**Oracle® ZFS Storage Appliance RESTful
API 설명서, 릴리스 OS8.8.x**

ORACLE®

부품 번호: F13772-12
2021년 8월

부품 번호: F13772-12

Copyright © 2014, 2021, Oracle and/or its affiliates.

본 소프트웨어와 관련 문서는 사용 제한 및 기밀 유지 규정을 포함하는 라이선스 합의서에 의거해 제공되며, 지적 재산법에 의해 보호됩니다. 라이선스 합의서 상에 명시적으로 허용되어 있는 경우나 법규에 의해 허용된 경우를 제외하고, 어떠한 부분도 복사, 복제, 번역, 방송, 수정, 라이선스, 전송, 배포, 진열, 공연, 출판, 또는 시연될 수 없습니다. 상호 운용을 위해 법령상 요청된 경우를 제외하고, 본 소프트웨어를 역 분석, 분해 또는 역 파일링하는 것은 금지됩니다.

여기에 포함된 내용은 사전 공지 없이 변경될 수 있으며 오라클은 동 내용에 대하여 오류가 존재하지 않음을 보증하지 않습니다. 만일 오류를 발견하면 오라클에 서면으로 통지해 주시기 바랍니다.

만일 본 소프트웨어나 관련 문서가 미국 정부기관 혹은 미국 정부기관을 대신하여 라이선스한 개인이나 법인에게 배송되는 경우, 다음 공지사항이 적용됩니다.

U.S. GOVERNMENT END USERS: Oracle programs (including any operating system, integrated software, any programs embedded, installed or activated on delivered hardware, and modifications of such programs) and Oracle computer documentation or other Oracle data delivered to or accessed by U.S. Government end users are "commercial computer software" or "commercial computer software documentation" pursuant to the applicable Federal Acquisition Regulation and agency-specific supplemental regulations. As such, the use, reproduction, duplication, release, display, disclosure, modification, preparation of derivative works, and/or adaptation of i) Oracle programs (including any operating system, integrated software, any programs embedded, installed or activated on delivered hardware, and modifications of such programs), ii) Oracle computer documentation and/or iii) other Oracle data, is subject to the rights and limitations specified in the license contained in the applicable contract. The terms governing the U.S. Government's use of Oracle cloud services are defined by the applicable contract for such services. No other rights are granted to the U.S. Government.

본 소프트웨어 혹은 하드웨어는 다양한 정보 관리 애플리케이션의 일반적인 사용을 목적으로 개발되었습니다. 본 소프트웨어 혹은 하드웨어는 개인적인 상해를 초래할 수 있는 애플리케이션을 포함하여 본질적으로 위험한 애플리케이션에서 사용할 목적으로 개발된 것이 아니며, 그러한 용도로 사용될 수 없습니다. 만일 본 소프트웨어 혹은 하드웨어를 위험한 애플리케이션에서 사용할 경우, 라이선스 사용자는 해당 애플리케이션의 안전한 사용을 위해 모든 적절한 비상-안전, 백업, 대비 및 기타 조치를 반드시 취해야 합니다. Oracle Corporation과 그 자회사는 본 소프트웨어 혹은 하드웨어를 위험한 애플리케이션에서 사용함으로써 인하여 발생하는 어떠한 손해에 대해서도 책임을 부담하지 않습니다.

Oracle과 Java는 Oracle Corporation 및/또는 그 자회사의 등록 상표입니다. 기타의 명칭들은 각 해당 명칭을 소유한 회사의 상표일 수 있습니다.

Intel 및 Intel Inside는 Intel Corporation의 상표 내지 등록상표입니다. SPARC 상표 일체는 라이선스에 의거하여 사용되며 SPARC International, Inc.의 상표 내지 등록상표입니다. AMD, Epyc, 및 AMD 로고는 Advanced Micro Devices의 상표 내지 등록상표입니다. UNIX는 The Open Group의 등록상표입니다.

본 소프트웨어 혹은 하드웨어와 관련문서(설명서)는 제3자 콘텐츠, 제품 및 서비스에 대한 접속 내지 정보를 제공할 수 있습니다. 사용자와 오라클 간의 합의서에 별도로 규정되어 있지 않는 한 Oracle Corporation과 그 자회사는 제3자의 콘텐츠, 제품 및 서비스와 관련하여 어떠한 책임도 지지 않으며 그에 대한 일체의 보증을 명시적으로 부인합니다. Oracle Corporation과 그 자회사는 제3자의 콘텐츠, 제품 및 서비스에 접속 내지 이를 사용함으로써 인하여 초래되는 어떠한 손실, 비용 또는 손해에 대해 어떠한 책임도 부담하지 않습니다. 단, 사용자와 오라클 간의 합의서에 규정되어 있는 경우는 예외입니다.

설명서 접근성

오라클의 접근성 개선 노력에 대한 자세한 내용은 <http://www.oracle.com/pls/topic/lookup?ctx=acc&id=docacc>에서 Oracle Accessibility Program 웹 사이트를 방문하십시오.

오라클 고객지원센터 액세스

지원 서비스를 구매한 오라클 고객은 My Oracle Support를 통해 온라인 지원에 액세스할 수 있습니다. 자세한 내용은 <http://www.oracle.com/pls/topic/lookup?ctx=acc&id=info>를 참조하거나, 청각 장애가 있는 경우 <http://www.oracle.com/pls/topic/lookup?ctx=acc&id=trs>를 방문하십시오.

목차

Oracle ZFS 스토리지 어플라이언스 RESTful API 시작하기	15
RESTful API 인증	15
RESTful API 버전	16
RESTful API 서비스 버전	16
RESTful API 서비스 버전 2.0	17
스크립트 가능 값	17
일관된 값	18
일반 RESTful 작업	19
HTTP 응답 본문	19
HTTP 응답 헤더	20
질의 매개변수	20
질의 매개변수: props	21
질의 매개변수: start	21
질의 매개변수: end	22
질의 매개변수: limit	22
질의 매개변수: depth	22
질의 매개변수: match	24
어플라이언스 오류	24
보안 프로토콜 및 암호화 설정	25
암호 복잡성	27
RESTful API 작업	29
서비스 액세스	29
서비스 나열	29
서비스 작업 나열	30
인증 토큰	31
비지속적 로그인 토큰 만들기	32
비지속적 로그인 토큰 로그아웃 및 삭제	32

인증서 관리	33
인증서 나열	34
PEM 형식으로 인증서를 반환합니다.	37
서버 인증서 만들기	38
요청 템플릿 반환	38
요청 채우기 및 업로드	39
CA에 요청 전송	41
키 또는 인증서 업로드	41
인증서를 신뢰해야 하는 서비스 지정	43
시스템 기본 인증서 설정	44
인증서 삭제	44
HTTP Strict Transport Security 사용	44
RESTful API 경보 서비스	47
경보 서비스 명령	47
경보 조치	48
모든 경보 조치 나열	50
단일 경보 조치 나열	51
경보 조치 만들기	51
경보 조치 수정	52
이벤트에 대한 응답 지정	53
이벤트에 대한 응답 수정	54
이벤트에 대한 응답 삭제	54
경보 조치 삭제	54
사용자정의 경보	54
사용자정의 경보 생성 및 게시에 필요한 권한 부여 구성	55
사용자정의 경보 만들기	55
사용자정의 경보 게시	56
임계값 경보	56
임계값 경보 나열	57
임계값 경보 만들기	58
임계값 경보 수정	59
임계값 경보 삭제	59
Analytics Services	61
Analytics 명령	61
Analytics 설정	62
설정 가져오기	62

설정 수정	63
Analytics 워크시트	63
워크시트 나열	64
Analytics 워크시트 가져오기	64
워크시트 만들기	65
워크시트 이름 바꾸기	65
워크시트 삭제	66
워크시트 데이터 세트 나열	66
워크시트 데이터 세트 추가	67
워크시트 데이터 세트 수정	67
Analytics 데이터 세트	68
데이터 세트 나열	70
데이터 세트 가져오기	70
데이터 세트 만들기	71
데이터 세트 수정	72
데이터 세트 삭제	72
데이터 세트 저장	72
데이터 세트 데이터 제거	73
데이터 세트 데이터 가져오기	73
하드웨어 서비스	77
클러스터	77
클러스터 등록 정보 나열	77
클러스터 리소스 수정	78
클러스터 링크 상태	79
클러스터 관리 명령	80
클러스터 설정	80
새시	81
새시 나열	81
새시 구성요소 가져오기	82
하드웨어 구성요소 가져오기	84
구성요소 등록 정보 수정	85
로그 명령	87
로그 관리 명령	87
로그 나열	87
로그 항목 가져오기	88
로그 다운로드	90

