



# **Workflow Automation 설계자 이해**

## OnCommand Workflow Automation

NetApp  
October 09, 2025

# 목차

Workflow Automation 설계자 이해	1
OnCommand Workflow Automation에서 빌딩 블록 작업	1
데이터 원본은 무엇입니까	1
사전 항목이 무엇입니까	2
명령이 작동하는 방식	2
필터란	3
어떤 장애물이 있는지 확인합니다	3
기능은 무엇입니까	4
구성표는 무엇입니까	4
어떤 원격 시스템 유형인지 확인하십시오	5
서식 파일 사용 방법	5
범주를 사용하는 방법	5
엔터티 버전 관리 작동 방식	6
운동장 데이터베이스입니다	8

# Workflow Automation 설계자 이해

WFA(Workflow Automation) 디자이너에서 파인더, 필터, 명령 등의 구성 요소를 사용하여 워크플로우를 생성합니다. 워크플로우를 생성하기 전에 구성 요소와 워크플로우 생성 프로세스를 이해하는 것이 중요합니다.

## OnCommand Workflow Automation에서 빌딩 블록 작업

WFA(Workflow Automation) 워크플로우는 여러 개의 구성 요소로 구성되며 WFA에는 사전 정의된 구성 요소 라이브러리가 포함되어 있습니다. WFA에서 제공하는 구성 요소를 사용하여 조직의 요구사항에 맞는 워크플로우를 생성할 수 있습니다.

WFA는 스토리지 자동화 프로세스에 대한 구조를 제공합니다. WFA의 유연성은 워크플로우 구성 요소를 사용하여 워크플로우를 구성하는 방식을 기반으로 합니다.

WFA 구성 요소는 다음과 같습니다.

- 사전 항목
- 명령
- 필터
- 파인더
- 기능
- 템플릿

WFA에서 구성 요소를 사용하여 워크플로우를 생성하는 방법을 이해해야 합니다.

### 데이터 원본은 무엇입니까

데이터 원본은 데이터를 추출하기 위해 다른 시스템, 파일 및 데이터베이스에 대한 연결을 설정하는 방법입니다. 예를 들어, 데이터 소스는 Active IQ Unified Manager 9.4 데이터 소스 유형의 Active IQ Unified Manager 데이터베이스에 대한 연결이 될 수 있습니다.

필수 데이터 소스 유형을 정의한 후 데이터 수집을 위해 WFA(OnCommand Workflow Automation)에 사용자 지정 데이터 소스를 추가할 수 있습니다. 캐싱 체계, 필요한 포트 및 획득 방법을 데이터 소스 유형과 연결하면 됩니다.

WFA는 다양한 데이터 소스를 통해 정보를 캐싱합니다. WFA는 데이터 소스에서 리소스 정보를 수집하고 캐싱 체계에 맞게 형식을 지정합니다. 캐시 테이블(캐싱 구조 내의 테이블)은 사전 항목 개체와 일치하도록 서식이 지정됩니다. 워크플로에서 파인더를 사용하면 사전 개체가 반환되고 사전 개체의 데이터가 캐시 테이블에서 채워집니다. 데이터 소스에서 데이터를 획득하는 프로세스를 \_데이터 소스 획득\_ 이라고 합니다. 스크립트 기반 방법이나 드라이버 기반 방법을 사용하여 데이터 소스를 획득할 수 있습니다. 소스는 서로 다를 수 있으며 데이터 소스 획득은 서로 다른 시간 간격으로 이를 샘플링할 수 있습니다. 그런 다음 WFA가 해당 정보를 데이터베이스에 병합하고 예약 데이터를 중첩하여 데이터베이스의 업데이트된 리소스 정보를 유지합니다.

