

Azure Logic Apps과 ACI로 대규모 작업

목차

EXERCISE. LOGIC APPS와 AZURE 컨테이너 인스턴스 통합.....	2
개요.....	2
TASK 01. 스토리지 계정 만들기	2
TASK 02. LOGIC APPS 만들기 및 FILES 커넥터 구성	5
TASK 03. AZURE 컨테이너 인스턴스 작업 구성	8
TASK 04. 작업이 완료된 컨테이너 인스턴스 삭제하기.....	14
TASK 05. 작업이 완료된 파일 삭제	17
TASK 06. LOGIC APPS 코드 수정.....	18
TASK 07. LOGIC APPS 실행 및 확인	20

EXERCISE. Logic Apps와 Azure 컨테이너 인스턴스 통합

개요

이 실습에서는 다음과 같은 작업을 Logic Apps와 Azure Container Instance를 통해 구현합니다.

- Azure Files 스토리지에 소스 동영상 파일을 업로드합니다.
- Logic Apps에서 일정 간격으로 Azure Files 스토리지에 새로 업로드된 파일을 확인합니다.
- 업로드된 동영상 파일 수에 따라 Azure 컨테이너 인스턴스를 실행합니다.
- Azure 컨테이너 인스턴스는 소스 동영상 파일을 GIF 형식으로 변환한 후 변환된 파일을 Azure Files 스토리지의 다른 폴더에 저장합니다.
- 동영상 변환이 완료되면 Azure Files 스토리지의 소스 동영상 파일을 삭제합니다.

이러한 워크플로 작업을 Logic Apps와 Azure 컨테이너 인스턴스를 사용하여 No-code 환경으로 구성합니다.

TASK 01. 스토리지 계정 만들기

먼저 Azure 컨테이너 인스턴스의 볼륨으로 탑재하여 작업을 진행할 Azure Files 공유를 만들기 위해 스토리지 계정을 만들고 준비합니다.

1. Azure 포털의 검색창에서 "스토리지 계정"을 검색한 후 클릭합니다. [스토리지 계정] 블레이드의 메뉴에서 [만들기]를 클릭합니다.



2. [저장소 계정 만들기] 블레이드의 [기본 사항] 탭에서 아래와 같이 구성한 후 [검토 + 만들기]를 클릭합니다. [검토 + 만들기] 탭에서 [만들기]를 클릭합니다.

- [프로젝트 정보 - 리소스 그룹]: "새로 만들기"를 클릭한 후 "02_logicAppsAciRg" 이름을 입력합니다.
- [인스턴스 정보 - 스토리지 계정 이름]: 중복되지 않는 고유한 이름을 입력합니다.
- [인스턴스 정보 - 지역]: (Asia Pacific) Korea Central
- [인스턴스 정보 - 기본 서비스]: 선택하지 않습니다.
- [인스턴스 정보 - 성능]: 표준
- [인스턴스 정보 - 중복]: LRS(로컬 중복 스토리지)

저장소 계정 만들기

기본 사항 고급 네트워크 데이터 보호 암호화 태그 검토 + 만들기

Azure Storage는 가용성, 보안, 내구성, 확장성 및 중복성이 뛰어난 클라우드 스토리지를 제공하는 Microsoft 관리 서비스입니다. Azure Storage는 Azure Blob(개체), Azure Data Lake Storage Gen2, Azure Files, Azure 큐 및 Azure 테이블을 포함합니다. 스토리지 계정의 비용은 사용량 및 아래에서 선택한 옵션에 따라 다릅니다. [Azure Storage 계정에 대한 자세한 정보](#)

프로젝트 정보
새 스토리지 계정을 만들 구독을 선택합니다. 다른 리소스와 함께 스토리지 계정을 구성하고 관리할 새 리소스 그룹 또는 기존 리소스 그룹을 선택합니다.

구독 * ME-MngEnvMCAPI195610-jhwoo-1
리소스 그룹 * (신규) 02_logicAppsAciRg
[새로 만들기](#)

인스턴스 정보
스토리지 계정 이름 * kormtt1031files
지역 * (Asia Pacific) Korea Central
[Azure 확장 영역에 배포](#)
기본 서비스 * 기본 서비스 선택
성능 * ☒ 표준: 대부분 시나리오에 권장됨(범용 v2 계정)
☐ 프리미엄: 짧은 대기 시간이 필요한 경우에 권장됩니다.
중복 * LRS(로컬 중복 스토리지)

3. 새로 만든 [스토리지 계정] 블레이드의 [데이터 스토리지 - 파일 공유]로 이동한 후 메뉴에서 [파일 공유]를 클릭합니다.

kormtt1031files | 파일 공유

스토리지 계정

검색 < + 파일 공유 새로 고침 피드백 제공

파일 공유 설정
ID 기반 액세스: 구성되지 않음 기본 공유 수준 권한: 비활성화 일시 삭제: 7일 최대 용량: 100 TiB
보안: 최대 호환성
접두사를 기준으로 파일 공유 검색(대소문자 구분) ☐ 삭제된 공유 표시

이름	수정됨	액세스 계층	할당량
파일 공유가 없습니다. 시작하려면 [+ 파일 공유]를 클릭하세요.			

4. [새 파일 공유] 블레이드의 [기본 사항] 탭에서 아래와 같이 구성한 후 [다음]을 클릭합니다.

- 이름: videos
- 액세스 계층: 트랜잭션 최적화됨

새 파일 공유

기본 사항 백업 검토 + 만들기

이름 * videos
액세스 계층 * 트랜잭션 최적화됨

성능
최대 IO/초 ① 20000
최대 용량 100 TiB

① 이 공유와 함께 SMB 프로토콜을 사용하려면 포트 445를 통해 통신할 수 있는지 확인하세요. 다음에 대한 이러한 스크립트는 [Windows 클라이언트](#) 및 [Linux 클라이언트](#) 도움이 될 수 있습니다. 방법 알아보기 [포트 445 문제를 피해 갑니다](#).

5. [백업] 탭에서 "백업 사용" 옵션을 선택하지 않고 [다음]을 클릭합니다. [검토 + 만들기] 탭에서 [만들기]를 클릭합니다.

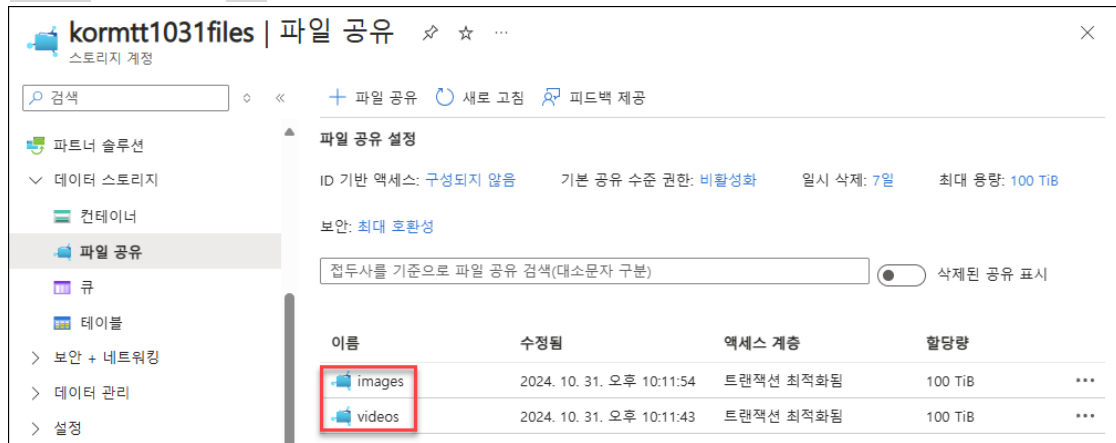
새 파일 공유

기본 사항 백업 검토 + 만들기

Azure Backup은 세분화된 복원 및 대규모 관리 기능을 사용하여 파일 공유를 실수로 삭제하거나 수정하지 못하도록 보호합니다. [자세히 알아보기](#)

백업 사용 ☐

6. 동일한 방법으로 "images" 이름의 파일 공유를 하나 더 만듭니다. videos 폴더는 소스 MP4 파일을 저장하는 파일 공유이며 images 폴더는 변환된 GIF 파일을 저장하는 파일 공유로 사용합니다.



7. [videos SMB 파일 공유] 블레이드의 [찾아보기]로 이동한 후 [업로드]를 클릭합니다. [파일 업로드] 창에서 소스로 사용할 MP4 동영상 파일을 여러 개 추가하고 [업로드]를 클릭합니다.



8. [스토리지 계정] 블레이드의 [보안 + 네트워크 - 액세스 키]로 이동한 후 "스토리지 계정 이름"과 "key1"의 값을 메모장에 복사합니다. 이 값은 Logic Apps에서 커넥터를 구성할 때 사용합니다.



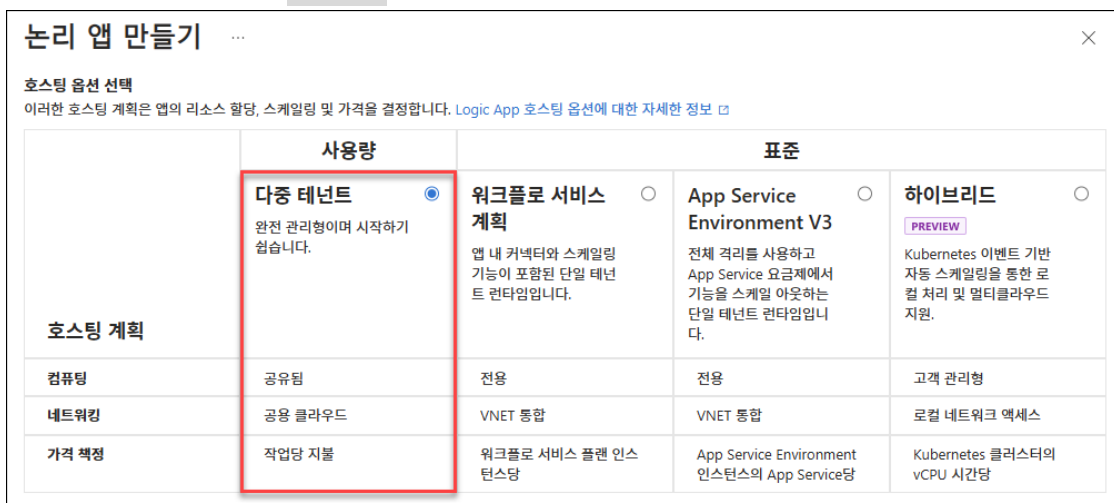
TASK 02. Logic Apps 만들기 및 Files 커넥터 구성

이 작업에서는 Logic Apps를 만들고 Azure Files 공유에 연결할 커넥터를 구성합니다.