로그 다운로드	90
네트워크 명령	91
네트워크 구성	91
네트워크 데이터 링크	91
네트워크 데이터 링크 나열	93
네트워크 데이터 링크 가져오기	94
네트워크 데이터 링크 만들기	94
네트워크 데이터 링크 수정	95
네트워크 데이터 링크 삭제	95
네트워크 장치	96
네트워크 장치 나열	96
네트워크 장치 가져오기	97
네트워크 인터페이스	97
네트워크 인터페이스 나열	98
네트워크 인터페이스 가져오기	99
네트워크 인터페이스 만들기	99
네트워크 인터페이스 수정	100
네트워크 인터페이스 삭제	100
네트워크 경로	101
경로 나열	101
경로 가져오기	102
경로 추가	103
경로 삭제	103
RESTful API 클라우드 서비스	105
클라우드 서비스 작업	105
클라우드 서비스 사용	107
클라우드 서비스 로그 파일 보기	107
클라우드 서비스 등록 정보 나열	107
클라우드 서비스 등록 정보 수정	108
대상 나열	109
대상 만들기	110
대상 수정	111
대상 삭제	111
클라우드 백업 나열	112
클라우드 백업 삭제	114
클라우드 백업 복원	115

작업 나열	116
작업 취소 또는 다시 시작	117
스냅샷 백업 작업	118
스냅샷 백업 나열	119
스냅샷 백업 만들기	119
증분 스냅샷 백업 만들기	120
증분 스냅샷 백업의 부모 찾기	120
스냅샷 백업 삭제	122
RESTful API 문제 서비스	125
문제 서비스 명령	125
문제 나열	125
문제 나열	126
문제 복구	127
문제 통지 일시 중지	127
통지 일시 중지 상태 보기	127
통지 일시 중지	128
통지 재개	128
RESTful API SAN 서비스	129
SAN 개요	129
SAN 개시자	129
개시자 나열	130
개시자 세부정보 가져오기	131
개시자 만들기	131
개시자 수정	132
개시자 삭제	133
개시자 그룹	133
개시자 그룹 나열	134
개시자 그룹 세부정보 가져오기	134
개시자 그룹 만들기	135
개시자 그룹 삭제	135
대상	136
대상 나열	137
대상 세부정보 가져오기	138
대상 만들기	138
대상 수정	139
대상 삭제	139

대상 그룹	140
대상 그룹 나열	140
대상 그룹 가져오기	141
대상 그룹 만들기	141
대상 그룹 삭제	142
서비스 명령	143
서비스 명령	143
서비스 나열	143
서비스 가져오기	144
서비스 상태 변경	145
서비스 구성 수정	145
서비스 리소스	148
RESTful API 스토리지 서비스	149
스토리지 풀 작업	149
풀 나열	149
풀 가져오기	150
풀 구성	151
풀에 스토리지 추가	153
풀에서 스토리지 제거	153
풀 스크러빙	154
풀 구성 해제	155
프로젝트 작업	156
프로젝트 나열	157
프로젝트 등록 정보 가져오기	158
프로젝트 만들기	159
프로젝트 수정	160
프로젝트 삭제	161
프로젝트 사용량	161
파일 시스템 작업	161
파일 시스템 나열	163
파일 시스템 가져오기	164
파일 시스템 만들기	166
파일 시스템 수정	167
파일 시스템 삭제	167
파일 시스템 쿼터 및 사용량	168
LUN 작업	168

LUN을 나열합니다.	169
LUN 가져오기	170
새 LUN 만들기	171
LUN 수정	172
LUN 삭제	173
스냅샷 및 복제 작업	174
스냅샷 나열	177
스냅샷 가져오기	178
스냅샷 만들기	178
스냅샷 이름 바꾸기	179
스냅샷 복제	179
스냅샷 롤백	181
스냅샷 삭제	182
스냅샷 종속 항목 나열	183
스키마	184
등록 정보 나열	185
등록 정보 가져오기	185
등록 정보 만들기	185
등록 정보 수정	186
등록 정보 삭제	186
복제	187
복제 서비스 등록 정보 나열	188
복제 서비스 등록 정보 수정	188
복제 대상	188
복제 대상 나열	189
지정된 복제 대상 나열	190
복제 대상 만들기	190
대상 인증서 확인	191
복제 대상 수정	192
복제 대상 삭제	193
복제 작업	193
일반 작업 인터페이스 사용	193
프로젝트, 파일 시스템 또는 LUN 컨텍스트에서 복제 작업	194
복제 작업 나열	197
복제 작업 가져오기	197
복제 작업 만들기	199
복제 작업 수정	200
복제 작업 진행률 모니터	202

업데이트 취소	203
업데이트 전송	204
복제 작업 삭제	204
복제 패키지	204
복제 소스 나열	208
복제 패키지 나열	208
패키지 수정	209
패키지 삭제	210
업데이트 취소	210
패키지 복제	211
패키지 단절	212
패키지 방향 바꾸기	212
스토리지 암호화	213
암호화 키 관리	213
로컬 키 저장소 구성	214
OKM 키 저장소 구성	215
KMIP 키 저장소 구성	216
암호화 키 만들기	217
암호화 키 나열	217
지정된 키로 암호화된 스토리지 나열	218
키 삭제	218
암호화된 풀, 프로젝트 또는 공유 만들기	218
시스템 명령	221
어플라이언스 시스템 명령	221
버전 가져오기	221
시스템 전원 끄기	222
시스템 재부트	222
시스템 관리 다시 시작	223
진단 재부트	223
공장 초기화 재설정	223
시스템 지원 번들	224
지원 번들 만들기	224
지원 번들 나열	225
지원 번들 가져오기	226
지원 번들 취소	226
지원 번들 업로드 재시도	227

지원 번들 업로드	227
지원 번들 삭제	228
시스템 업데이트	228
시스템 업데이트 나열	229
시스템 업데이트 가져오기	230
플랫폼 펌웨어 업데이트 상태 가져오기	230
구성요소 펌웨어 업데이트 상태 가져오기	231
시스템 업데이트 업로드	231
업그레이드	232
롤백	232
업데이트 이미지 삭제	233
RESTful API 사용자 서비스	235
사용자 서비스 명령	235
사용자 서비스 등록 정보	236
사용자 등록 정보	236
사용자 역할 및 예외사항	237
사용자 환경 설정 등록 정보	238
CLI 시간 초과	239
SSH 키	239
REST 로그인 토큰	239
사용자 나열	240
특정 사용자 나열	241
사용자 만들기	242
사용자 등록 정보 수정	243
사용자 삭제	244
토큰 관리	244
RESTful API 역할 서비스	249
역할 서비스 명령	249
역할 서비스 등록 정보	250
역할 나열	250
역할 가져오기	250
역할 만들기	251
역할 수정	252
역할 취소	252
역할 삭제	253
역할 권한 부여 나열	253

역할 권한 부여 만들기	254
역할 권한 부여 수정	254
역할 권한 부여 삭제	255
워크플로우 및 스크립트 명령	257
워크플로우 및 스크립트 서비스 명령	257
워크플로우 업로드	257
워크플로우 나열	258
워크플로우 가져오기	259
워크플로우 수정	261
워크플로우 실행	261
워크플로우 삭제	262
스크립트 업로드 및 실행	262
실행 중인 모든 스크립트 나열	263
실행 중인 스크립트에 재연결	264
실행 중인 스크립트 중지	265
RESTful 클라이언트	267
Curl REST 클라이언트	267
리소스 데이터 가져오기	267
새 리소스 만들기	268
기존 리소스 수정	268
기존 리소스 삭제	269
Python RESTful 클라이언트	269
리소스 가져오기	270
리소스 만들기	270
리소스 수정	271
기존 리소스 삭제	272

Oracle ZFS 스토리지 어플라이언스 RESTful API 시작하기

Oracle ZFS 스토리지 어플라이언스는 네트워크를 통해 효율적인 파일 및 블록 데이터 서비스를 제공합니다. 이 설명서에서는 어플라이언스를 관리하는 데 사용할 수 있는 Oracle ZFS 스토리지 어플라이언스 RESTful API(응용 프로그램 프로그래밍 인터페이스)에 대해 설명합니다. RESTful 아키텍처는 계층화된 클라이언트-서버 모델을 기반으로 하며, 클라이언트 구성 없이 표준 허브, 라우터 및 기타 네트워크 시스템을 통해 서비스를 투명하게 재지정할 수 있도록 해줍니다.

RESTful API 인증

Oracle ZFS 스토리지 어플라이언스 RESTful API는 BUI(브라우저 사용자 인터페이스) 및 CLI(명령줄 인터페이스)와 동일한 인증 자격 증명을 사용합니다. 외부 클라이언트에서 들어오는 모든 요청은 어플라이언스 자격 증명을 사용하여 개별적으로 인증되며, 포트 215에서 HTTPS 연결을 통해 수행됩니다. RESTful API는 사용자정의 가능 시간 초과가 15분으로 설정된 HTTPS 세션을 지원합니다.

인증은 다음 형식 중 하나를 사용할 수 있습니다.

- 기본 인증 – 각 요청에는 사용자 로그인이 포함되어 있어야 합니다. 권한 부여 문자열은 이후 Base64로 인코딩되는 `username:password`로 연결됩니다.