WFA 데이터베이스에는 여러 가지 다른 캐싱 스키마가 포함되어 있습니다. 캐싱 체계는 테이블 집합이며 각 테이블에는 특정 사전 항목 형식의 정보가 포함되어 있지만 특정 데이터 소스 형식의 여러 원본에서 가져온 결합된 정보가 포함될 수 있습니다. WFA는 데이터베이스 정보를 사용하여 리소스 상태를 이해하고, 계산을 수행하고, 리소스에 대한 명령을

실행합니다.

## 사전 항목이 무엇입니까

사전 항목은 WFA(OnCommand Workflow Automation) 구성 요소 중 하나입니다. 사전 항목을 사용하여 스토리지 및 스토리지 관련 환경에서 개체 유형과 해당 관계를 나타낼 수 있습니다. 그런 다음 워크플로의 필터를 사용하여 사전 항목의 기본 키 값을 반환할 수 있습니다.

사전 항목은 WFA에서 지원하는 개체 유형의 정의입니다. 각 사전 항목은 지원되는 스토리지 및 스토리지 관련 환경에서 개체 유형과 해당 관계를 나타냅니다. 사전 개체는 형식 검사를 받을 수 있는 특성 목록으로 구성됩니다. 완전한 값이 있는 사전 개체는 형식의 개체 인스턴스를 설명합니다. 또한 참조 속성은 개체와 환경의 관계를 설명합니다. 예를 들어 볼륨 사전 개체에는 이름, size\_MB 및 volume\_보증 같은 여러 특성이 있습니다. 또한 볼륨 사전 개체에는 array\_id 및 aggregate\_id 형식으로 볼륨을 포함하는 배열과 Aggregate에 대한 참조가 포함됩니다.

개체의 캐시 테이블은 캐시용으로 표시된 사전 항목의 일부 또는 모든 특성이 포함된 데이터베이스입니다. 캐시 테이블을 포함하는 사전 항목의 경우 캐싱 대상으로 적어도 하나의 사전 항목의 특성을 표시해야 합니다. 사전 항목에는 개체의 고유 식별자인 고유 키가 포함됩니다. 예를 들어 7-Mode 볼륨은 고유한 이름 및 해당 항목이 포함된 어레이의 IP 주소로 식별됩니다. qtree는 qtree 이름, 볼륨 이름 및 어레이 IP 주소로 식별됩니다. 사전 항목을 만들 때 사전 항목의 기본 키에 속하는 사전 특성을 식별해야 합니다.

## 명령이 작동하는 방식

OnCommand Workflow Automation 명령은 워크플로우의 실행 블록입니다. 워크플로의 각 단계에 대해 명령을 사용할 수 있습니다.

WFA 명령은 PowerShell 및 Perl 스크립트를 사용하여 작성됩니다. 패키지가 설치된 경우 PowerShell 명령은 Data ONTAP PowerShell 툴킷과 VMware PowerCLI를 사용합니다. Perl 명령은 WFA 서버에 설치된 Perl 배포 및 Perl 모듈을 사용합니다. 명령에 PowerShell 및 Perl과 같은 여러 스크립팅 언어를 포함할 경우, 해당 스크립트가 설치된 운영 체제 및 WFA 구성 메뉴에 지정된 기본 언어 순서에 따라 WFA에서 적절한 스크립트를 선택합니다.

WFA 명령에 대한 스크립트에는 여러 매개 변수가 포함됩니다. 이러한 매개 변수는 사전 항목 속성에 매핑될 수 있습니다.

각 WFA 명령에는 여러 Data ONTAP 명령이 포함될 수 있습니다.

일부 WFA 명령은 오래 실행되는 작업을 기다리고 주기적으로 폴링할 수 있기 때문에 대기 명령이라고 합니다. 예를 들어 \* 여러 볼륨 이동 대기 \* 명령. 폴링 명령이 실행되는 대기 간격은 작업이 완료되었는지 확인하도록 구성할 수 있습니다.

WFA 명령은 워크플로가 실행 단계에 있는 동안 WFA에서 시작됩니다. WFA는 왼쪽에서 오른쪽 및 위에서 아래로 순서대로 명령을 순차적으로 실행합니다. 워크플로를 계획하면 명령에 제공되는 매개 변수의 사용 가능성 및 유효성을 확인할 수 있습니다. WFA 서버는 명령을 실행하기 전에 명령에 필요한 모든 매개 변수를 제공합니다.