1. Azure 포털의 검색창에서 "Logics Apps"를 검색한 후 클릭합니다. [Logic Apps] 블레이드의 메뉴에서 [추가]를 클릭합니다.

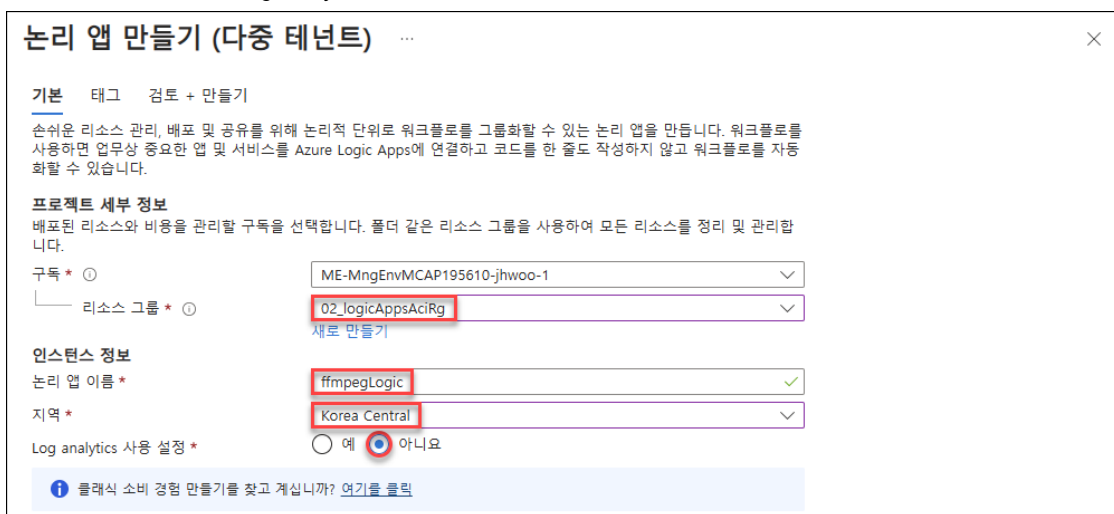


2. [논리 앱 만들기] 블레이드에서 "다중 테넌트"를 선택하고 [선택]을 클릭합니다.



3. [논리 앱 만들기 (다중 테넌트)] 블레이드의 [기본] 탭에서 아래와 같이 구성한 후 [검토 + 만들기]를 클릭합니다. [검토 + 만들기] 탭에서 [만들기]를 클릭합니다.

- [프로젝트 세부 정보 - 리소스 그룹]: 02_logicAppsAciRg
- [인스턴스 정보 - 논리 앱 이름]: ffmpegLogic
- [인스턴스 정보 - 지역]: Korea Central
- [인스턴스 정보 - Log Analytics 사용 설정]: 아니요

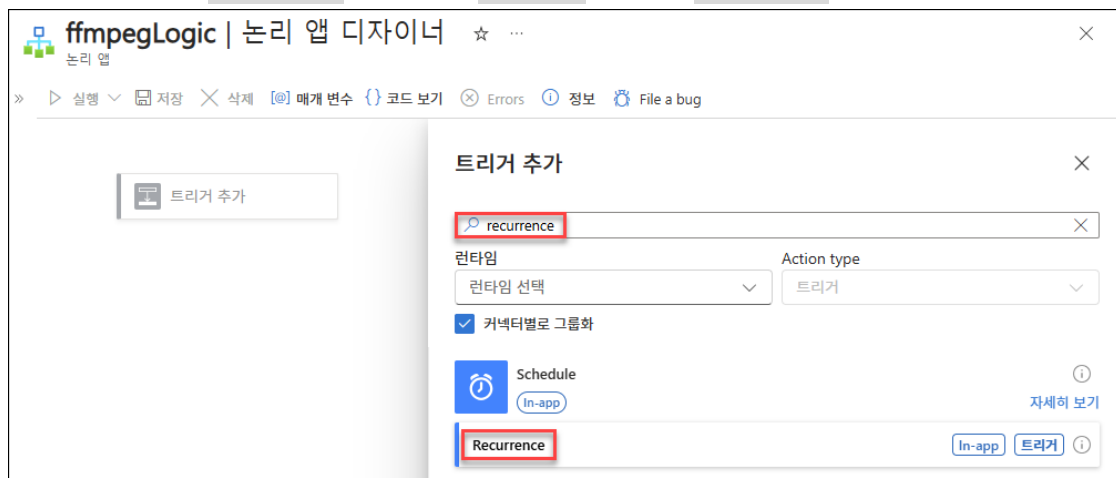


4. 새로 만든 [ffmpegLogic 논리 앱] 블레이드의 [개발 도구 - 논리 앱 디자이너]로 이동합니다. 디자이너 캔버스에서 [트리거

추가]를 클릭합니다.

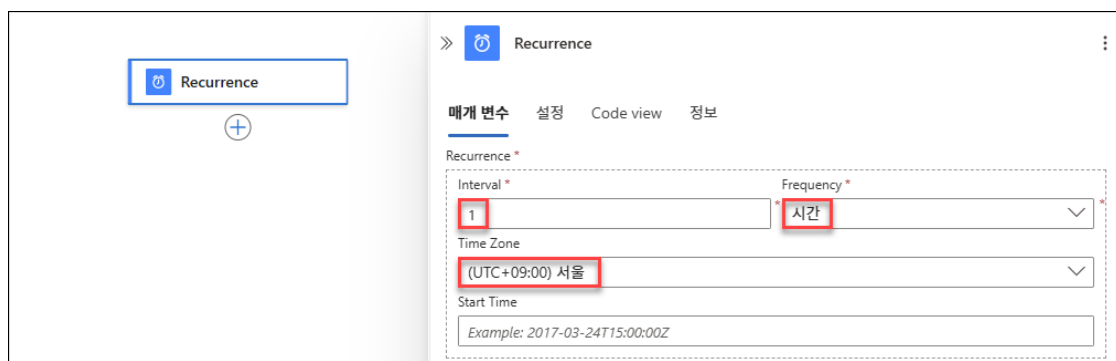


5. [트리거 추가] 창에서 "recurrence"를 검색한 후 "Schedule" 커넥터의 "Recurrence" 트리거를 선택합니다.

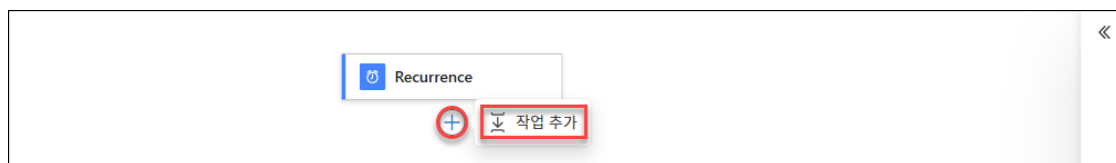


6. [Recurrence] 창의 [매개 변수] 탭에서 아래와 같이 구성합니다.

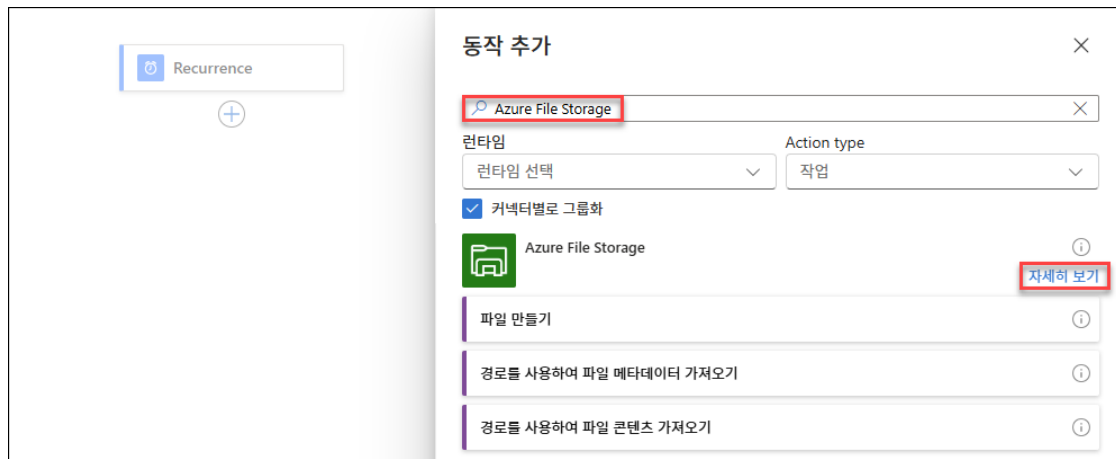
- Interval: 1 시간
- Time Zone: (UTC +09:00) 서울



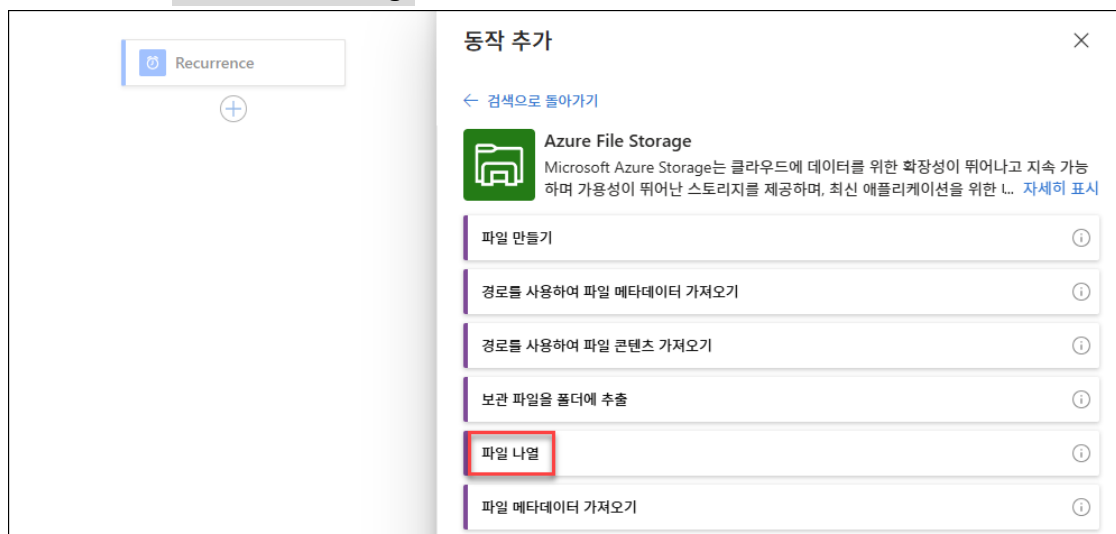
7. 디자이너 캔버스에서 [Recurrence] 트리거 하단의 [+ - 작업 추가]를 클릭합니다.



8. [동작 추가] 창에서 "Azure File Storage"를 검색한 후 "Azure File Storage" 커넥터의 "자세히 보기" 링크를 클릭합니다.

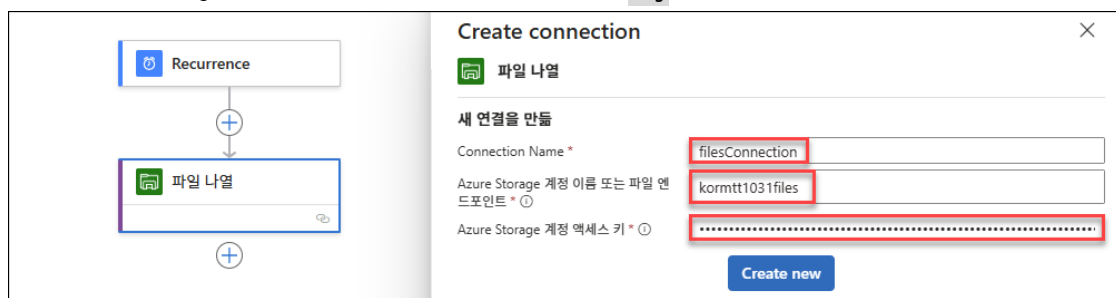


9. [동작 추가] 창의 "Azure File Storage"에서 [파일 나열]을 클릭합니다.