HTTP 헤더 예:

```
Authorization: Basic Tm8gcGV1a2luZyE=
```

- 사용자 인증 – BUI 또는 CLI 로그인 자격 증명에 인증에 사용됩니다. 이 경우 X-Auth-User 헤더는 로그인 이름을 포함해야 하며 X-Auth-Key 헤더는 로그인 암호를 포함해야 합니다.

HTTP 헤더 예:

```
X-Auth-User: login-name  
X-Auth-Key: password-xxx
```

- 토큰 인증 – 토큰이 인증된 경우 토큰이 만료될 때까지 계속 명령을 실행하는 데 토큰 헤더를 사용할 수 있습니다. 토큰이 만료된 후 명령이 수락되기 전 다시 인증을 수행해야 합니다.

토큰 헤더 예:

```
X-Auth-Session: qYftpufrrTx1DztkMh11LoyTfSDUSIR
```

RESTful API 버전

지정된 어플라이언스 릴리스에 대한 RESTful API 버전에 어플라이언스 소프트웨어 버전과 일치하는 전역 버전 번호가 있습니다. 모든 요청의 응답 헤더에 이 버전 번호가 반환됩니다.

```
X-Zfssa-Version: nas.2013.1.1
```

RESTful API 서비스 버전

서비스에 액세스하기 위한 URI(Uniform Resource Identifier)의 일부분으로 버전 번호가 각 RESTful API 서비스에 포함됩니다. 버전에는 주 번호와 부 번호가 있습니다. 요청은 주 버전 번호를 제공해야 하지만, 부 버전 번호는 선택사항이므로 제공하지 않을 경우 기본적으로 값 0으로 지정됩니다. 주 번호는 서비스의 주 번호와 일치해야 합니다. 부 번호는 서비스의 부 번호보다 작거나 같아야 합니다.

예를 들어, 다음 표에서는 클라이언트에서 버전 2.1이 실행되고 있는 서비스를 요청하는 경우 지정된 버전을 클라이언트 요청에서 사용할 수 있는지 여부를 보여줍니다.

요청 버전	허용됨
1	아니오: 주 버전과 서비스가 실행 중인 버전이 일치하지 않습니다.
2	예: 주 버전이 일치하며 부 버전(기본값: 0)이 이전 버전과 호환됩니다.
2.1	예: 주 및 부 버전 값과 서비스가 실행 중인 버전이 일치합니다.
2.2	아니오: 주 버전은 일치하지만 부 버전은 서비스가 실행 중인 버전보다 최신입니다.

다음 등록 정보 변경에 대해서는 서비스 API 버전 변경이 필요하지 않습니다. 어플라이언스 버전 번호 및 모델은 사용 가능한 등록 정보를 결정하는 데 사용되어야 합니다. 또한 이러한 등록 정보 변경사항은 CLI 및 BUI에 반영되고 해당 어플라이언스 인스턴스의 기능을 표시합니다.

- 이전 등록 정보를 제거하지 않는 새 출력 등록 정보입니다.
- 기존 명령에 추가된 새 입력 등록 정보로, 명령을 이전 버전에서처럼 작동하게 하는 기본값을 포함합니다.

이전 버전과 호환되는 명령의 최신 버전이 추가 등록 정보를 반환할 수 있으므로 클라이언트는 새 등록 정보를 무시하도록 코딩되어야 합니다. 부 번호는 서비스 API에 이전 버전과 호환되는 변경사항이 있을 때 증가합니다.

- 기존 서비스에 새 명령을 추가합니다.
- 서비스 명령에 새 질의 매개변수를 추가합니다.

주 번호는 서비스 API에 호환되지 않는 변경사항이 있을 때 증가합니다.

- 명령 질의 매개변수를 제거합니다.
- 기존 서비스에서 명령을 제거합니다.

어플라이언스 소프트웨어의 주요 릴리스에 호환되지 않는 버전 변경사항이 포함될 수 있습니다. 주요 업데이트 동안 지정된 서비스의 이전 버전이 있거나 없을 수 있습니다. 각 명령 응답은 지정된 모듈에 대한 현재 버전의 어플라이언스 API와 함께 HTTP 헤더를 포함해야 합니다.

X-Zfssa-Nas-API: 1.1

RESTful API 서비스 버전 2.0

이 절에서는 RESTful API 서비스 버전 2와 RESTful API 서비스 버전 1 간의 차이에 대해 설명합니다.

- [“스크립트 가능 값” \[17\]](#)
- [“일관된 값” \[18\]](#)

RESTful API 버전 2와 RESTful API 버전 1은 동시에 사용할 수 있으며 본 설명서의 나머지 부분에서는 버전 1의 예를 보여줍니다. 요청 URI의 서비스 버전 부분(v1 또는 v2)을 통해 사용할 REST API 버전을 선택할 수 있습니다.

스크립트 가능 값

RESTful API 버전 2 작업은 항상 스크립트 가능 값을 반환합니다. 스크립트 가능 값의 형식은 각 등록 정보 유형에 대해 동일한 안정적인 형식입니다.

RESTful API 버전 1 작업은 스크립트 가능 값을 반환하는 경우가 많지만 항상 반환하는 것은 아닙니다. 예를 들어, RESTful API 버전 1은 전체 JavaScript 날짜 형식으로 날짜 시간 문자열을 반환하기도 하고, ISO 8601 날짜 시간 형식으로 날짜 시간 문자열을 반환하기도 합니다. RESTful API 버전 2는 항상 ISO 8601 날짜 시간 형식으로 날짜 시간 문자열을 반환합니다.

다음 예에서 GET /api/system/v1/updates 작업은 전체 JavaScript 날짜 시간 형식으로 날짜 시간 문자열을 반환하고, GET /api/system/v2/updates 작업은 ISO 8601 날짜 시간 형식으로 날짜 시간 문자열을 반환합니다.

```
GET /api/system/v1/updates
{
  "updates": [{
    "status": "previous",
    "href": "/api/system/v1/updates/ak-nas@2013.06.05.4.0,1-1.7",
    "release_date": "Fri May 01 2015 20:13:00 GMT+0000 (UTC)",
```

```

    "install_date": "Tue Nov 15 2016 01:01:07 GMT+0000 (UTC)",
    "version": "2013.06.05.4.0,1-1.7",
    "date": "Fri May 01 2015 20:13:00 GMT+0000 (UTC)"
  }]
}
GET /api/system/v2/updates
{
  "updates": [{
    "status": "previous",
    "href": "/api/system/v2/updates/ak-nas@2013.06.05.4.0,1-1.7",
    "release_date": "2015-05-01T20:13:00Z",
    "install_date": "2016-11-15T01:01:07Z",
    "version": "2013.06.05.4.0,1-1.7",
    "date": "2015-05-01T20:13:00Z"
  }]
}

```

일관된 값

RESTful API 버전 1 작업은 등록 정보에 액세스하는 방식에 따라 동일한 등록 정보에 대해 다른 값을 반환하기도 합니다. RESTful API 버전 2 작업은 등록 정보에 액세스하는 방식에 관계 없이 일관된 값을 반환합니다.

다음 예에서는 모든 복제 작업이 나열될 때 `max_bandwidth` 등록 정보의 값이 `-1`로 반환됩니다.

```

GET /api/storage/v1/replication/actions
{
  "actions": [{
    "id": "71b1b8b9-9c57-c969-aab9-f96d5f4e5d54",
    ...
    "max_bandwidth": -1,
    ...
  }]
}

```

복제 작업이 하나만 지정되면 기본 값이 변경되지 않은 경우에도 `max_bandwidth` 등록 정보의 값이 `0`으로 반환됩니다.

```

GET /api/storage/v1/replication/actions/71b1b8b9-9c57-c969-aab9-f96d5f4e5d54
{
  "action": {
    "id": "71b1b8b9-9c57-c969-aab9-f96d5f4e5d54",
    ...
    "max_bandwidth": 0,
    ...
  }
}

```

RESTful API 버전 2 작업은 등록 정보 값에 액세스하는 방식에 관계없이 항상 특정 등록 정보에 대해 동일한 값을 반환합니다.

```

GET /api/storage/v2/replication/actions
{
  "actions": [{
    "id": "71b1b8b9-9c57-c969-aab9-f96d5f4e5d54",
    ...
    "max_bandwidth": -1,
    ...
  }]
}

```

```

    }]
  }
GET /api/storage/v2/replication/actions/71b1b8b9-9c57-c969-aab9-f96d5f4e5d54
{
  "action": {
    "id": "71b1b8b9-9c57-c969-aab9-f96d5f4e5d54",
    ...
    "max_bandwidth": -1,
    ...
  }
}

```

일반 RESTful 작업

다음 표는 지정된 리소스에 대한 일반 RESTful 작업을 보여줍니다.

표 1 일반 RESTful 작업

요청	경로	설명
GET	resources	모든 리소스를 나열합니다.
GET	resources/name	선택된 리소스를 설명하는 JSON 객체를 가져옵니다.
POST	resources	새 리소스를 만듭니다.
PUT	resources/name	선택된 리소스를 수정합니다.
DELETE	resources/name	선택된 리소스를 삭제합니다.