명령에 대한 매개 변수는 워크플로 계획 중에 완료됩니다. 그런 다음 워크플로는 실행 시간 동안 이러한 매개 변수를 명령에 전달합니다. 명령은 매개 변수를 워크플로로 다시 전달할 수 없습니다. 하지만 워크플로의 명령 간에 실행 시간 중에 얻은 정보를 교환하려면 지정된 WFA PowerShell cmdlet 또는 Perl 기능을 사용할 수 있습니다.

WFA PowerShell 명령은 PowerShell cmdlet에 대해 -ErrorAction 중지 플래그를 사용하지 않으므로 오류로 인해 cmdlet이 실패할 경우에도 워크플로우 작업이 계속 진행됩니다. 특정 명령에 -ErrorAction 중지 플래그를 포함하려는 경우 명령을 복제하고 PowerShell 스크립트를 수정하여 플래그를 추가할 수 있습니다.

다음은 명령 간 정보 교환을 위해 WFA에 포함된 PowerShell cmdlet 및 Perl 기능입니다.

PowerShell cmdlet	Perl 기능
Add-WfaWorkflowParameter 를 참조하십시오	addWfaWorkflowParameter 를 참조하십시오
Get-WfaWorkflowParameter 를 참조하십시오	getWfaWorkflowParameter를 참조하십시오

""add"" cmdlet 또는 명령에 추가된 파라미터는 이후에 실행되는 명령으로 검색하여 ""get"" cmdlet 또는 기능을 사용할 수 있습니다. 예를 들어 PowerShell WFA 명령에서 다음 코드를 사용하여 volumeId라는 매개 변수를 추가할 수 있습니다. "Add-WfaWorkflowParameter -Name "VolumeUUID" -Value "12345" -AddAsReturnParameter\$ true" 그런 다음 후속 명령에서 다음을 사용하여 volumeId 값('\$volumeId=get-WfaWorkflowParameter-Name volumeId')을 검색할 수 있습니다.

WFA 명령은 WFA 데이터베이스를 쿼리하고 필요한 결과를 가져올 수 있습니다. 이렇게 하면 필터와 파인더를 사용하지 않고 명령을 구성할 수 있습니다. 다음 함수를 사용하여 데이터베이스를 쿼리할 수 있습니다.

PowerShell cmdlet	Perl 함수
Invoke - MySqlQuery 예: "Invoke - MySqlQuery - Query" cm_storage.cluster ""에서 cluster.name 를 'Cluster Name'으로 선택합니다	invokeMySqlQuery 를 선택합니다

- 관련 정보 \*

## 필터를 만듭니다

### 파인더를 생성합니다

### 명령을 생성합니다

## 필터란

워크플로우에서 WFA 필터를 사용하여 필요한 리소스를 선택할 수 있습니다.

WFA 필터는 WFA 데이터베이스에서 작동하는 SQL 기반 쿼리입니다. 각 필터는 특정 사전 형식의 요소 목록을 반환합니다. 반환된 요소는 SQL 쿼리에 지정된 선택 조건을 기반으로 합니다. 필터를 만들거나 편집하려면 SQL 구문을 알고 있어야 합니다.

## 어떤 장애물이 있는지 확인합니다

파인더는 일반적인 결과를 식별하기 위해 함께 사용되는 하나 이상의 필터를 조합한 것입니다. 워크플로우의 파인더를 사용하여 워크플로우 실행에 필요한 리소스를 선택할 수 있습니다.

파인더는 정렬 순서를 적용하여 해당 결과를 구분할 수 있습니다. 파인더는 선택 기준 및 정렬을 기반으로 최적의 리소스를 반환합니다.

