established. You can change the connection). At the bottom right of the designer, there's a button labeled '+새 단계' (New Step).



Serverless Korea

서식 삽입 그리기 옵션

! ↓ | 한국어 | 맞춤법 검사 | 사용권한 | 확대/축소 | 찾기

보낸 사람: Hongmin.Kim@studentambassadors.com

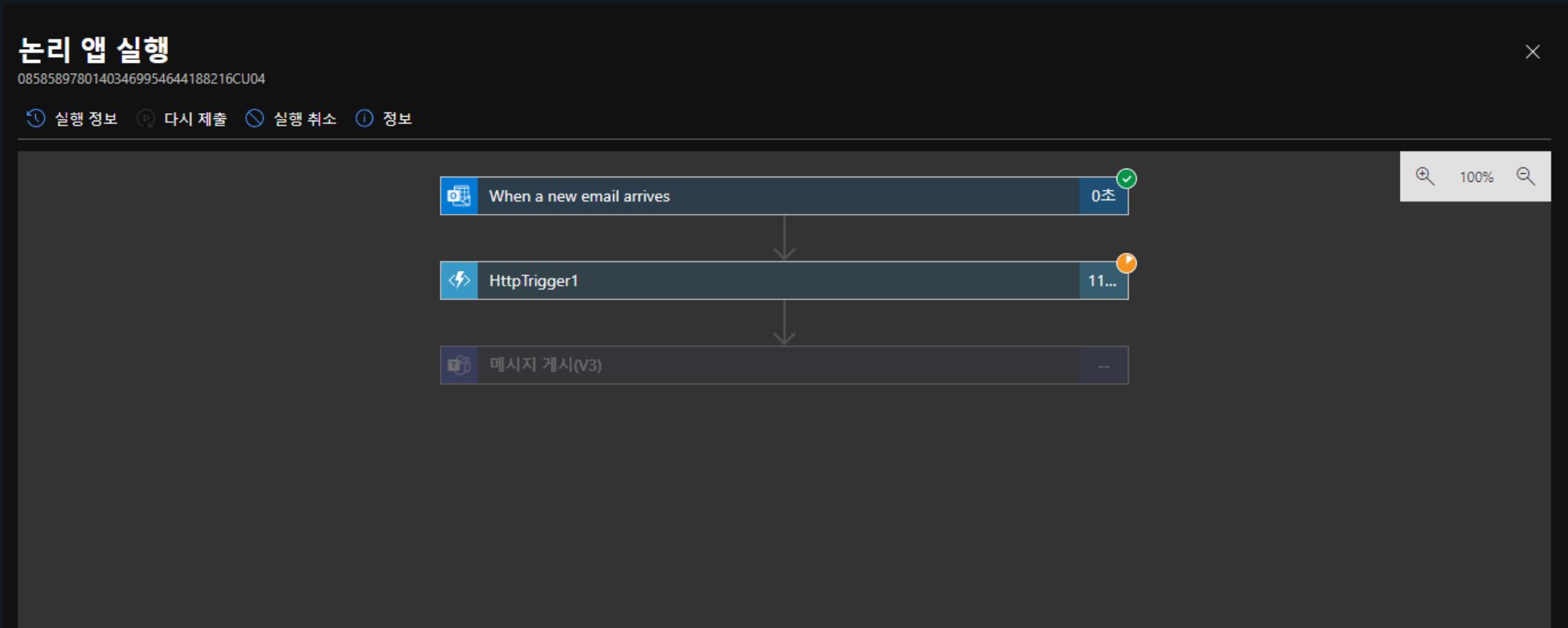
받는 사람: 김홍민;

긴급

긴급|



Serverless Korea





Serverless Korea

검색

팀

활동

채팅 1

팀

일정

통화

파일

...

앱

도움말

참가 또는 팀 만들기

새 대화

팀

일반 게시물 파일 2개 더 + 조작 ⌂ ⓘ ...

다른 사람 추가 추가 채널 만들기 FAQ 열기 2020년 10월 16일

김홍민 20. 10. 16. 오후 2:39
이 채널 상단에 새 탭이 추가되었습니다. 여기에 링크가 있습니다.

↪ 회신

오늘

김홍민 오후 6:51
긴급

↪ 회신

새 대화

로직 앱 vs 파워 오토메이트



Serverless Korea

로직 앱

- 애저 포탈을 통해 사용 가능
- 다양한 SaaS 비즈니스에 연계
- ARM 템플릿을 통해 배포

파워 오토메이트

- Microsoft 365 라이선스를 통해 사용 가능
- Microsoft 365 비즈니스 연계에 친화
- 페이지 내의 다양한 템플릿을 통해 조합 및 배포

활용방안



Serverless Korea

- 1) 파워 앱을 통해 비즈니스 UI 설계
- 2) 파워 오토메이트를 통해 파워 앱 연계
- 3) 로직 앱을 통해 내부 비즈니스 워크플로우 설계
- 4) 애저 평션을 통해 개별 코드 작성
- 5) 파워 앱 배포

감사합니다