HTTP 응답 본문

모든 응답 데이터는 [RFC 4627](#)에 의해 정의된 대로 JSON 형식으로 인코딩됩니다. 별도로 지정되지 않는 한 단일 리소스에 대한 명령은 리소스 이름을 등록 정보로 사용하는 단일 JSON 결과 객체를 반환합니다. 각 명령 섹션에서는 이 JSON 결과 객체에 반환된 등록 정보 이름을 문서화합니다.

별도로 설명하지 않는 한 만들기(POST) 및 수정(PUT) 명령은 만들어진 리소스 또는 수정된 리소스의 등록 정보를 반환합니다. 내용은 GET 요청에 의해 반환된 값과 일치해야 합니다.

본문 예:

```

{
  "resource_name": {
    "href": "path/to/this/resource",
    "property_01": "value_01",
    "property_02": "value_01"
  }
}

```

일부 GET 명령은 리소스 목록을 반환합니다.

```
{
  "resource_list_name": [
    {
      "href": "path/to/resource_01",
      "property_01": "value_01"
    }, {
      "href": "path/to/resource_02",
      "property_02": "value_02"
    }
  ]
}
```

주 - 이 문서에서는 명령이 보다 읽기 쉽게 만들기 위해 리턴 키 및 공백을 추가하여 형식이 지정된 JSON 반환 결과를 보여 줍니다. 실제 출력은 이 형식을 포함하지 않습니다.

HTTP 응답 헤더

데이터를 전송하는 모든 어플라이언스 서비스 명령은 JSON 데이터 형식을 사용하고 다음 헤더 값을 필요로 합니다.

```
Accept: application/json
Content-Type: application/json
```

응답 헤더에는 다음 정보가 포함됩니다.

```
Date: Tue, 23 Jul 2013 13:07:37 GMT X-Zfs-Sa-Appliance-API: 1.0 Content-Type: application/json Content-Length: 357
```

목록 결과의 경우 데이터가 다시 전송되기 전에는 내용 길이를 알 수 없습니다. 내용 길이가 제공되지 않는 경우 클라이언트는 EOF까지 응답 본문을 읽어 반환된 모든 데이터를 읽어야 합니다.

질의 매개변수

일부 요청은 반환되거나 작동되는 데이터를 선택할 질의 매개변수(선택사항)를 사용할 수 있습니다. 이 절에서는 둘 이상의 리소스 유형에 사용 가능한 질의 매개변수를 문서화합니다. 해당 리소스와 관련된 질의 매개변수 및 이 절에서 설명되는 질의 매개변수의 특별한 용도는 각 리소스의 설명서를 참조하십시오.

표 2 일반 질의 매개변수

매개변수	설명
props=true	리소스에 대한 등록 정보 메타데이터를 나열합니다. 기본값은 false입니다.
start=index	지정된 시간 이후 또는 객체 ID 뒤에 반환할 가장 오래된 데이터 또는 객체를 지정합니다.

매개변수	설명
<code>end=index</code>	지정된 시간 이전 또는 객체 ID 앞에 반환할 최신 데이터 또는 객체를 지정합니다.
<code>limit=n</code>	n 개 이하의 목록 요소를 반환합니다.
<code>depth=n</code>	반환된 데이터의 세부정보 레벨을 지정합니다.
<code>match_property-name=value</code>	지정된 값의 지정된 등록 정보가 있는 목록 객체를 반환합니다.

질의 매개변수: props

`props` 질의 매개변수는 등록 정보 메타데이터 값을 보여줍니다. 데이터를 변경하거나 새 데이터를 만들 작업에서 `props=true`를 사용하면 등록 정보 및 메타데이터가 표시되고 작업이 수행되지 않습니다. 이를 통해 리소스 수정 또는 생성에 유용한 현재 데이터 값을 표시할 수 있습니다.

표 3 등록 정보 메타데이터 값

등록 정보	설명
<code>name</code>	등록 정보 이름
<code>label</code>	등록 정보의 설명
<code>immutable</code>	등록 정보를 수정할 수 없음을 나타내는 플래그
<code>type</code>	등록 정보 유형(예: String, Integer, Boolean 또는 ChooseOne)
<code>choices</code>	열거된 등록 정보의 경우 사용 가능한 값 배열

질의 매개변수: start

`start` 질의 매개변수는 객체 인덱스 번호 또는 시간일 수 있습니다.

- 해당 인덱스가 선택한 객체 및 지정된 객체가 만들어진 후 만들어진 가장 오래된 객체를 포함하는 목록을 반환할 객체 인덱스 번호를 지정합니다.
- 가장 오래된 객체 또는 지정된 시간 또는 그 이후에 만들어진 데이터 목록을 반환할 UTC 시간을 지정합니다. 일부 리소스는 `start` 질의 매개변수에 대한 시간 값을 지원하지 않습니다.

시간 값은 다음 표에 나타난 형식의 UTC 시간으로 표시되어야 합니다.

서비스 버전	시간 값 형식	시간 값 예
v1 경로	<code>%Y%m%dT%H:%M:%SZ</code>	20200723T14:11:49
v2 경로	<code>%Y-%m-%dT%H:%M:%SZ</code>	2020-07-23T14:11:49Z

`start` 질의 매개변수를 사용하는 예는 다음 절을 참조하십시오.

- “데이터 세트 데이터 가져오기” [73]
- “클라우드 백업 나열” [112]

질의 매개변수: end

`end` 질의 매개변수는 객체 인덱스 번호 또는 시간일 수 있습니다.

- 해당 인덱스가 선택한 객체 및 지정된 객체가 만들어지기 전 만들어진 최신 객체를 포함하는 목록을 반환할 객체 인덱스 번호를 지정합니다.
- 최신 객체 또는 지정된 시간 또는 그 이전에 만들어진 데이터 목록을 반환할 UTC 시간을 지정합니다.

시간 값은 `%Y-%m-%dT%H:%M:%SZ` 형식의 UTC 시간으로 표시되어야 합니다.

`end` 질의 매개변수 사용 예는 “클라우드 백업 나열” [112]을 참조하십시오.

질의 매개변수: limit

`limit` 질의 매개변수는 반환할 최대 요소 수를 지정합니다.

`start`와 `end`도 지정되지 않은 경우 `limit=n`은 가장 최근 n 개 요소를 반환합니다.

질의 매개변수: depth

`depth` 질의 매개변수는 반환된 리소스 목록에 대한 세부정보 레벨을 지정합니다. `depth` 값이 클수록 아래 표에서와 같이 세부정보가 더 많이 반환됩니다.

depth 값	목록에 반환되는 정보
<code>depth=0</code>	노드의 등록 정보 및 자식 항목의 이름
<code>depth=1</code>	노드의 등록 정보, 자식 항목의 이름 및 등록 정보, 2차 자식 항목의 이름
<code>depth=2</code>	노드의 등록 정보, 자식 항목의 이름 및 등록 정보, 2차 자식 항목에 대한 <code>depth=0</code> 출력

주 - `depth` 질의 매개변수는 `/api/log/v{1|2}`를 통해 로그를 나열하거나 `/api/storage/v{1|2}`를 통해 풀, 프로젝트, 파일 시스템 또는 LUN을 나열하는 데 사용할 수 없습니다.

질의 매개변수 `depth`를 통한 사용자 목록에 대한 요청 예:

```
GET /api/user/v1/users?depth=2 HTTP/1.1
Host: zfs-storage.example.com:215
X-Auth-User: root
```

X-Auth-Key: *password-xxx*

응답 예:

간략히 나타내기 위해 추가 사용자는 생략됩니다.

HTTP/1.1 200 OK

Content-Type: application/json

Content-Length: 1558

X-Zfssa-Access-API: 1.0

```
{
  "users": [
    {
      "name": "root",
      ...
    },
    {
      "name": "firstlast",
      "properties": {
        "logname": "firstlast",
        "type": "directory",
        "uid": uid,
        "fullname": "First Last",
        "require_annotation": false,
        "roles": [
          "basic"
        ],
        "kiosk_mode": false,
        "kiosk_screen": "status/dashboard"
      },
      "children": [
        {
          "name": "exceptions",
          "properties": {},
          "children": [],
          "list": [
            {
              "name": "auth-000",
              "properties": {
                "scope": "ad",
                "name": "*",
                "allow_domain": true,
                "allow_workgroup": false
              },
              "children": [],
              "list": []
            },
            {
              "name": "auth-001",
              "properties": {
                "scope": "alert",
                "allow_configure": true,
                "allow_post": true
              },
              "children": [],
              "list": []
            }
          ]
        }
      ],
      {
        "name": "preferences",
        "properties": {
          "locale": "C",

```

```

        "login_screen": "configuration/preferences",
        "session_timeout": 15,
        "cli_idle_timeout": "infinite",
        "advanced_analytics": false
    },
    "children": [
        {
            "name": "keys",
            "properties": {},
            "children": [],
            "list": []
        },
        {
            "name": "tokens",
            "properties": {},
            "children": [],
            "list": []
        }
    ],
    "list": []
}
],
"list": [],
"href": "/api/user/v1/users/firstlast"
},
{
...
}
]
}

```

질의 매개변수: match

`matchproperty-name=value` 질의 매개변수는 지정된 값의 지정된 등록 정보 이름이 있는 리소스 목록을 반환합니다.