파인더는 결과를 하나 또는 전혀 반환하지 않으므로 특정 스토리지 요소의 존재 여부를 확인하는 데 사용할 수 있습니다. 그러나 finder 를 반복 행 정의의 일부로 사용하면 결과 집합을 사용하여 그룹의 멤버 목록을 구성할 수 있습니다. 파인더에 사용되는 필터는 최소한 사전 형식의 기본 키를 반환하지만 값을 참조할 수 있는 추가 필드를 반환할 수 있습니다. 정렬 순서는 필터 SQL 쿼리의 반환된 필드에 적용될 수 있습니다.

파인더의 결과를 테스트할 수 있습니다. 파인더를 테스트할 때 모든 WFA 필터의 일반적인 결과를 볼 수 있으며, 검색 결과에서 파인더의 유효 결과가 강조 표시됩니다. 워크플로우에서 파인더를 사용할 경우 스토리지 운영자에게 의미 있는 정보를 전달하기 위해 사용자 지정 오류 메시지를 생성할 수 있습니다.

## 기능은 무엇입니까

워크플로의 계획 단계에서 완료해야 하는 복잡한 작업에 대해 워크플로의 함수를 사용할 수 있습니다.

MVFLEX Expression Language(MVEL)를 사용하여 함수를 작성할 수 있습니다. 함수를 사용하면 일반적으로 사용되는 논리는 물론 명명된 함수에 더 복잡한 논리를 결합하여 명령 매개 변수나 필터 매개 변수의 값으로 다시 사용할 수 있습니다. 함수를 한 번 작성하여 워크플로 전체에서 사용할 수 있습니다. 함수를 사용하면 복잡한 명명 규칙을 정의하는 등 복잡한 반복 작업과 작업을 처리할 수 있습니다.

함수는 실행 중에 다른 함수를 사용할 수 있습니다.

## 구성표는 무엇입니까

스키마는 시스템의 데이터 모델을 나타냅니다. 데이터 모델은 사전 항목의 모음입니다. 스키마를 정의한 다음 데이터 소스 형식을 정의할 수 있습니다. 데이터 원본은 데이터 획득 방법과 스키마가 채워지는 방법을 정의합니다. 예를 들어 VC 스키마는 가상 머신, 호스트 및 데이터 저장소와 같은 가상 환경에 대한 데이터를 가져옵니다.

또한 특정 문제를 해결하도록 사용자 지정된 워크플로우를 통해 스키마에 데이터를 직접 채울 수 있습니다.

사전 항목은 사전 항목을 만들 때 기존 스키마와 연결됩니다. 사전 항목은 캐시 쿼리에도 연결되며 캐시 쿼리에는 SQL 쿼리가 포함됩니다.

스키마는 스크립트 기반 데이터 소스 유형 또는 SQL 데이터 소스 유형을 사용하여 데이터를 획득할 수 있습니다. 데이터 소스 형식을 만드는 동안 스크립트가 정의되고 SQL 쿼리가 캐시 쿼리에 정의됩니다.

WFA에는 다음 구성표가 포함되어 있습니다.

- \* 7-Mode (스토리지) \*

7-Mode로 운영되는 Data ONTAP에서 Active IQ Unified Manager를 통해 데이터를 획득하기 위한 구성표,

- \* clustered Data ONTAP(cm\_storage) \*

Clustered Data ONTAP에서 Active IQ Unified Manager를 통해 데이터를 취득하기 위한 계획

- \* 7-Mode 성능(성능) \*

Performance Advisor를 통해 7-Mode로 운영되는 Data ONTAP의 성능 데이터를 획득하기 위한 계획

- \* clustered Data ONTAP 성능(cm\_performance) \*

Performance Advisor를 통해 clustered Data ONTAP의 성능 데이터를 획득하기 위한 계획

- \* VMware vCenter(VC) \*

VMware vCenter에서 데이터를 획득하기 위한 체계입니다.