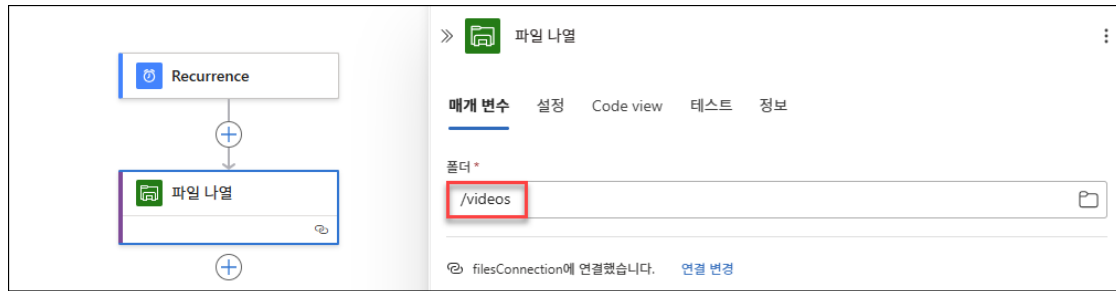


10. [Create connection – 파일 나열] 창에서 아래와 같이 구성한 후 [Create new]를 클릭합니다.

- Connection Name: filesConnection
- Azure Storage 계정 이름 또는 파일 엔드포인트: 앞서 복사한 스토리지 계정 이름을 입력합니다.
- Azure Storage 계정 액세스 키: 앞서 복사한 스토리지 계정의 key 값을 입력합니다.



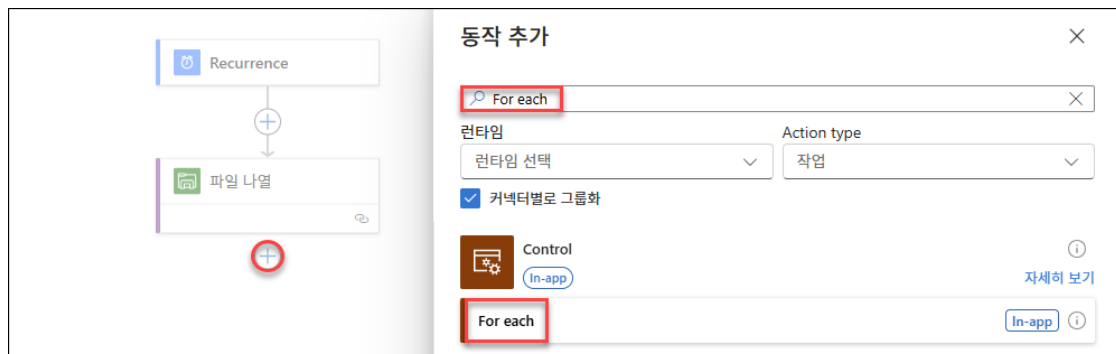
11. [파일 나열] 창의 [매개 변수] 탭에서 [찾아보기]를 클릭한 후 앞서 만든 스토리지 계정의 "/videos" 폴더를 선택합니다. 이제 설정한 간격으로 이 Azure Files 폴더에 있는 모든 파일을 나열하게 됩니다.



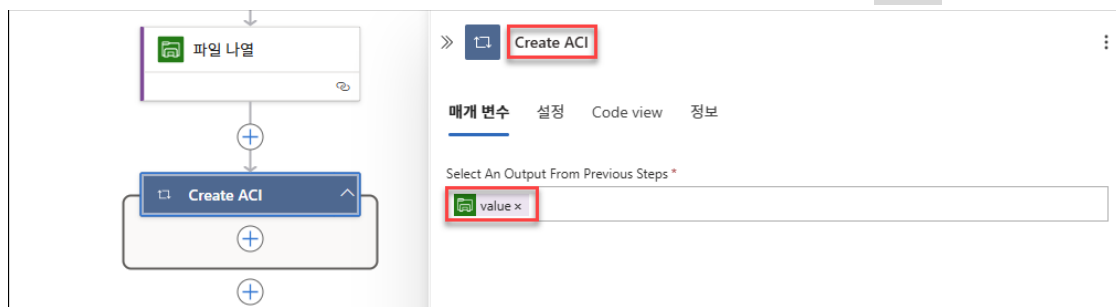
TASK 03. Azure 컨테이너 인스턴스 작업 구성

이 작업에서는 Azure Files 공유를 탑재하고 이 공유에 있는 파일을 작업할 Azure 컨테이너 인스턴스를 구성합니다.

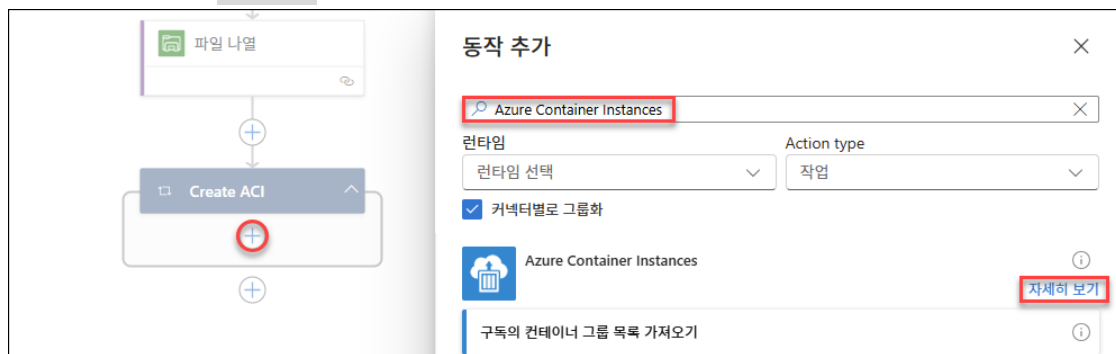
1. 디자이너 캔버스에서 [파일 나열] 아래의 [+ - 작업 추가]를 클릭합니다. [동작 추가] 창에서 "For each"를 검색한 후 선택합니다.



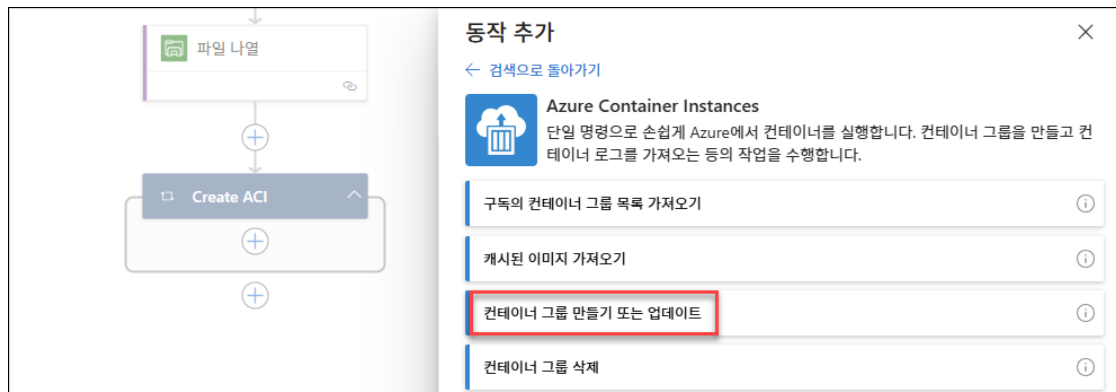
2. 추가한 작업의 이름을 "Create ACI"로 변경하고 아래와 같이 설정합니다.
 - Select An Output From Previous Steps: [동작 콘텐츠] 창을 열고 [파일 나열] 작업의 "value"를 선택합니다.



3. [Create ACI] 작업 바로 아래에 새 작업을 추가합니다. [동작 추가] 창에서 "Azure Container Instances"를 검색한 후 이 커넥터의 "자세히 보기" 링크를 클릭합니다.



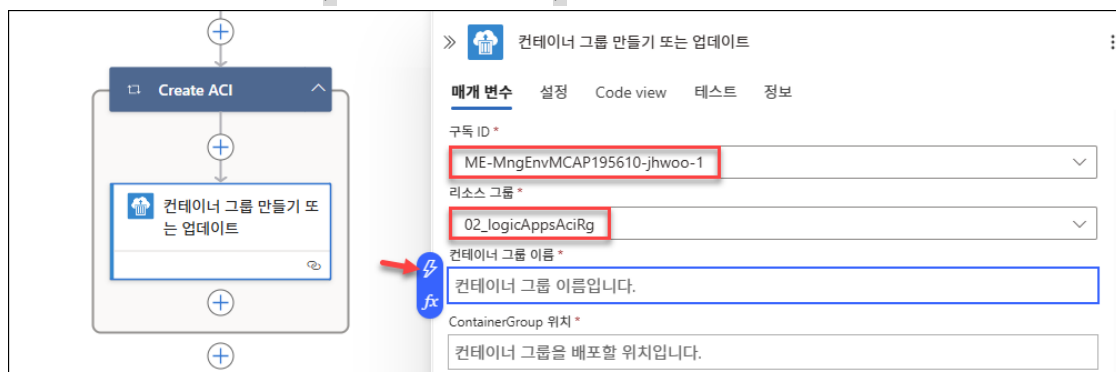
4. [동작 추가] 창에서 [컨테이너 그룹 만들기 또는 업데이트] 작업을 클릭합니다.



5. [Create connection] 창에서 OAuth 인증을 선택하고 [로그인]을 클릭합니다. 로그인 창이 표시되면 Azure 구독 계정으로 로그인합니다.

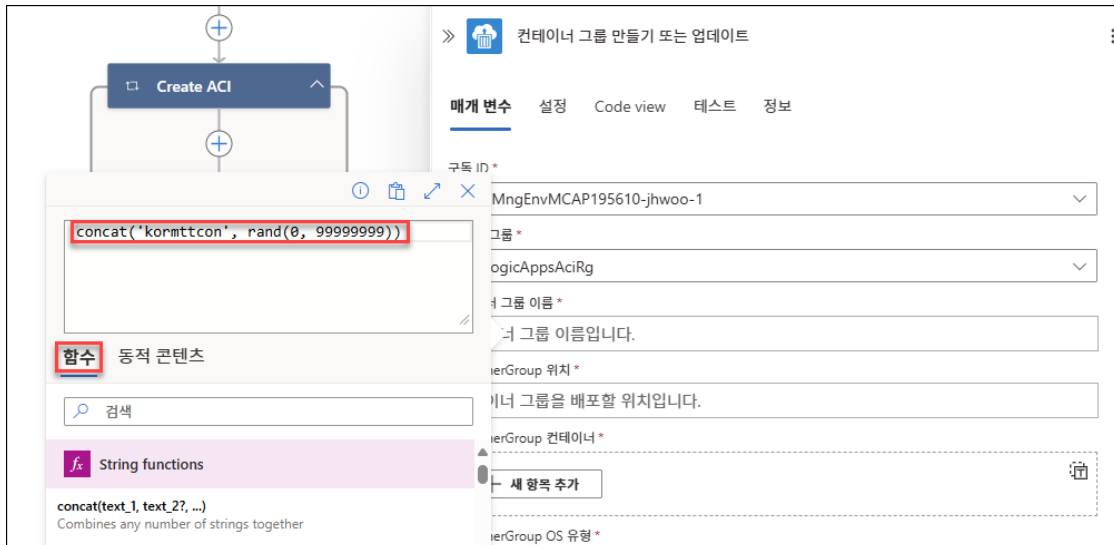


6. [컨테이너 그룹 만들기 또는 업데이트] 창의 [매개 변수] 탭에서 아래와 같이 구성합니다.
 - 구독 ID: 자신의 구독을 선택합니다.
 - 리소스 그룹: 02_logicAppsAciRg
 - 컨테이너 그룹 이름: 클릭한 후 함수 및 동작 콘텐츠를 볼 수 있는 아이콘을 클릭합니다. 또는 "/"를 입력하여 동일한 작업을 수행할 수 있습니다. "/"를 직접 입력하는 경우 "/" 문자열이 삭제되지 않도록 주의할 필요가 있습니다.