다음 예에서는 `kiosk_mode` 등록 정보의 값이 `true`인 사용자 목록을 반환합니다.

```
match_kiosk_mode=true
```

다음 예에서는 `roles` 등록 정보의 값에 `super`가 포함되고 `require_annotation` 등록 정보의 값이 `true`인 사용자 목록을 반환합니다.

```
match_roles='*super*&match_require_annotation=true
```

주 - `match_property-name=value` 질의 매개변수는 `/api/log/v{1|2}`를 통해 로그를 나열하거나 `/api/storage/v{1|2}`를 통해 풀, 프로젝트, 파일 시스템 또는 LUN을 나열하는 데 사용할 수 없습니다.

어플라이언스 오류

오류는 다음 결합 응답 페이로드와 함께 오류를 나타내는 HTTP 상태 코드를 반환합니다.

JSON 결함 응답:

```
{
  fault: {
    message: 'ERR_INVALID_ARG',
    details: 'Error Details...',
    code: 500
  }
}
```

표 4 일반 오류 코드

오류	코드	설명
ERR_INVALID_ARG	400	잘못된 입력 인수입니다.
ERR_UNKNOWN_ARG	400	처리되지 않은 추가 입력 인수입니다.
ERR_MISSING_ARG	400	필수 입력 인수가 누락되었습니다.
ERR_UNAUTHORIZED	401	이 사용자에게는 명령을 실행할 수 있는 권한이 없습니다.
ERR_DENIED	403	작업이 거부되었습니다.
ERR_STATE_CHANGED		시스템 상태가 충돌합니다.
ERR_NOT_FOUND	404	요청된 항목을 찾을 수 없습니다.
ERR_OBJECT_EXISTS	409	요청이 이미 존재하는 객체를 만듭니다.
ERR_CONFIRM_REQUIRED	409	요청을 완료하려면 <code>?confirm=true</code> 질의 매개변수가 필요합니다.
ERR_OVER_LIMIT	413	입력 요청이 너무 커서 처리할 수 없습니다.
ERR_UNSUPPORTED_MEDIA	415	요청된 매체 유형이 요청에서 지원되지 않습니다.
ERR_NOT_IMPLEMENTED	501	작업이 구현되지 않습니다.
ERR_BUSY	503	제한된 리소스로 인해 서비스를 사용할 수 없습니다.

보안 프로토콜 및 암호화 설정

프로토콜 버전 및 연관된 암호화 명령을 통해 어플라이언스 액세스에 필요한 SSL/TLS 프로토콜 버전 및 암호화를 관리할 수 있습니다.

기본적으로 SSL/TLS 프로토콜 버전 TLSv1.1, TLSv1.2 및 연관된 암호화가 사용으로 설정되어 있습니다. TLSv1.0은 HTTPS 서비스로 PUT 요청을 전송하여 `tls_version` 등록 정보 설정을 통해 사용으로 설정할 수 있습니다.

요청 예:

```
PUT /api/service/v1/services/https HTTP/1.1
Host: zfs-storage.example.com:215
Content-Type: application/json
```

```
{ "tls_version": ["TLSv1.0", "TLSv1.1", "TLSv1.2"] }
```

예제 결과(간단한 설명을 위해 출력이 생략됨):

```
HTTP/1.1 202 Accepted
Content-Length: 1265
X-Zfssa-Service-API: 1.1
X-Zfssa-API-Version: 1.0
Content-Type: application/json; charset=utf-8

{
  "service": {
    "href": "/api/service/v1/services/https",
    "<status>": "online",
    "tls_version": "TLSv1 TLSv1.1 TLSv1.2",
    "ciphers": "SRP-DSS-AES-256-CBC-SHA:SRP-RSA-AES-256-CBC-SHA:SRP-AES-256-CBC-SHA:
    ...
    3DES-EDE-CBC-SHA:EDH-RSA-DES-CBC3-SHA:EDH-DSS-DES-CBC3-SHA:DH-RSA-DES-CBC3-SHA:
    DH-DSS-DES-CBC3-SHA:DES-CBC3-SHA"
  }
}
```

TLSv1.0만 사용으로 설정하려면 `ciphers` 등록 정보를 TLSv1.0에 대해서만 사용 가능한 암호화 목록으로 설정합니다.

예제 요청(간단한 설명을 위해 출력이 생략됨):

```
PUT /api/service/v1/services/https HTTP/1.1
Host: zfs-storage.example.com:215
Content-Type: application/json

{
  "tls_version": ["TLSv1.0"],
  "ciphers": ["SRP-DSS-AES-256-CBC-SHA", "SRP-RSA-AES-256-CBC-SHA", "SRP-AES-256-CBC-SHA",
  ...
  "EDH-RSA-DES-CBC3-SHA", "EDH-DSS-DES-CBC3-SHA", "DH-RSA-DES-CBC3-SHA", "DH-DSS-DES-CBC3-SHA",
  "DES-CBC3-SHA"]
}
```

예제 결과(간단한 설명을 위해 출력이 생략됨):

```
HTTP/1.1 202 Accepted
Content-Length: 809
X-Zfssa-Service-API: 1.1
X-Zfssa-API-Version: 1.0
Content-Type: application/json; charset=utf-8

{
  "service": {
    "href": "/api/service/v1/services/https",
    "<status>": "online", "tls_version": "TLSv1",
    "ciphers": "SRP-DSS-AES-256-CBC-SHA:SRP-RSA-AES-256-CBC-SHA:SRP-AES-256-CBC-SHA:
    ...
    3DES-EDE-CBC-SHA:SRP-3DES-EDE-CBC-SHA:EDH-RSA-DES-CBC3-SHA:EDH-DSS-DES-CBC3-SHA:DH-
    RSA-DES-CBC3-SHA:DH-DSS-DES-CBC3-SHA:DES-CBC3-SHA"
  }
}
```

주 - RESTful API 또는 BUI 사용이 차단되지 않도록 하려면 오라클 고객지원센터의 요구 또는 지침이 없는 한 `tls_version` 및 `ciphers` 등록 정보에 대한 기본 설정을 유지하십시오.

암호 복잡성

암호 RESTful API는 `changeProperties` 권한이 부여된 사용자가 모든 로컬 사용자에게 대해 암호 복잡성 규칙을 설정할 수 있게 해줍니다. 사용자 권한 부여에 대한 자세한 내용은 [RESTful API 역할 서비스 \[249\]](#)를 참조하십시오.

암호 요구사항은 로컬 사용자가 암호를 변경할 때 적용됩니다. 기존 암호는 암호 규칙 변경사항의 영향을 받지 않습니다.

다음 명령을 사용하여 변경 가능한 등록 정보를 표시하십시오.

요청 예:

```
GET /api/setting/v2/password HTTP/1.1
Host: zfs-storage.example.com:215
Authorization: Basic Tm8gcGV1a2luZyE=
Accept: application/json
```

결과 예:

```
HTTP/1.1 200 OK
Date: Fri, 14 May 2021 17:07:39 GMT
Content-Type: application/json; charset=utf-8
X-Zfssa-Api-Version: 2.0
X-Zfssa-Setting-Api: 2.0
Content-Length: 196
```

```
{
  "complexity": {
    "href": "/api/setting/v2/password",
    "passlength": 8,
    "min_letters": 0,
    "min_upper": 0,
    "min_lower": 0,
    "min_digit": 0,
    "min_punctuation": 0,
    "max_repeats": 0,
    "namecheck": true
  }
}
```

이러한 등록 정보에 대한 설명은 [“Password Complexity Properties” in Oracle ZFS Storage Appliance Administration Guide, Release OS8.8.x](#)를 참조하십시오.

다음 예제는 대문자, 소문자, 숫자, 문장 부호 문자를 최소 하나 이상 요구하도록 암호 규칙을 변경합니다. 새 `min_upper` 및 `min_lower` 값을 고려하도록 `min_letters` 값을 업데이트해야 합니다.

요청 예:

```
PUT /api/setting/v2/password HTTP/1.1
Host: zfs-storage.example.com:215
Content-Type: application/json
```

```
{"min_letters": 2, "min_upper": 1, "min_lower": 1, "min_digit": 1, "min_punctuation": 1}
```

결과 예:

```
HTTP/1.1 202 Accepted
Date: Fri, 14 May 2021 17:38:40 GMT
Content-Type: application/json; charset=utf-8
X-Zfssa-API-Version: 2.0
X-Zfssa-Setting-API: 2.0
Content-Length: 196
```

```
{
  "complexity": {
    "href": "/api/setting/v2/password",
    "passlength": 8,
    "min_letters": 2,
    "min_upper": 1,
    "min_lower": 1,
    "min_digit": 1,
    "min_punctuation": 1,
    "max_repeats": 0,
    "namecheck": true
  }
}
```

RESTful API 작업

서비스 액세스는 Oracle ZFS 스토리지 어플라이언스의 모든 RESTful API 서비스에 대한 시작점입니다. 이 서비스는 사용자 자격 증명을 인증하고 버전 및 액세스 지점을 포함하여 사용 가능한 RESTful API 서비스를 나열하는 데 사용됩니다.