- \* 운동장(운동장) \*

데이터를 직접 채울 수 있는 구성표

## 어떤 원격 시스템 유형인지 확인하십시오

OnCommand Workflow Automation(WFA)는 원격 시스템 유형과 통신합니다. 원격 시스템 유형은 WFA가 통신할 수 있는 원격 시스템의 유형을 지정합니다. WFA에서 원격 시스템 유형을 구성할 수 있습니다. 예를 들어 Data ONTAP 시스템을 원격 시스템 유형으로 구성할 수 있습니다.

원격 시스템 유형에는 다음과 같은 특성이 있습니다.

- 이름
- 설명
- 버전
- 프로토콜
- 포트
- 시간 초과

각 원격 시스템 유형에 대해 Perl 스크립트를 사용하여 원격 시스템의 자격 증명을 검증할 수 있습니다. WFA에 구성된 원격 시스템에 대한 자격 증명을 저장할 수 있습니다. 새 사용자 지정 원격 시스템 유형을 추가하거나 편집할 수 있습니다. 기존 원격 시스템 유형을 복제할 수도 있습니다. 연결된 시스템이 없는 경우에만 원격 시스템 유형을 삭제할 수 있습니다.

## 서식 파일 사용 방법

WFA 템플릿을 레퍼런스 또는 사용 정책에 따라 워크플로우에 사용할 수 있습니다.

WFA 템플릿은 객체 정의의 청사진 역할을 합니다. 객체의 속성과 개체의 속성 값을 포함하여 템플릿을 정의할 수 있습니다. 그런 다음 템플릿을 사용하여 워크플로에서 개체 정의의 속성을 채울 수 있습니다.

서식 파일을 사용할 때는 서식 파일에서 가져온 값을 포함하는 필드를 편집할 수 없습니다. 따라서 템플릿을 사용하여 사용 정책을 설정하고 개체를 만들 수 있습니다. 템플릿을 적용한 후 템플릿과 워크플로의 연결을 제거하면 템플릿에 입력된 값은 그대로 유지되지만 필드를 편집할 수 있습니다.

## 범주를 사용하는 방법

워크플로를 분류하여 워크플로를 보다 효율적으로 구성하고 워크플로에 액세스 제어 기능을 적용할 수 있습니다.

WFA 포털에서 특정 그룹에 표시되도록 워크플로우를 분류할 수 있습니다. 워크플로 범주에 액세스 제어 기능을 적용할 수도 있습니다. 예를 들어, 특정 스토리지 운영자 또는 승인자만 특정 워크플로우 범주를 볼 수 있도록 허용할 수 있습니다. 스토리지 운영자 또는 승인자는 액세스 권한이 부여된 범주 내의 워크플로우만 실행할 수 있습니다.

Active Directory 그룹을 사용하여 범주에 대한 액세스 제어를 수행할 수도 있습니다.

## 엔터티 버전 관리 작동 방식

명령 및 워크플로우와 같은 WFA(OnCommand Workflow Automation) 엔터티의 버전은 분산됩니다. 버전 번호를 사용하여 WFA 엔터티의 변경 사항을 쉽게 관리할 수 있습니다.

각 WFA 엔터티에는 1.1.20과 같이 'jor.minor.revision' 형식의 버전 번호가 포함되어 있습니다. 버전 번호의 각 부분에 최대 3자리 숫자를 포함할 수 있습니다.

WFA 엔터티의 버전 번호를 수정하기 전에 다음 규칙을 알고 있어야 합니다.

- 버전 번호는 현재 버전에서 이전 버전으로 변경할 수 없습니다.
- 버전 각 부분은 0에서 999 사이의 숫자여야 합니다.
- 새로운 WFA 엔터티의 버전은 기본적으로 1.0.0으로 관리됩니다.
- 개체의 버전 번호는 복제 시 또는 \* 다른 이름으로 저장 \* 을 사용하여 개체의 복사본을 저장할 때 유지됩니다.
- WFA 설치 버전에는 여러 버전의 엔터티가 포함되어 있을 수 없습니다.