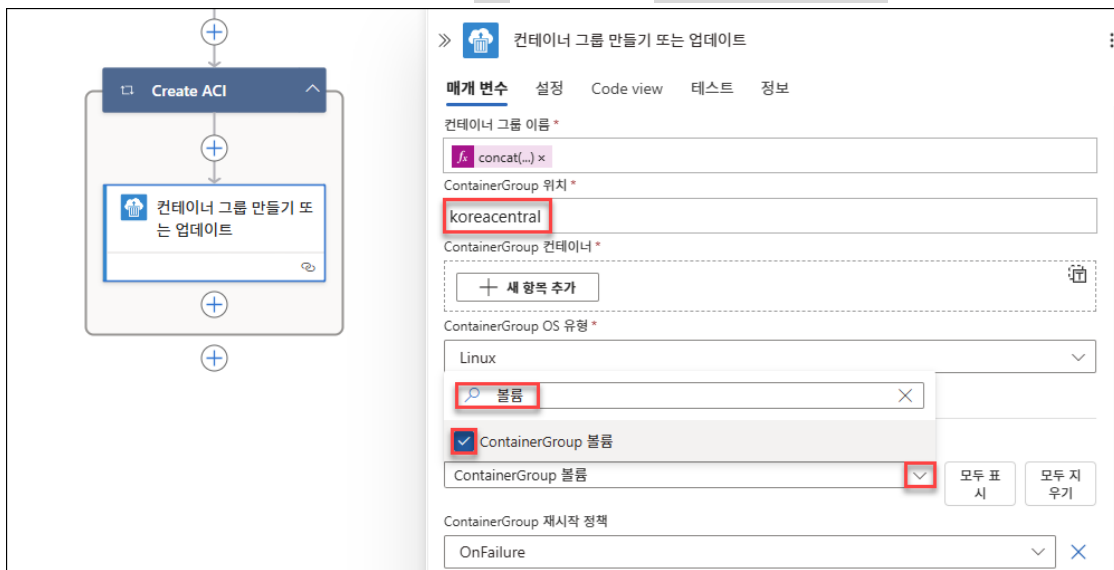
7. [함수]의 입력창에 다음과 같은 내용을 입력하고 [추가]를 클릭합니다. 이 함수는 "kormttcon<XXXXXXXX>" 형식의 임의 이름을 만듭니다.

```
concat('kormttcon', rand(0, 99999999))
```



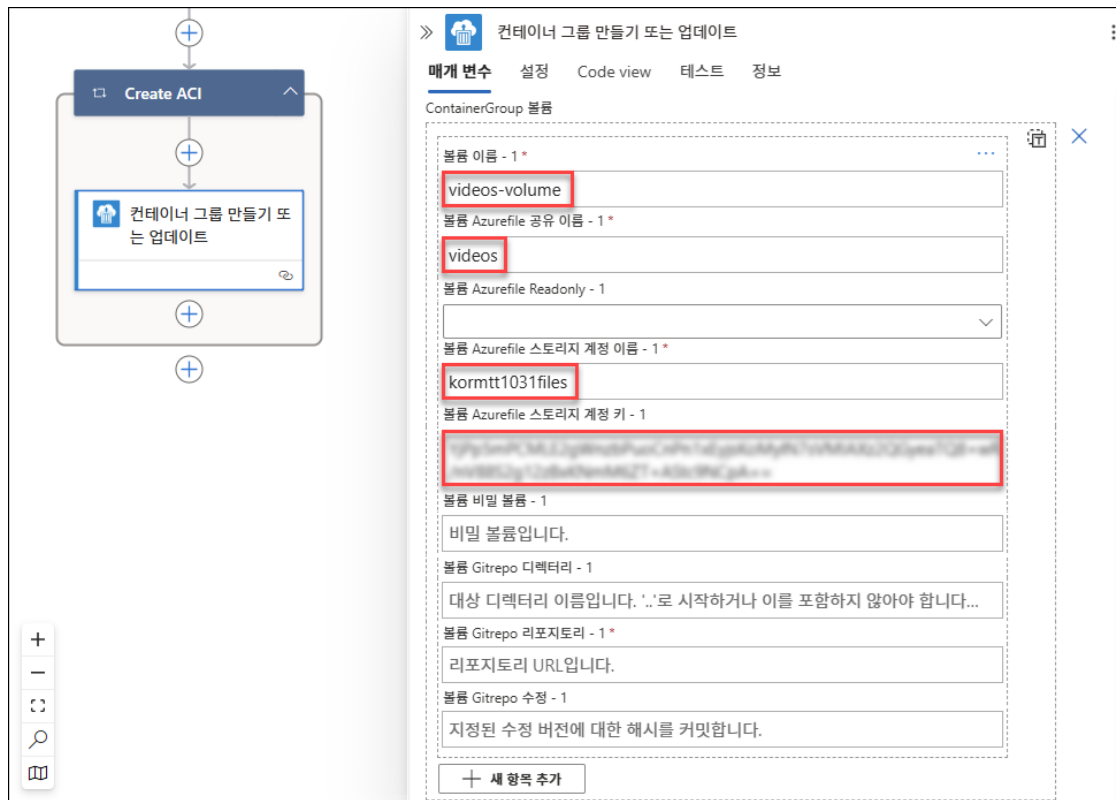
8. [매개 변수] 탭에서 계속 다음 설정을 구성합니다.

- ContainerGroup 위치: koreacentral
- 고급 매개 변수: 드롭 다운 메뉴를 클릭한 후 "볼륨"으로 검색하고 "ContainerGroup 볼륨"을 선택합니다.



9. 추가한 "ContainerGroup 볼륨" 영역에서 아래와 같이 구성합니다.

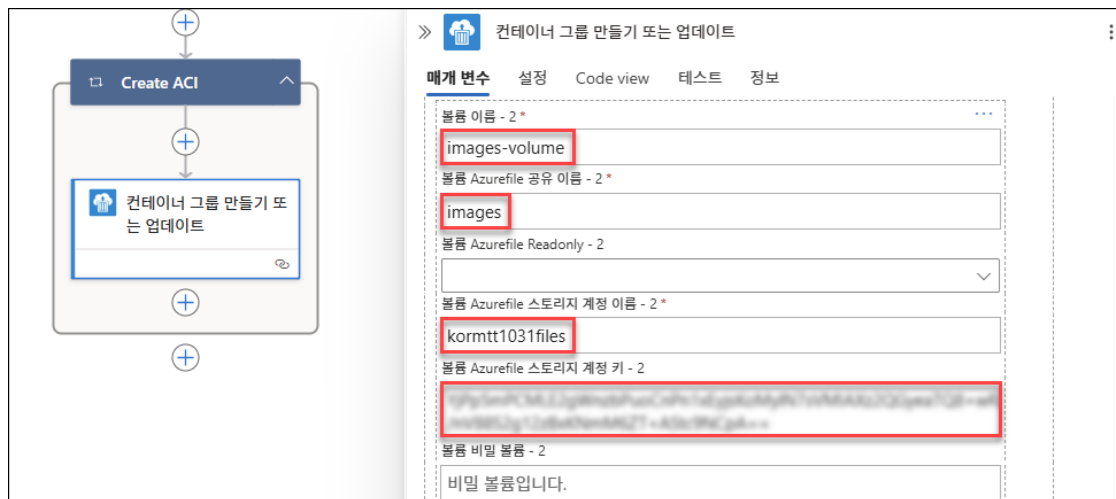
- 볼륨 이름 - 1: "videos-volume"을 입력합니다. 대문자는 사용할 수 없습니다.
- 볼륨 Azurefile 공유 이름 - 1: Azure Files에 만든 파일 공유 이름인 "videos"를 입력합니다.
- 볼륨 Azurefile 스토리지 계정 이름 - 1: 스토리지 계정의 이름을 붙여 넣습니다.
- 볼륨 Azurefile 스토리지 계정 키 - 1: 스토리지 계정의 키 값을 붙여 넣습니다.
- 다른 설정은 구성하지 않습니다.



10. 소스 동영상 파일에서 GIF 파일을 만든 다음 이를 저장할 Azure Files 디렉터리를 추가 볼륨으로 구성해야 합니다.

"ContainerGroup 볼륨" 영역에서 [새 항목 추가]를 클릭한 후 다음과 같이 추가 볼륨을 구성합니다.

- 볼륨 이름 - 2: images-volume
- 볼륨 Azurefile 공유 이름 - 2: Azure Files에 만든 파일 공유 이름인 "images"를 입력합니다.
- 볼륨 Azurefile 스토리지 계정 이름 - 2: 스토리지 계정의 이름을 붙여 넣습니다.
- 볼륨 Azurefile 스토리지 계정 키 - 2: 스토리지 계정의 키 값을 붙여 넣습니다.
- 다른 설정은 구성하지 않습니다.



11. 이제 추가한 볼륨을 탑재하고 FFmpeg 명령을 실행할 컨테이너 인스턴스를 구성해야 합니다. [매개 변수] 탭에서

"ContainerGroup 컨테이너" 영역에서 [새 항목 추가]를 클릭합니다.



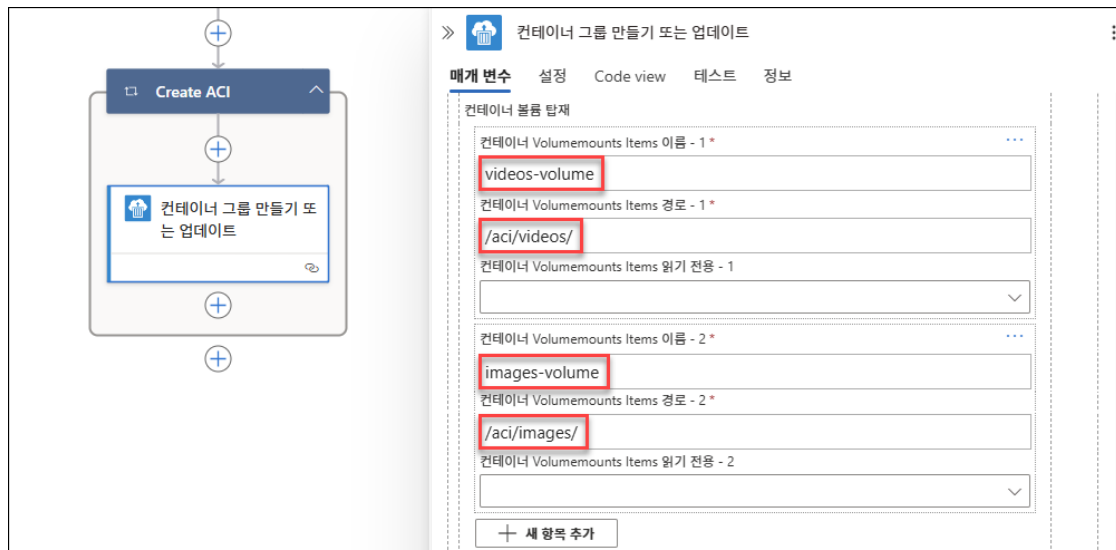
12. "ContainerGroup 컨테이너" 영역에서 아래와 같이 구성합니다.