서비스 액세스

서비스에 액세스하려면 다음 URL 중 하나를 사용하십시오.

- `http://hostname:215/api/access/v2`
- `http://hostname:215/api/access/v1`

다른 서비스에 액세스하려면 액세스 서비스를 사용하여 로그인하여 사용 가능한 서비스의 위치 및 버전을 가져온 다음 반환된 URI를 사용하여 이러한 서비스에 액세스합니다. 서비스 위치는 현재 어플라이언스 구성 또는 릴리스 레벨을 기준으로 변경될 수 있습니다.

표 5 서비스 액세스 명령

요청	경로	설명
GET	<code>/api/access/v{1 2}</code>	RESTful API 서비스 액세스 지점을 나열합니다.
POST	<code>/api/access/v{1 2}</code>	비지속적 로그인 토큰을 만듭니다.
DELETE	<code>/api/access/v{1 2}</code>	비지속적 로그인 토큰을 로그아웃한 후 삭제합니다.

서비스 나열

`list services` 명령은 사용 가능한 서비스 액세스 URI를 나열합니다. 로그인 세션을 원하지 않는 경우 `list services` 명령을 적합한 자격 증명과 함께 사용하여 사용 가능한 서비스 액세스 URI를 나열할 수 있습니다. 이 명령은 해당 어플라이언스에서 사용할 수 있는 모든 RESTful API 서비스 및 버전을 나열합니다.

요청 예:

```
GET /api/access/v1 HTTP/1.1
Host: zfs-storage.example.com:215
X-Auth-User: admin1
X-Auth-Key: password
```

결과 예:

```
HTTP/1.1 200 OK
Content-Type: application/json
Content-Length: 190
X-Zfssa-Access-API: 1.0
```

```
{
  "services": [
    {
      "version": "1.1",
      "name": "hardware",
      "uri": "https://hostname:215/api/hardware/v1"
    },
    {
      "version": "1.0",
      "name": "san",
      "uri": "https://hostname:215/api/san/v1"
    },
    {
      "version": "1.0",
      "name": "network",
      "uri": "https://hostname:215/api/network/v1"
    }
  ]
}
```

서비스 작업 나열

이 명령은 지정된 서비스에 사용 가능한 작업(메소드) 목록을 반환합니다. 해당하는 경우 이 명령은 지정된 서비스의 리소스에 대한 정보를 반환합니다. 다음 예에서는 클러스터 하드웨어 구성요소에 이후 검사 및 구성 가능한 리소스가 포함됩니다.

요청 예. 이 요청은 인증 토큰을 사용합니다. [“인증 토큰” \[31\]](#)을 참조하십시오.

```
GET /api/hardware/v1 HTTP/1.1
Host: zfs-storage.example.com:215
X-Auth-Session: puPnHZKgSrUmXqYz0wFCrGcLOGwPODj
```

결과 예. 간략히 나타내기 위해 이 출력은 대부분 생략됩니다.

```
HTTP/1.1 200 OK
Content-Type: application/json
Content-Length: 204
X-Zfssa-Access-API: 1.0
```

```
{
  "service": {
    "methods": [
      {
        "path": "",

```

```

    "href": "/api/hardware/v1",
    "request": "GET",
    "description": "List the hardware service commands."
  },
  {
    "path": "/cluster",
    "href": "/api/hardware/v1/cluster",
    "request": "GET",
    "description": "Get cluster properties and cluster resource list"
  },
  {
    "path": "/cluster/resources/<resource:path>",
    "href": "/api/hardware/v1/cluster/resources/<resource:path>",
    "request": "GET",
    "description": "Get properties for the specified cluster resource"
  },
  {
    "path": "/cluster/resources/<resource:path>",
    "href": "/api/hardware/v1/cluster/resources/<resource:path>",
    "request": "PUT",
    "description": "Modify the specified cluster resource"
  },
  {
    "path": "/chassis",
    "href": "/api/hardware/v1/chassis",
    "request": "GET",
    "description": "List hardware chassis"
  }
],
"version": "1.1",
"name": "hardware",
"uri": "https://hostname:215/api/hardware/v1"
}
}

```

인증 토큰

POST 요청을 전송하여 액세스 서비스에서 비지속적 로그인 토큰을 얻습니다. 이 비지속적 로그인 토큰은 기타 모든 서비스가 ID 자격 증명으로 사용할 수 있습니다. 비지속적 로그인 토큰은 사용자의 세션 시간 초과 등록 정보에 의해 설정된 시간 초과 기간 이후 무효화됩니다. 기본값은 대개 15분입니다. DELETE 요청은 비지속적 로그인 토큰을 로그아웃한 후 무효화하는 데 사용될 수 있습니다.

이 비지속적 로그인 토큰은 이전 인증 세션 ID와 동등합니다. RESTful API 버전 2와 RESTful API 버전 1에서 모두 지원됩니다. 이는 ID가 만들어지고 클러스터 피어 간에 동기화되지 않는 클러스터 노드와 관련이 있습니다.

사용자는 RESTful API에 액세스하는 데 필요한 지속적 토큰을 만들 수도 있습니다. 지속적 토큰 만들기는 RESTful API 버전 2 이상에서만 지원됩니다. 지속적 토큰은 클러스터 피어 간에 동기화되므로 하나의 클러스터 노드에서 만들어져서 다른 클러스터 노드와 통신하는 데 사용될 수 있습니다. [RESTful API 사용자 서비스 \[235\]](#)를 참조하십시오.

비지속적 로그인 토큰 만들기

POST 요청은 새로운 비지속적 로그인 토큰을 요청합니다. 성공 시 HTTP 상태 201이 사용 가능한 RESTful API 서비스 목록을 포함하는 단일 등록 정보 `access`가 있는 JSON 객체와 함께 반환됩니다. 선택적 등록 정보인 `name`은 토큰의 이름을 설정하는 데 사용할 수 있습니다.

만들기 요청 예:

```
POST /api/access/v2 HTTP/1.1
Host: zfs-storage.example.com:215
X-Auth-User: root
X-Auth-Key: password-xxx
```

성공한 요청은 HTTP 상태 201(Created)과 함께 X-Auth-Session HTTP 헤더를 통해 비지속적 로그인 토큰을 반환합니다. 응답 본문은 이 로그인을 통해 액세스할 수 있는 서비스 목록을 포함합니다.

응답 헤더:

```
HTTP/1.1 201 Created
X-Auth-Session: puPnHZKgSrUmXqYz0wFCrGcLOGwPODj
X-Auth-Name: REST-YG02oRod
Content-Type: application/json
Content-Length: 378
X-Zfssa-Access-API: 1.0
```

```
{
  "access": {
    "services": [{
      ...
    }]
  }
}
```

비지속적 로그인 토큰 로그아웃 및 삭제

비어 있는 DELETE는 비지속적 로그인 토큰 로그아웃 및 무효화 요청을 전송합니다.

로그아웃 요청 예:

```
DELETE /api/access/v2 HTTP/1.1
X-Auth-Session: puPnHZKgSrUmXqYz0wFCrGcLOGwPODj
```

결과 예:

```
HTTP/1.1 204 No Content
X-Zfssa-Access-API: 1.0
```

인증서 관리

RESTful API를 사용하면 CSR(인증서 서명 요청), 시스템 또는 신뢰할 수 있는 사용자 인증서, CA(인증 기관) 인증서를 관리할 수 있습니다.

다음 표에서 *request*는 CSR의 *uuid*이고 *certificate*는 시스템 또는 신뢰할 수 있는 사용자 인증서 또는 CA 인증서의 *uuid*입니다.

표 6 인증서 작업

요청	경로 <i>/api/setting/v{12}</i> 에 추가	설명
GET	<i>/certificates/system/template</i>	템플릿 CSR을 반환합니다.
POST	<i>/certificates/system</i>	CSR을 만듭니다. 새 시스템 인증서를 업로드합니다.
GET	<i>/certificates/system/request</i>	지정된 CSR의 등록 정보를 나열합니다. PEM 형식으로 CSR을 반환합니다.
GET	<i>/certificates/system</i>	모든 시스템 인증서 및 요청의 등록 정보를 나열합니다.
PUT	<i>/certificates/system</i>	기본 시스템 인증서의 값을 설정합니다.
GET	<i>/certificates/system/certificate</i>	지정된 시스템 인증서의 등록 정보를 나열합니다. PEM 형식으로 인증서를 반환합니다.
DELETE	<i>/certificates/system/certificate</i>	지정된 시스템 인증서를 삭제합니다.
GET	<i>/certificates/trusted</i>	모든 신뢰할 수 있는 인증서의 등록 정보를 나열합니다.
POST	<i>/certificates/trusted</i>	새 신뢰할 수 있는 인증서를 업로드합니다.
GET	<i>/certificates/trusted/certificate</i>	지정된 신뢰할 수 있는 인증서의 등록 정보를 나열합니다. PEM 형식으로 인증서를 반환합니다.
PUT	<i>/certificates/trusted/certificate</i>	지정된 신뢰할 수 있는 인증서의 <i>services</i> 등록 정보 값을 설정합니다.
DELETE	<i>/certificates/trusted/certificate</i>	지정된 신뢰할 수 있는 인증서를 삭제합니다.