WFA Entity의 버전을 업데이트하면 해당 상위 엔터티의 버전이 자동으로 업데이트됩니다. 예를 들어, \* 볼륨 생성 \* 명령의 버전을 업데이트하면 \* NFS 볼륨 생성 \* 워크플로우가 \* 볼륨 생성 \* 명령의 직접적인 상위 엔터티이므로 \* NFS 볼륨 생성 \* 워크플로우가 업데이트됩니다. 버전에 대한 자동 업데이트는 다음과 같이 적용됩니다.

- 엔티티의 주 버전을 수정하면 직접 실행 모체 엔티티의 부 버전이 업데이트됩니다.
- 요소의 부 버전을 수정하면 직접 실행 모체 요소의 수정본 버전이 업데이트됩니다.
- 요소의 수정본 버전을 수정해도 그 직계 모체 요소의 버전 일부가 업데이트되지 않습니다.

다음 표에는 WFA 엔터티와 해당 상위 엔터티가 나와 있습니다.

요소	직접 모체 요소
캐시 쿼리입니다	<ul style="list-style-type: none"><li>데이터 소스 유형입니다</li></ul>
템플릿	<ul style="list-style-type: none"><li>워크플로우</li></ul>
기능	<ul style="list-style-type: none"><li>워크플로우</li><li>템플릿</li></ul>



함수에 특수 또는 혼합 대/소문자 문자가 포함된 경우 해당 상위 요소의 직접 버전은 업데이트되지 않을 수 있습니다.

요소	직접 모체 요소
사전	<ul style="list-style-type: none"> <li>템플릿</li> <li>필터</li> <li>캐시 쿼리입니다</li> <li>명령</li> <li>스크립트 메서드를 사용하는 데이터 소스 유형입니다</li> </ul>
명령	<ul style="list-style-type: none"> <li>워크플로우</li> </ul>
필터	<ul style="list-style-type: none"> <li>파인더</li> <li>워크플로우</li> </ul>
파인더	<ul style="list-style-type: none"> <li>워크플로우</li> </ul>
데이터 소스 유형입니다	없음
워크플로우	없음

WFA에서 버전 번호의 일부 또는 전체 버전 번호를 사용하여 엔터티를 검색할 수 있습니다.

모체 요소를 삭제하면, 종속 요소는 보존되고 해당 버전은 삭제 대상으로 업데이트되지 않습니다.

#### 엔터티를 가져올 때 버전 관리 작동 방식

Workflow Automation 2.2 이전 버전에서 요소를 가져오면 기본적으로 엔터티의 버전이 1.0.0으로 지정됩니다. 가져온 엔터티가 WFA 서버에 이미 있는 경우 가져온 엔터티로 기존 엔터티를 덮어씁니다.

가져오는 동안 WFA 엔터티가 변경될 수 있습니다.

- 요소 업그레이드

요소가 최신 버전으로 대체됩니다.

- 요소 롤백

요소가 이전 버전으로 대체됩니다.



요소의 롤백을 수행하면 해당 인접 상위 요소의 버전이 업데이트됩니다.

- 새 요소 불러오기



'.dar' 파일에서 요소를 선택적으로 불러올 수 없습니다.

요소의 차기 버전을 불러오면, 즉시 모체 요소의 버전이 업데이트됩니다.

불러온 모체 요소에 여러 개의 종속 요소가 있을 경우, 종속 요소에 대한 가장 높은 변경 수준(주, 부 또는 수정본)만 모체 요소에 적용됩니다. 다음 예에서는 이 규칙의 작동 방식을 설명합니다.

- 불러온 모체 요소의 경우, 사소한 변경 사항이 있는 종속 요소와 수정본이 변경된 다른 종속 요소가 있으면, 모체 요소에 사소한 변경 사항이 적용됩니다.

모체 버전의 수정본 부분이 증가합니다.

- 불러온 모체 요소의 경우, 주요 변경 사항이 있는 종속 요소와 사소한 변경 사항이 있는 다른 종속 요소가 있으면, 주요 변경 사항이 모체 요소에 적용됩니다.