- 컨테이너 이름 - 1: "/"를 입력하여 [함수] 창을 열고 `concat('kormttcon', rand(0, 99999999))` 식을 입력합니다.
- 컨테이너 이미지 - 1: `mirror.gcr.io/linuxserver/ffmpeg`
- 컨테이너 Resources Requests 메모리 - 1: 1
- 컨테이너 Resources Requests CPU - 1: 1
- 다른 리소스 설정은 빈 값을 유지합니다. 이러한 값 중에 필수 값이 있으나 이 값은 실제 필수 값이 아니기 때문에 나중에 수정합니다.
- "컨테이너 볼륨 탑재" 영역에서 [새 항목 추가]를 클릭합니다.

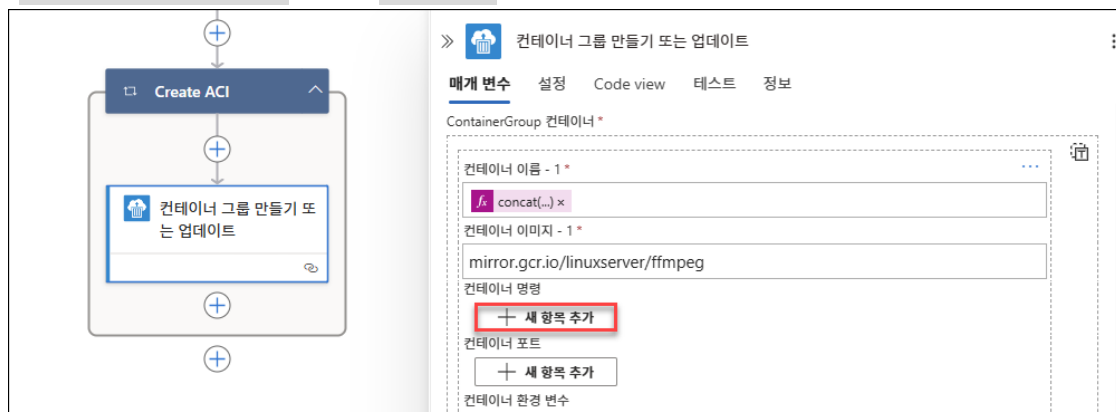


13. "컨테이너 볼륨 탑재" 영역에서 아래와 같은 2개의 볼륨을 추가합니다.

번호	컨테이너 Volumemounts Items 이름	컨테이너 Volumemounts Items 경로
1	videos-volume	/aci/videos/
2	images-volume	/aci/images/

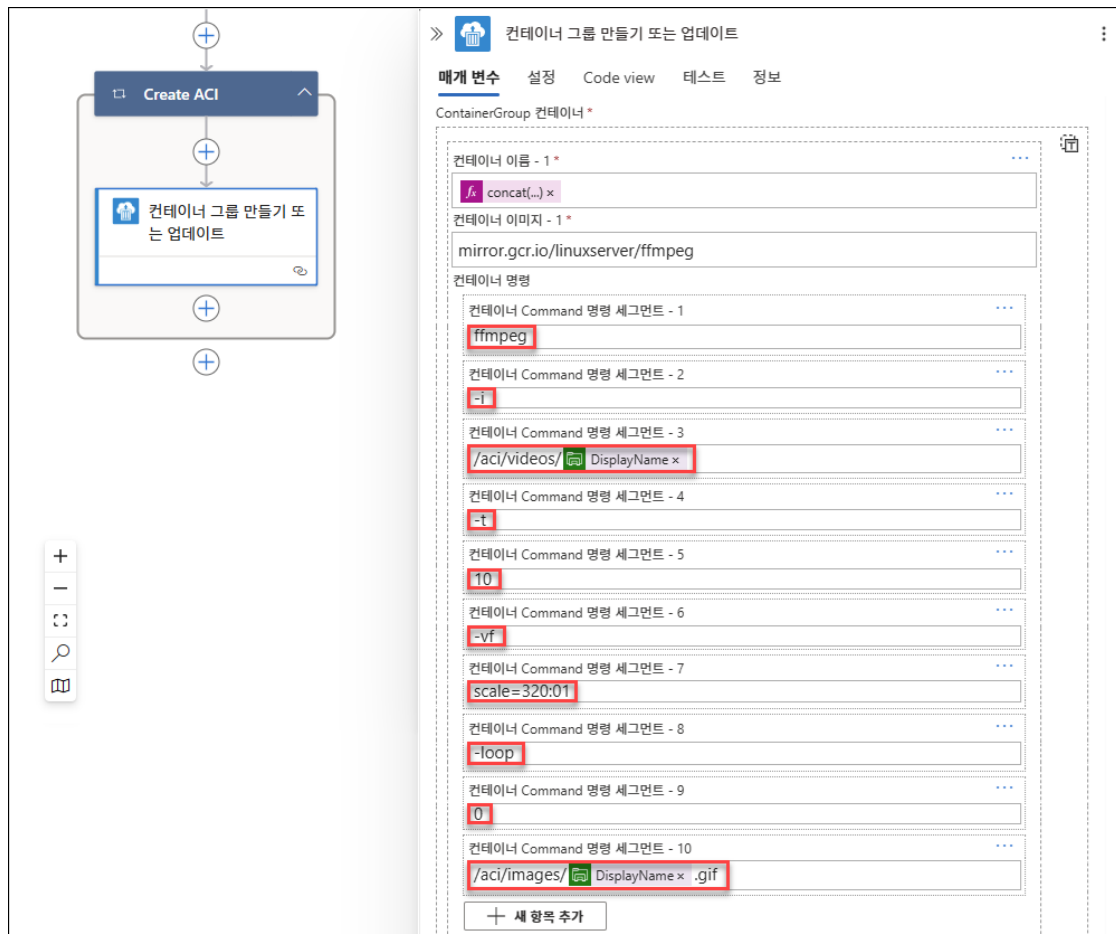


14. "ContainerGroup 컨테이너" 영역의 "컨테이너 명령"에서 [새 항목 추가]를 클릭합니다.



15. "컨테이너 명령" 영역에서 아래와 같은 값을 추가합니다. 실행할 명령은 공백으로 분할되어 한 줄마다 세그먼트를 하나씩 입력해야 합니다.

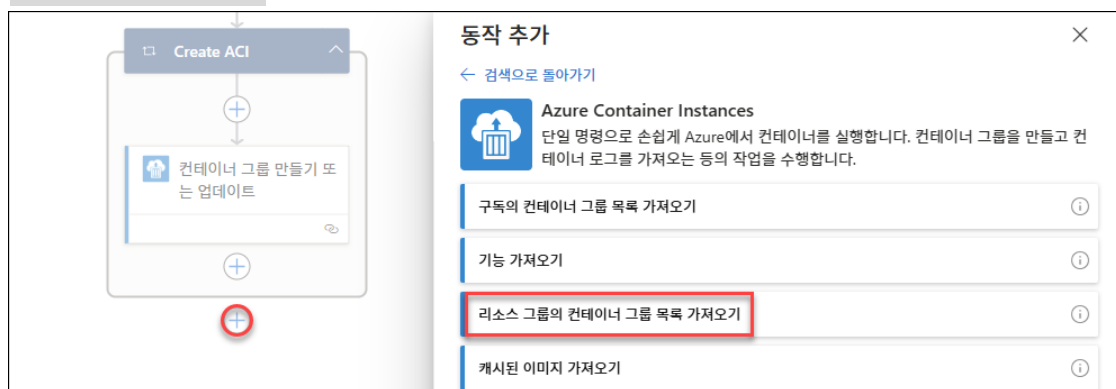
- 컨테이너 Command 명령 세그먼트 - 1: `ffmpeg`
- 컨테이너 Command 명령 세그먼트 - 2: `-i`
- 컨테이너 Command 명령 세그먼트 - 3: `"/aci/videos/"`까지 입력한 후 [동적 콘텐츠] 창을 열고 [파일 나열] 작업의 `DisplayName` 값을 선택합니다.
- 컨테이너 Command 명령 세그먼트 - 4: `-t`
- 컨테이너 Command 명령 세그먼트 - 5: `10`
- 컨테이너 Command 명령 세그먼트 - 6: `-vf`
- 컨테이너 Command 명령 세그먼트 - 7: `scale=320:-1`
- 컨테이너 Command 명령 세그먼트 - 8: `-loop`
- 컨테이너 Command 명령 세그먼트 - 9: `0`
- 컨테이너 Command 명령 세그먼트 - 10: `"/aci/images/"`까지 입력한 후 [동적 콘텐츠] 창을 열고 [파일 나열] 작업의 `DisplayName` 값을 선택합니다. 마지막에 `".gif"` 확장자를 추가합니다.



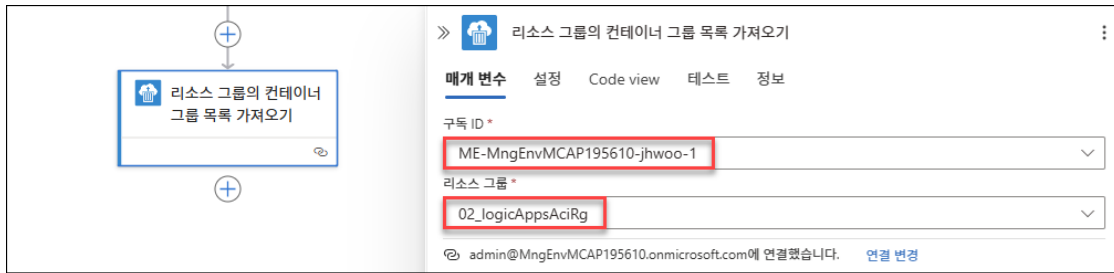
TASK 04. 작업이 완료된 컨테이너 인스턴스 삭제하기

컨테이너 인스턴스의 상태를 확인하여 작업이 완료된 컨테이너 인스턴스를 삭제해야 합니다. 이 작업을 Logic Apps에 구성합니다.