인증서 나열

다음 요청은 호스트에 있는 모든 시스템 인증서의 등록 정보와 default 등록 정보의 값을 나열합니다.

요청 예:

```
GET /api/setting/v2/certificates/system HTTP/1.1
Host: alice.example.com:215
Authorization: Basic Tm8gcGVla2luZyE=
Accept: application/json
```

결과 예:

다음 예제에서 첫번째 인증서는 자동 생성된 기존 인증서입니다(도메인 또는 IP 주소 기반). 두 번째 인증서는 ASN(어플라이언스 일련 번호) UUID를 기반으로 하는 자동 생성된 인증서입니다.

이 결과의 끝에서 default 등록 정보의 값은 시스템 기본 인증서가 자동으로 선택된 것을 보여줍니다.

```
HTTP/1.1 200 OK
Date: Sat, 08 May 2021 00:37:21 GMT
Content-Type: application/json; charset=utf-8
X-Zfssa-API-Version: 2.0
X-Zfssa-Setting-API: 2.0
Content-Length: 1975

{
  "certificates": [
    {
      "uuid": "system-cert1-uuid",
      "type": "cert",
      "data": {
        "subject": [
          {
            "countryName": "US"
          },
          {
            "stateOrProvinceName": "CA"
          },
          {
            "localityName": "Exampleton"
          },
          {
            "organizationName": "Example Corp, Inc"
          },
          {
            "commonName": "alice.example.com"
          }
        ],
        "issuer": [
          {
            "countryName": "US"
          },
          {
            "stateOrProvinceName": "AK"
          },
          {
            "localityName": "Trustville"
          }
        ]
      }
    }
  ]
}
```

```

    },
    {
      "organizationName": "Totally Trustworthy Certificates, Inc"
    },
    {
      "commonName": "Most Trusted Certificate"
    }
  ],
  "serialNumber": "64",
  "validity": {
    "notBefore": "20210520T21:08:27",
    "notAfter": "20220520T21:08:27"
  },
  "extensions": {
    "basicConstraints": {
      "value": [
        {
          "CA": false
        }
      ]
    },
    "subjectKeyIdentifier": {
      "value": "subjectKeyIdentifier"
    },
    "authorityKeyIdentifier": {
      "value": [
        {
          "keyid": "authorityKeyIdentifier"
        }
      ]
    },
    "subjectAltName": {
      "value": [
        {
          "DNS": "alice.example.com"
        },
        {
          "IP": "alice.example.com-ipaddr"
        }
      ]
    }
  }
},
"sha1fingerprint": "sha1fingerprint",
"href": "/api/setting/v2/certificates/system/system-cert1-uuid"
},
{
  "uuid": "system-cert2-uuid",
  "type": "cert",
  "asn": "8bf7f9bc-8b3a-4064-e59f-bf09e3dba275",
  "data": {
    "subject": [
      {
        "commonName": "8bf7f9bc-8b3a-4064-e59f-bf09e3dba275"
      }
    ],
    "issuer": [
      {
        "commonName": "8bf7f9bc-8b3a-4064-e59f-bf09e3dba275"
      }
    ],
    "serialNumber": "59:8A:73:7B:00:00:00:07",
    "validity": {
      "notBefore": "20060215T18:00:00",

```

```

        "notAfter": "20380119T03:14:07"
      },
      "extensions": {
        "nsComment": {
          "value": "Automatically generated"
        },
        "subjectAltName": {
          "critical": true,
          "value": [
            {
              "DirName": "8bf7f9bc-8b3a-4064-e59f-bf09e3dba275"
            }
          ]
        }
      }
    },
    "sha1fingerprint": "sha1fingerprint",
    "href": "/api/setting/v2/certificates/system/system-cert2-uuid"
  }
},
"default": "auto"
}

```

다음 요청은 지정된 *trusted-cert1-uuid* 신뢰할 수 있는 인증서의 등록 정보를 나열합니다.

요청 예:

```

GET /api/setting/v2/certificates/trusted/trusted-cert1-uuid HTTP/1.1
Host: alice.example.com:215
Authorization: Basic Tm8gcGVla2luZyE=
Accept: application/json

```

결과 예:

```

HTTP/1.1 200 OK
Date: Sat, 08 May 2021 00:37:57 GMT
Content-Length: 984
Content-Type: application/json; charset=utf-8
X-Zfssa-Setting-API: 2.0
X-Zfssa-API-Version: 2.0

```

```

{
  "certificate": {
    "uuid": "trusted-cert1-uuid",
    "type": "cert_ca",
    "data": {
      "subject": [
        {
          "countryName": "US"
        },
        {
          "stateOrProvinceName": "AK"
        },
        {
          "localityName": "Trustville"
        },
        {
          "organizationName": "Totally Trustworthy Certificates, Inc"
        },
        {
          "commonName": "Most Trusted Certificate"
        }
      ],
      "issuer": [

```

```

        {
          "countryName": "US"
        },
        {
          "stateOrProvinceName": "AK"
        },
        {
          "localityName": "Trustville"
        },
        {
          "organizationName": "Totally Trustworthy Certificates, Inc"
        },
        {
          "commonName": "Most Trusted Certificate"
        }
      ],
      "serialNumber": "83:F7:79:02:5F:44:4D:60",
      "validity": {
        "notBefore": "20210316T17:28:37",
        "notAfter": "20210415T17:28:37"
      },
      "extensions": {
        "subjectKeyIdentifier": {
          "value": "subjectKeyIdentifier"
        },
        "authorityKeyIdentifier": {
          "value": [
            {
              "keyid": "authorityKeyIdentifier"
            }
          ]
        },
        "basicConstraints": {
          "value": [
            {
              "CA": true
            }
          ]
        }
      }
    }
  },
  "sha1fingerprint": "sha1fingerprint",
  "services": [
    "ldap",
    "cloud"
  ],
  "href": "/api/setting/v2/certificates/trusted/trusted-cert1-uuid"
}
}

```

PEM 형식으로 인증서를 반환합니다.

PEM 형식으로 인증서를 반환하려면 Accept 헤더에 다음 값 중 하나를 지정합니다.

```

application/pkix-cert
application/x-x509-ca-cert
application/x-x509-user-cert
application/x-pem-file

```

요청 예:

```
GET /api/setting/v2/certificates/system/system-cert1-uuid HTTP/1.1
Host: alice.example.com:215
Authorization: Basic Tm8gcGVla2luZyE=
Accept: application/x-pem-file
```

결과 예:

```
HTTP/1.1 200 OK
Date: Thu, 13 May 2021 06:29:33 GMT
Content-Length: 1440
Content-Type: application/x-pem-file; charset=utf-8
X-Zfssa-Setting-API: 2.0
X-Zfssa-API-Version: 2.0

-----BEGIN CERTIFICATE-----
MIID+TCCAuGgAwIBAgIIXKtieQAAAAIwDQYJKoZIhvcNAQELBQAwVjEgMB4GA1UE
...
sUSSZgilvMJ4G8jtx6JSbG4DzDkvo8vq7GSika7h+hi5cbDiZds0L9kDtBIqSAVN
Z1gjaFpzgio6wRvaIA==
-----END CERTIFICATE-----
```

서버 인증서 만들기

서버 인증서를 만드는 첫번째 단계는 CSR(인증서 서명 요청)을 만드는 것입니다. 애플리케이션에 CSR을 게시하고 이를 CA에 전송합니다. CA에서 서명된 인증서를 받은 후 [“키 또는 인증서 업로드” \[41\]](#)에 설명된 대로 서명된 인증서를 업로드합니다. 서명된 인증서가 요청을 바꿉니다.

요청 템플릿 반환

template 명령은 필요한 최소 등록 정보의 기본값을 포함하여 CSR의 프레임워크를 반환합니다.