상위 버전의 부 부분이 증가합니다.

불러온 하위 요소의 버전이 모체 버전에 영향을 주는 방법의 예

WFA에서 다음 워크플로우를 고려하십시오. "Create Volume and export using NFS-Custom" 1.0.0.

워크플로에 포함된 기존 명령은 다음과 같습니다.

- 수출정책 만들기-사용자 정의 1.0.0
- Create Volume - Custom" 1.0.0

가져올 .dar 파일에 포함된 명령어는 다음과 같다.

- 수출정책 만들기-사용자지정 1.1.0
- Create Volume - Custom" 2.0.0

이 ".dar" 파일을 가져오면 "Create Volume and export using NFS-Custom" 워크플로우의 부 버전이 1.1.0으로 증가합니다.

## 운동장 데이터베이스입니다

운동장 데이터베이스는 MySQL 데이터베이스로, WFA(Workflow Automation) 서버 설치에 포함됩니다. 운동장 데이터베이스에 테이블을 추가하여 정보를 포함할 수 있습니다. 이러한 정보는 필터 및 SQL 쿼리에서 사용자 입력에 사용할 수 있습니다.

운동장 데이터베이스는 WFA 웹 포털을 통해 액세스할 수 없는 스키마입니다. SQLyog, MySQL용 Toad, MySQL Workbench 또는 CLI(명령줄 인터페이스)와 같은 MySQL 클라이언트를 사용하여 데이터베이스에 액세스할 수 있습니다.

운동장 데이터베이스에 액세스하려면 다음 자격 증명을 사용해야 합니다.

- 사용자 이름: WFA
- 암호: Wfa123

자격 증명을 통해 운동장 데이터베이스에 완벽하게 액세스하고 WFA MySQL 데이터베이스에 정의된 다른 스키마에 읽기 전용 액세스를 제공할 수 있습니다. 운동장 데이터베이스에 필요한 테이블을 만들 수 있습니다.

환경의 스토리지 개체에 사용 중인 태그나 메타데이터를 놀이터 데이터베이스의 테이블에 추가할 수 있습니다. 그런 다음 WFA 필터 및 사용자 입력 쿼리를 통해 태그나 메타데이터를 다른 WFA 캐시 표의 정보와 함께 사용할 수

있습니다.

예를 들어 다음과 같은 사용 사례에서 운동장 데이터베이스를 사용할 수 있습니다.

- BU(Business Unit) 이름으로 애그리게이트를 태깅하고 이러한 태그를 기반으로 볼륨을 버스에 할당합니다
- vFiler 유닛을 BU 이름으로 태그 지정
- 스토리지 객체에 지역 또는 위치 세부 정보 추가
- 데이터베이스 관리자의 데이터베이스 액세스 정의

예를 들어 BU의 이름을 애그리게이트 및 vFiler 유닛과 같은 스토리지 객체의 태그로 사용하는 경우, BU 이름이 포함된 테이블을 놀이터 데이터베이스에 생성할 수 있습니다. 그런 다음 BU 이름을 워크플로에 대한 필터 및 사용자 입력 쿼리에서 사용할 수 있습니다.

다음은 Playground 데이터베이스 테이블(Playground.volume\_bu)의 예입니다.

Array_IP입니다	볼륨_이름	사업부
10.225.126.23	데이터_11	마케팅
10.225.126.28	아치_11	시간

다음은 BU로 볼륨을 필터링하는 데 사용할 수 있는 SQL 쿼리의 예입니다.

```
SELECT
    vol.name,
    array.ip AS 'array.ip'
FROM
    storage.volume AS vol,
    storage.array AS array,
    playground.volume_bu AS vol_bu
WHERE
    vol.array_id = array.id
    AND array.ip = vol_bu.array_ip
    AND vol.name = vol_bu.volume_name
    AND vol_bu.bu = '{$bu}'
```

- 관련 정보 \*

"SQLyog"

"MySQL 워크벤치"

"MySQL용 TOAD"

## 저작권 정보

Copyright © 2025 NetApp, Inc. All Rights Reserved. 미국에서 인쇄됨 본 문서의 어떠한 부분도 저작권 소유자의 사전 서면 승인 없이는 어떠한 형식이나 수단(복사, 녹음, 녹화 또는 전자 검색 시스템에 저장하는 것을 비롯한 그레픽, 전자적 또는 기계적 방법)으로도 복제될 수 없습니다.