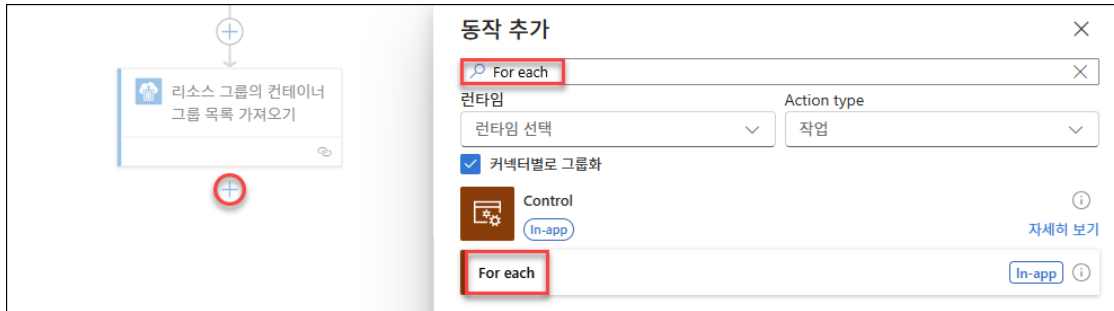
1. Logic Apps의 디자이너 캔버스에서 [Create ACI] 작업 아래에 새 작업을 추가합니다. [동작 추가] 창에서 "Azure Container Instances" 작업의 [리소스 그룹의 컨테이너 그룹 목록 가져오기]를 선택합니다.



2. [리소스 그룹의 컨테이너 그룹 목록 가져오기] 창의 [매개 변수] 탭에서 아래와 같이 구성합니다.
 - 구독 ID: 자신의 구독을 선택합니다.
 - 리소스 그룹: 02_logicAppsAcIRg

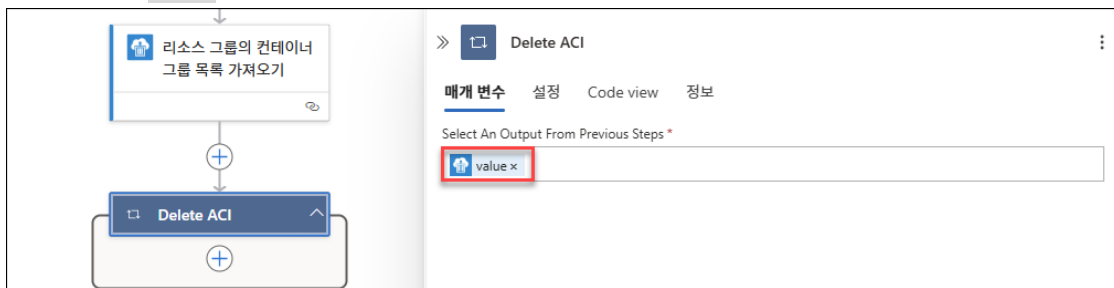


3. [리소스 그룹의 컨테이너 그룹 목록 가져오기] 타일 아래에서 새 작업을 추가합니다. [동작 추가] 창에서 "For each"를 검색한 후 선택합니다.

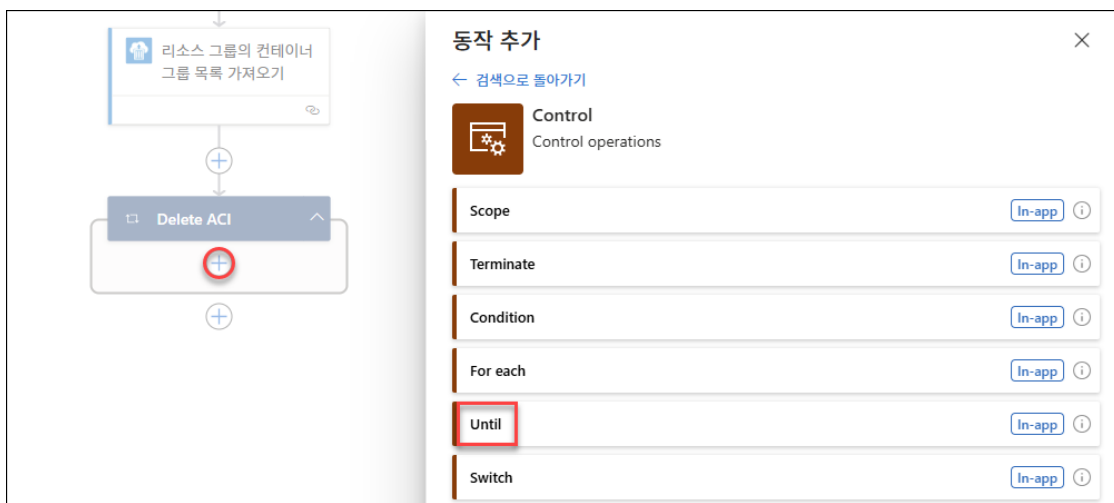


4. 추가한 [For each] 작업의 이름을 "Delete ACI"로 변경하고 아래와 같이 설정합니다.

- Select An Output From Previous Steps: [동작 콘텐츠] 창을 열고 [리소스 그룹의 컨테이너 그룹 목록 가져오기] 작업의 "value"를 선택합니다.



5. [Delete ACI] 작업 내에서 새 작업을 추가합니다. [동작 추가] 창에서 "Control"을 검색하고 "자세히 보기"를 클릭합니다. [Control] 창에서 [Until]을 클릭합니다.

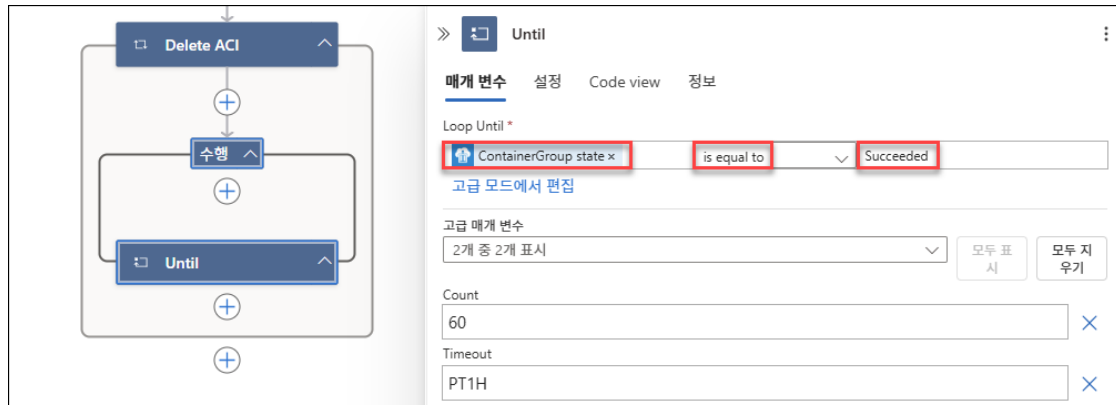


6. [Until] 창에서 아래와 같이 구성합니다.

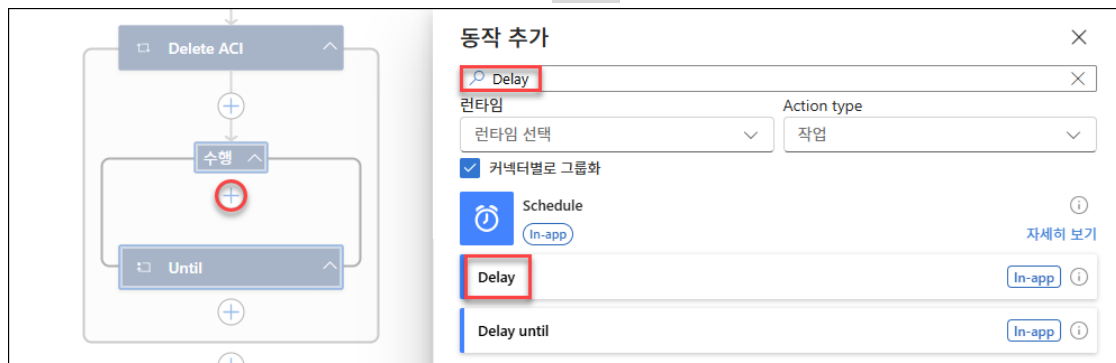
- Loop Until: [동작 콘텐츠] 창을 열고 [리소스 그룹의 컨테이너 그룹 목록 가져오기] 작업 중에서 "ContainerGroup

state"를 선택합니다. "ContainerGroup 컨테이너"로 시작하는 작업은 컨테이너 그룹의 개별 컨테이너에 대한 작업이며 "ContainerGroup"으로 표시되는 작업은 컨테이너 그룹에 대한 작업이니 혼동하지 않아야 합니다.

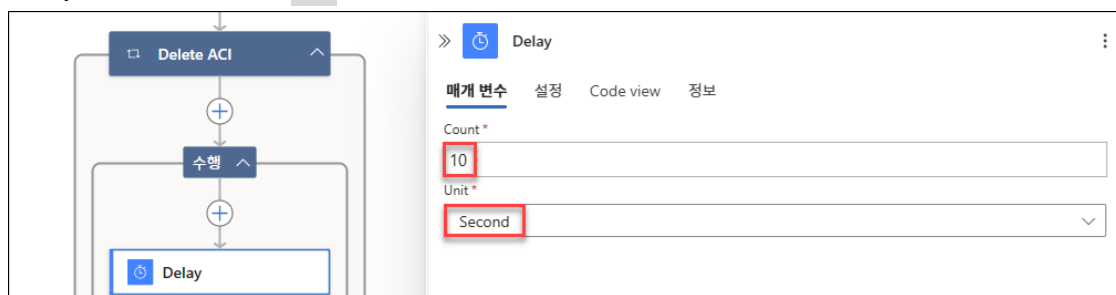
- 조건: is equal to
- 값 선택: "Succeeded"를 입력합니다. Azure 컨테이너 인스턴스는 정상적으로 작업이 완료되고 컨테이너가 종료되면 이 로그를 출력합니다.



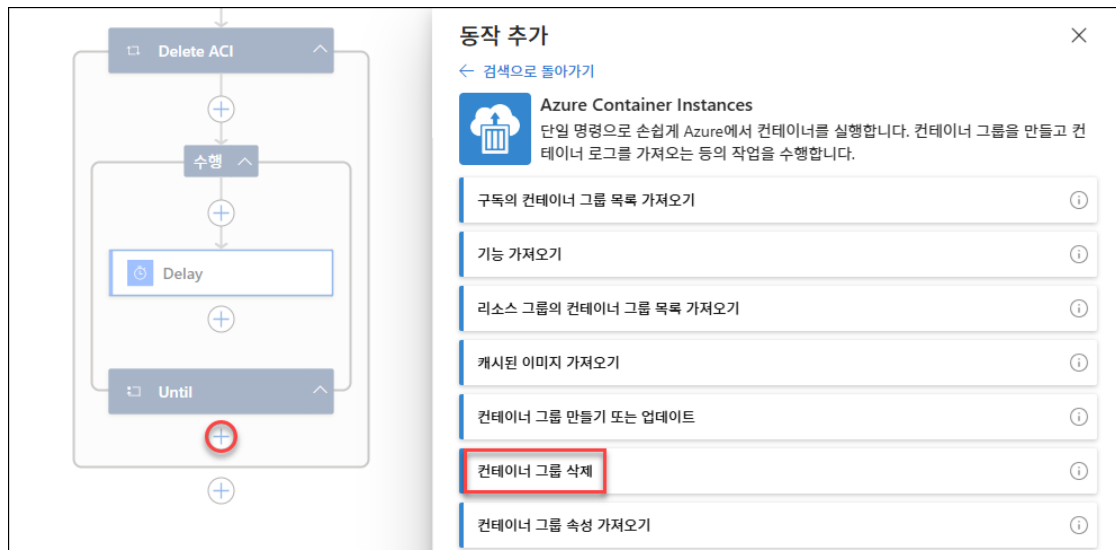
7. [수행] 부분 아래에 새 작업을 추가합니다. [동작 추가] 창에서 "Delay"를 검색한 후 선택합니다.