NetApp이 저작권을 가진 자료에 있는 소프트웨어에는 아래의 라이센스와 고지사항이 적용됩니다.

본 소프트웨어는 NetApp에 의해 '있는 그대로' 제공되며 상품성 및 특정 목적에의 적합성에 대한 명시적 또는 묵시적 보증을 포함하여(이에 제한되지 않음) 어떠한 보증도 하지 않습니다. NetApp은 대체품 또는 대체 서비스의 조달, 사용 불능, 데이터 손실, 이익 손실, 영업 중단을 포함하여(이에 국한되지 않음), 이 소프트웨어의 사용으로 인해 발생하는 모든 직접 및 간접 손해, 우발적 손해, 특별 손해, 징벌적 손해, 결과적 손해의 발생에 대하여 그 발생 이유, 책임론, 계약 여부, 엄격한 책임, 불법 행위(과실 또는 그렇지 않은 경우)와 관계없이 어떠한 책임도 지지 않으며, 이와 같은 손실의 발생 가능성이 통지되었다 하더라도 마찬가지입니다.

NetApp은 본 문서에 설명된 제품을 언제든지 예고 없이 변경할 권리를 보유합니다. NetApp은 NetApp의 명시적인 서면 동의를 받은 경우를 제외하고 본 문서에 설명된 제품을 사용하여 발생하는 어떠한 문제에도 책임을 지지 않습니다. 본 제품의 사용 또는 구매의 경우 NetApp에서는 어떠한 특허권, 상표권 또는 기타 지적 재산권이 적용되는 라이센스도 제공하지 않습니다.

본 설명서에 설명된 제품은 하나 이상의 미국 특허, 해외 특허 또는 출원 중인 특허로 보호됩니다.

제한적 권리 표시: 정부에 의한 사용, 복제 또는 공개에는 DFARS 252.227-7013(2014년 2월) 및 FAR 52.227-19(2007년 12월)의 기술 데이터-비상업적 품목에 대한 권리(Rights in Technical Data -Noncommercial Items) 조항의 하위 조항 (b)(3)에 설명된 제한사항이 적용됩니다.

여기에 포함된 데이터는 상업용 제품 및/또는 상업용 서비스(FAR 2.101에 정의)에 해당하며 NetApp, Inc.의 독점 자산입니다. 본 계약에 따라 제공되는 모든 NetApp 기술 데이터 및 컴퓨터 소프트웨어는 본질적으로 상업용이며 개인 비용만으로 개발되었습니다. 미국 정부는 데이터가 제공된 미국 계약과 관련하여 해당 계약을 지원하는 데에만 데이터에 대한 전 세계적으로 비독점적이고 양도할 수 없으며 재사용이 불가능하며 취소 불가능한 라이센스를 제한적으로 가집니다. 여기에 제공된 경우를 제외하고 NetApp, Inc.의 사전 서면 승인 없이는 이 데이터를 사용, 공개, 재생산, 수정, 수행 또는 표시할 수 없습니다. 미국 국방부에 대한 정부 라이센스는 DFARS 조항 252.227-7015(b)(2014년 2월)에 명시된 권한으로 제한됩니다.

## 상표 정보

NETAPP, NETAPP 로고 및 <http://www.netapp.com/TM>에 나열된 마크는 NetApp, Inc.의 상표입니다. 기타 회사 및 제품 이름은 해당 소유자의 상표일 수 있습니다.