8. [Delay] 창의 [매개 변수]에서 "10초"를 기다리도록 설정합니다.

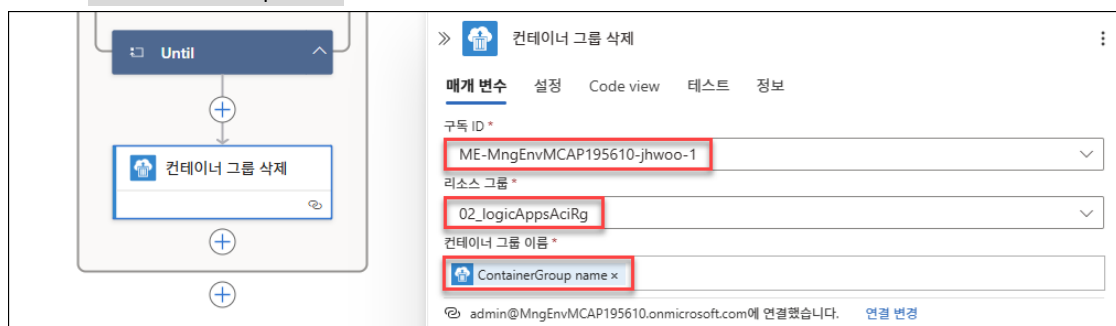


9. [Until] 작업 바로 아래에서 작업을 추가합니다. [동작 추가] 창에서 "Azure Container Instances"의 [컨테이너 그룹 삭제] 작업을 선택합니다.



10. [컨테이너 그룹 삭제] 창의 [매개 변수] 탭에서 아래와 같이 구성합니다.

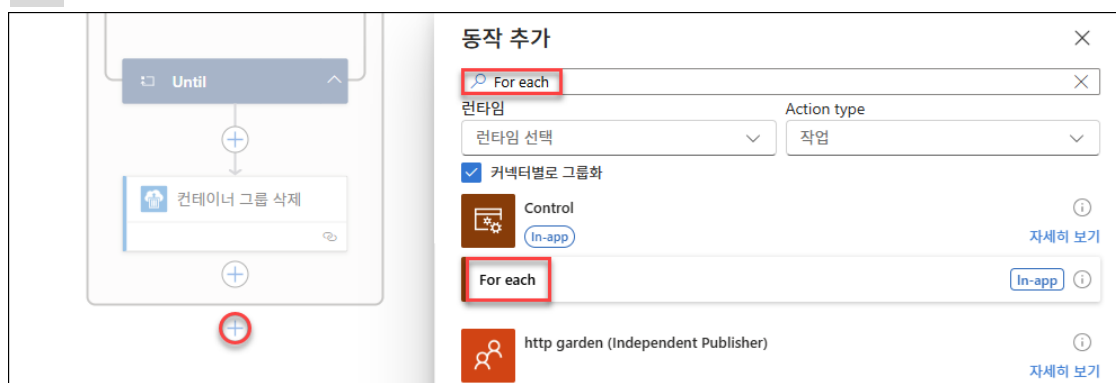
- 구독 ID: 자신의 구독 ID를 선택합니다.
- 리소스 그룹: 02_logicAppsAcirg
- 컨테이너 그룹 이름: [동적 콘텐츠] 창을 열고 [리소스 그룹의 컨테이너 그룹 목록 가져오기] 작업의 "ContainerGroup name"을 선택합니다.



TASK 05. 작업이 완료된 파일 삭제

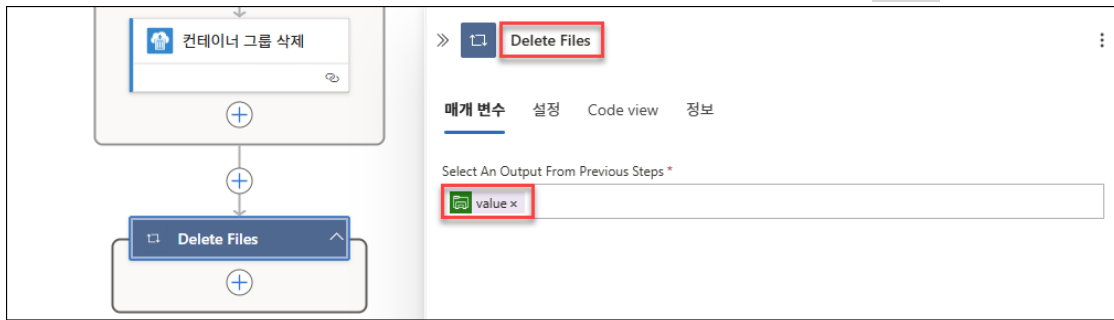
이 작업에서는 GIF로 변환이 완료된 원본 동영상 파일을 삭제하는 작업을 진행합니다.

- Logic Apps의 디자이너 캔버스에서 [Delete ACI] 작업 아래에서 새 작업을 추가합니다. [동작 추가] 창에서 "ForEach"를 검색하고 선택합니다.

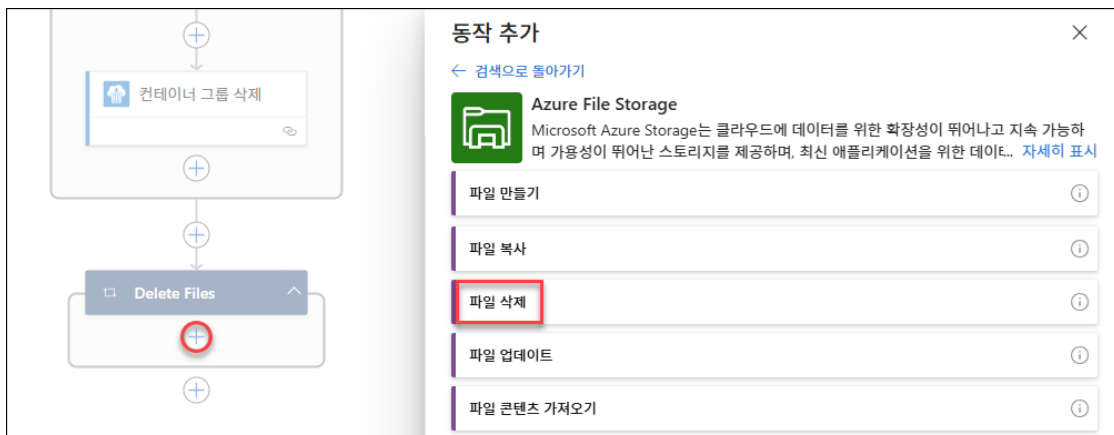


- [ForEach]의 작업 이름을 "Delete Files"로 변경하고 아래와 같이 구성합니다.

- Select An Output From Previous Steps: [동적 콘텐츠] 창을 열고 [파일 나열] 작업의 "value"를 선택합니다.

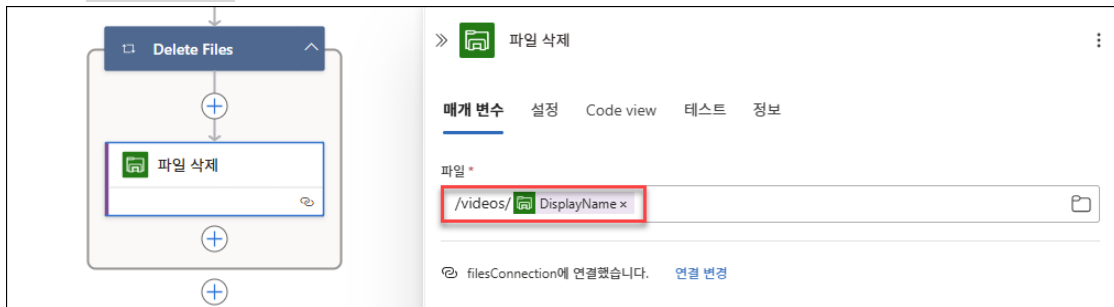


3. [Delete Files] 작업 아래에 새 작업을 추가합니다. [동작 추가] 창에서 "Azure File Storage"의 [파일 삭제] 작업을 선택합니다.



4. [파일 삭제] 창의 [매개 변수] 탭에서 아래와 같이 구성합니다.

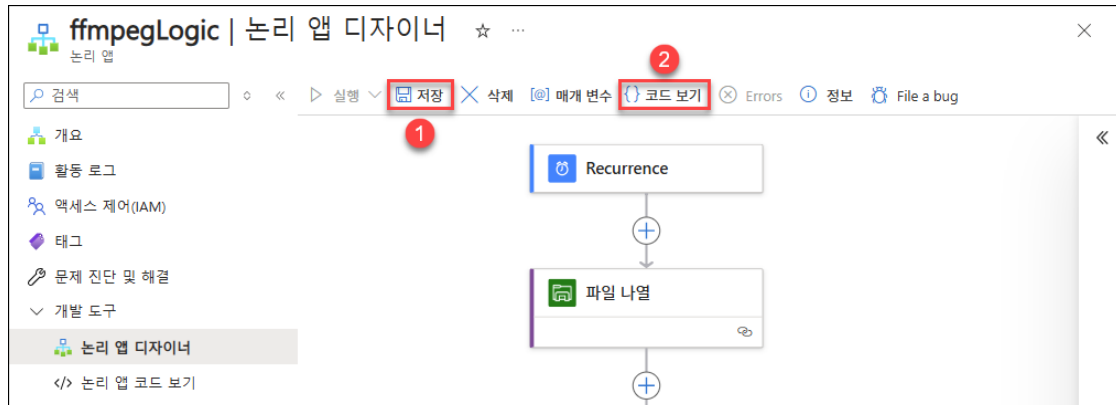
- 파일: "/videos/" 경로를 입력하고 [동적 콘텐츠] 창을 엽니다. [동적 콘텐츠] 창에서 [파일 나열] 작업의 "DisplayName"을 선택합니다.



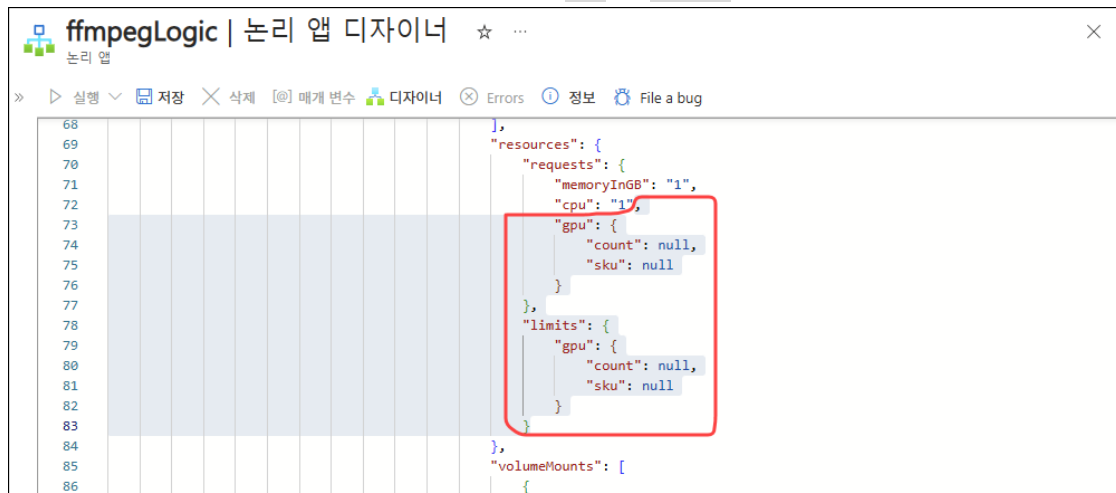
TASK 06. Logic Apps 코드 수정

Logic Apps의 디자이너 캔버스에서 컨테이너 인스턴스를 구성하면 필수 값이 아닌 GPU 설정이나 Git Repos 설정이 필수로 구성됩니다. 이러한 불필요한 값을 제거하지 않으면 Logic Apps이 실행될 때 오류가 발생합니다. 이 작업에서는 이러한 불필요한 설정을 JSON 코드 편집에서 삭제합니다.

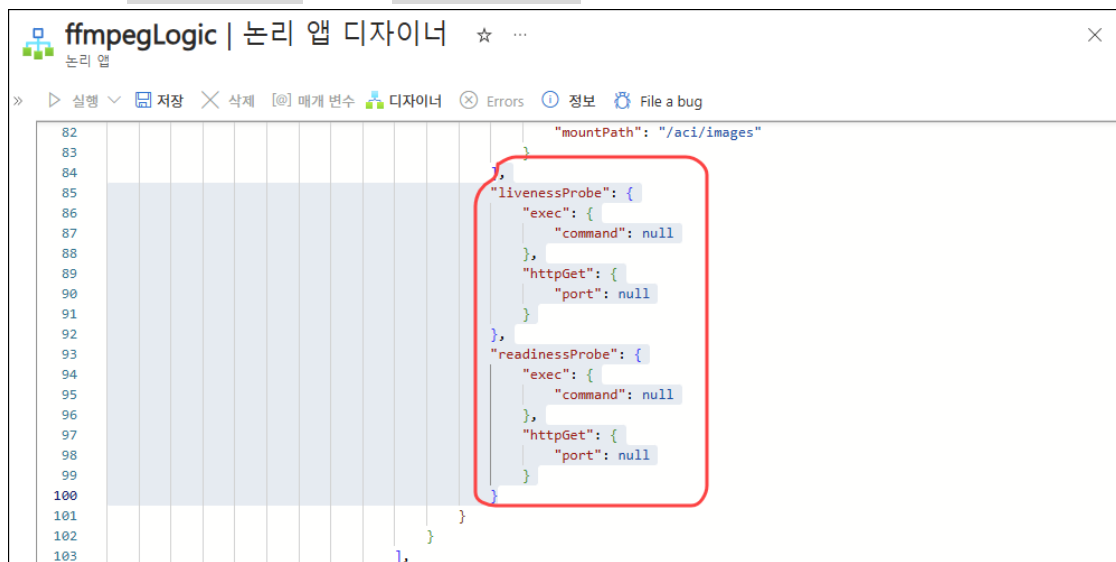
1. [ffmpegLogic 논리 앱] 블레이드의 [개발 도구 - 논리 앱 디자이너]에서 [저장]을 클릭한 후 [코드 보기]를 클릭합니다. 설정에 문제가 있는 경우 [저장]을 클릭하면 관련 오류가 표시되고 저장 작업이 진행되지 않습니다.



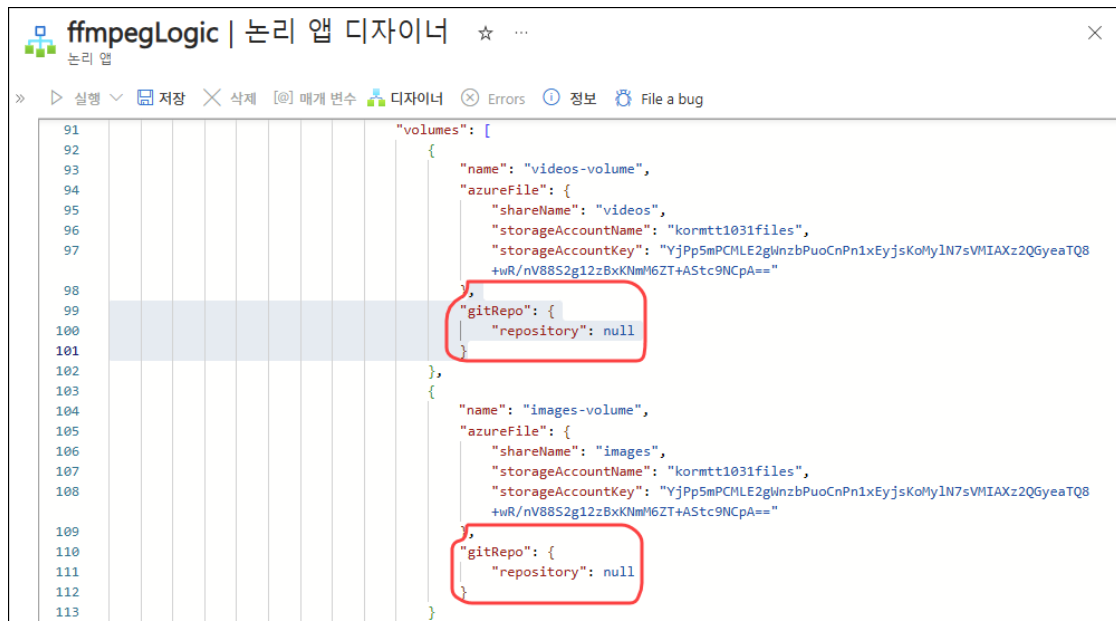
2. JSON 코드 창에서 컨테이너 인스턴스의 리소스 설정 영역에서 "gpu"와 "limits" 속성을 모두 삭제합니다.



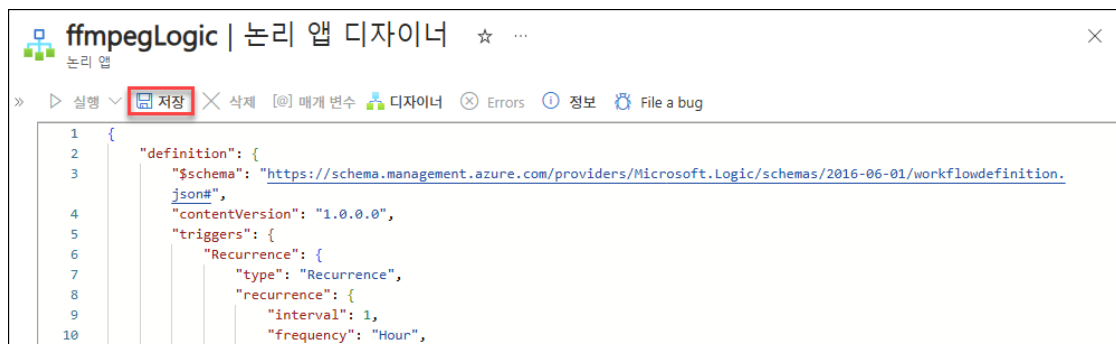
3. 코드 창에서 "livenessProbe" 설정과 "readinessProbe" 설정을 모두 삭제합니다.



4. 코드 창에서 "volumes" 영역에 있는 "gitRepo" 설정을 모두 삭제합니다.



5. 메뉴에서 [저장]을 클릭하여 코드 변경 사항을 저장합니다.



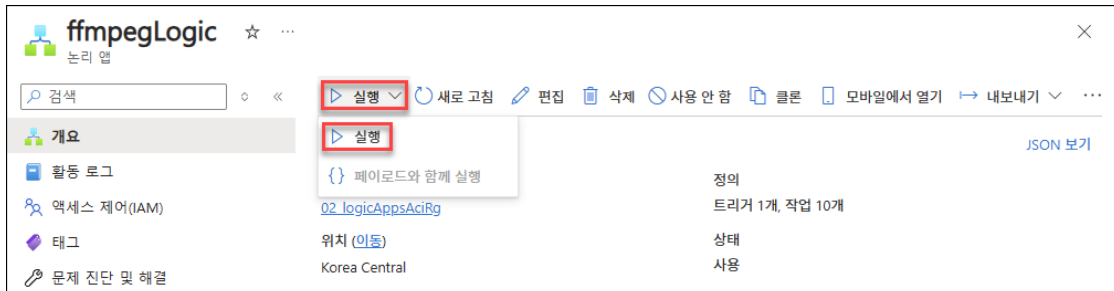
TASK 07. Logic Apps 실행 및 확인

Logic Apps이 디자이너 캔버스에서 설정한 작업은 아래와 같이 구성되어 있어야 합니다.



이 작업에서는 구성된 Logic Apps를 통해 ACI가 만들어지고 작업이 완료된 후 Azure Files의 파일이 삭제되는지 확인합니다.

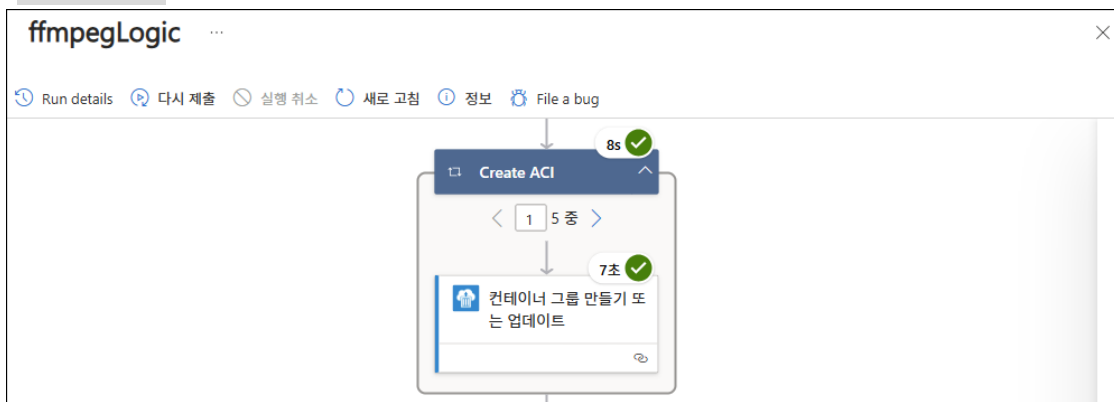
1. `ffmpegLogic` 논리 앱] 블레이드의 [개요]에서 [실행 - 실행]을 클릭합니다.



2. [ffmpegLogic] 논리 앱] 블레이드의 [개요]에서 [기록 실행] 탭으로 이동합니다. 실행 중인 작업을 클릭합니다.



3. [ffmpegLogic] 실행 창에서 작업 실행을 모니터링할 수 있습니다.



4. Azure 포털의 검색창에서 "Container Instances"를 검색한 후 클릭합니다. [Container Instances] 블레이드에서 Azure Files 공유에 있는 파일 수만큼의 컨테이너 인스턴스가 생성되고 실행되는 것을 확인합니다.



5. [images SMB 파일 공유] 블레이드로 이동한 후 [찾아보기]를 클릭합니다. 컨테이너 인스턴스를 통해 만들어진 GIF 파일이

표시되는지 확인합니다.



6. 모든 작업이 완료되면 작업을 완료한 컨테이너 인스턴스가 삭제되고 원본 동영상 파일도 삭제되는 것을 확인합니다.