요청 예:

```
GET /api/setting/v2/certificates/system/template HTTP/1.1
Host: alice.example.com:215
Authorization: Basic Tm8gcGVla2luZyE=
Accept: application/json
```

결과 예:

```
HTTP/1.1 200 OK
Date: Thu, 13 May 2021 08:03:03 GMT
Content-Length: 261
Content-Type: application/json; charset=utf-8
X-Zfssa-Setting-API: 2.0
X-Zfssa-API-Version: 2.0

{
  "request": {
    "type": "request",
```

```

    "data": {
      "subject": [
        {
          "commonName": "alice.example.com"
        }
      ],
      "extensions": {
        "subjectAltName": {
          "value": [
            {
              "IP": "alice.example.com-ipaddr"
            },
            {
              "DNS": "alice.example.com"
            }
          ]
        }
      }
    },
    "href": "/api/setting/v2/certificates/system/template"
  }
}

```

요청 채우기 및 업로드

이 `template` 출력을 사용하는 경우 `data` 요소만 포함합니다.

CSR에 지정하려는 추가 등록 정보의 경우 “인증서 나열” [34]에 표시된 것처럼 기존 시스템 인증서의 등록 정보를 나열합니다.

CSR에 만족하면 다음 예제에 표시된 것처럼 호스트에 CSR을 업로드합니다. CSR을 업로드한 후에는 더 이상 이를 변경할 수 없습니다.

요청 예:

```

POST /api/setting/v2/certificates/system HTTP/1.1
Host: alice.example.com:215
Authorization: Basic Tm8gcGVla2luZyE=
Content-type: application/json

```

```

{
  "data": {
    "subject": [
      {
        "commonName": "alice.example.com"
      },
      {
        "organizationName": "Example Corp, Inc"
      },
      {
        "localityName": "Exampleton"
      },
      {
        "stateOrProvinceName": "CA"
      },
      {
        "countryName": "US"
      }
    ]
  }
}

```

```

    ],
    "extensions": {
      "subjectAltName": {
        "value": [
          {
            "DNS": "alice.example.com"
          },
          {
            "IP": "alice.example.com-ipaddr"
          }
        ]
      }
    }
  }
}

```

결과 예:

```

HTTP/1.1 201 Created
Date: Fri, 14 May 2021 01:17:45 GMT
Content-Type: application/json; charset=utf-8
X-Zfssa-API-Version: 2.0
X-Zfssa-Setting-API: 2.0
Location: /api/setting/v2/certificates/system/65119889-98d3-4fc4-bff5-f007a55f6cb3
Content-Length: 379

```

```

{
  "request": {
    "uuid": "csr-uuid",
    "type": "request",
    "data": {
      "subject": [
        {
          "commonName": "alice.example.com"
        },
        {
          "organizationName": "Example Corp, Inc"
        },
        {
          "localityName": "Exempleton"
        },
        {
          "stateOrProvinceName": "CA"
        },
        {
          "countryName": "US"
        }
      ],
      "extensions": {
        "subjectAltName": {
          "value": [
            {
              "DNS": "alice.example.com"
            },
            {
              "IP": "alice.example.com-ipaddr"
            }
          ]
        }
      }
    }
  },
  "href": "/api/setting/v2/certificates/system/csr-uuid"
}

```

CA에 요청 전송

업로드한 CSR에는 PEM 형식으로 요청을 검색하거나 등록 정보를 표시하기 위해 사용할 수 있는 UUID가 포함됩니다.

PEM 형식으로 CSR을 반환하려면 Accept 헤더에 다음 값 중 하나를 지정합니다.

```
application/pkcs10
application/x-pem-file
```

요청 예:

```
GET /api/setting/v2/certificates/system/csr-uuid HTTP/1.1
Host: alice.example.com:215
Authorization: Basic Tm8gcGVla2luZyE=
Accept: application/x-pem-file
```

결과 예:

```
HTTP/1.1 200 OK
Date: Fri, 14 May 2021 03:47:21 GMT
Content-Type: application/x-pem-file; charset=utf-8
X-Zfssa-API-Version: 2.0
X-Zfssa-Setting-API: 2.0
Content-Length: 997

-----BEGIN CERTIFICATE REQUEST-----
MIICpjCCAY4CAQAwJDEiMCAgA1UEAwwZYXJkb2NoLWt6LTludWsub3JhY2x1LmNv
...
Bc0Q9FVRVv89AkmeAlF7727aIqmgmFcIUIIrEKTG4PSacdaoBsbjprizCuMhyo
vgUKOPE/0xLafw==
-----END CERTIFICATE REQUEST-----
```

정해진 방식으로 CA에 CSR을 전송합니다. CA에서 서명된 인증서가 수신되면 [“키 또는 인증서 업로드” \[41\]](#)에 표시된 대로 서명된 인증서를 업로드합니다.

키 또는 인증서 업로드

CA에서 서명된 인증서가 수신되면 다음 명령을 사용하여 인증서를 업로드합니다. Content-Type 헤더에 다음 값 중 하나를 지정합니다.

```
application/pkix-cert
application/x-x509-ca-cert
application/x-x509-user-cert
application/x-pem-file
```

요청 예:

```
POST /api/setting/v2/certificates/system HTTP/1.1
Host: alice.example.com:215
Authorization: Basic Tm8gcGVla2luZyE=
Content-Type: application/x-pem-file
```

```
-----BEGIN CERTIFICATE-----
MIIDdTCCA12gAwIBAgICH5cwDQYJKoZIhvcNAQELBQAwazELMAkGA1UEBhMCdXMx
...
cgfvvd1NUEvSdlb2+cjRbd9uSdtLfv7H5BKtKEdOXikv9+f150MytMEo4ABt0pEyp
/KwtRsdIxMzAjmNqfQPR6eAHVfQ/CGwh6Q==
-----END CERTIFICATE-----
```

결과 예:

```
HTTP/1.1 200 OK
Date: Tue, 11 May 2021 18:04:22 GMT
Content-Type: application/json; charset=utf-8
X-Zfssa-API-Version: 2.0
X-Zfssa-Setting-API: 2.0
Location: /api/setting/v2/certificates/system/system-cert3-uuid
Content-Length: 987
```

```
{
  "certificate": {
    "uuid": "system-cert3-uuid",
    "type": "cert",
    "data": {
      "subject": [
        {
          "countryName": "US"
        },
        {
          "stateOrProvinceName": "CA"
        },
        {
          "localityName": "Exampleton"
        },
        {
          "organizationName": "Example Corp, Inc"
        },
        {
          "commonName": "alice.example.com"
        }
      ],
      "issuer": [
        {
          "countryName": "US"
        },
        {
          "stateOrProvinceName": "AK"
        },
        {
          "localityName": "Trustville"
        },
        {
          "organizationName": "Totally Trustworthy Certificates, Inc"
        },
        {
          "commonName": "Most Trusted Certificate"
        }
      ],
      "serialNumber": "64",
      "validity": {
        "notBefore": "20210520T21:08:27",
        "notAfter": "20220520T21:08:27"
      },
      "extensions": {
        "basicConstraints": {
          "value": [
```

```

        {
          "CA": false
        }
      ],
      "subjectKeyIdentifier": {
        "value": "subjectKeyIdentifier"
      },
      "authorityKeyIdentifier": {
        "value": [
          {
            "keyid": "authorityKeyIdentifier"
          }
        ]
      },
      "subjectAltName": {
        "value": [
          {
            "DNS": "alice.example.com"
          },
          {
            "IP": "alice.example.com-ipaddr"
          }
        ]
      }
    }
  },
  "sha1fingerprint": "sha1fingerprint",
  "href": "/api/setting/v2/certificates/system/system-cert3-uuid"
}

```

인증서를 신뢰해야 하는 서비스 지정

시스템 인증서 또는 CSR의 등록 정보는 수정할 수 없습니다. CSR을 게시하기 전 CSR의 등록 정보를 설정합니다.

신뢰할 수 있는 인증서의 `services` 등록 정보의 값을 설정할 수 있습니다. `services` 등록 정보는 신뢰해야 하는 인증서에 대한 서비스 목록입니다.

다음 예제는 신뢰할 수 있는 인증서의 `services` 등록 정보를 설정합니다.

```

PUT /api/setting/v2/certificates/trusted/trusted-cert2-uuid HTTP/1.1
Authorization: Basic Tm8gcGVla2luZyE=
Host: alice.example.com:215
Content-Type: application/json

```

```
{"certificate": { "services": ["ldap"] }}
```

다음 예제는 인증서를 신뢰해야 하는 여러 서비스를 설정합니다.

```

PUT /api/setting/v2/certificates/trusted/trusted-cert2-uuid HTTP/1.1
Authorization: Basic Tm8gcGVla2luZyE=
Host: alice.example.com:215
Content-Type: application/json

```

```
{"certificate": {"services": [ "ldap", "cloud" ] }}
```

시스템 기본 인증서 설정

다음 예에서는 기본 시스템 인증서를 지정합니다.

```
PUT /api/setting/v2/certificates/system HTTP/1.1
Authorization: Basic Tm8gcGVla2luZyE=
Host: alice.example.com:215
Accept: application/json
Content-Type: application/json

{ "default": "system-cert1-uuid" }
```

인증서 삭제

DELETE 명령은 지정된 인증서, 요청 또는 키를 삭제합니다. 성공하면 HTTP 상태 204(No Content)가 반환됩니다.

```
DELETE /api/setting/v2/certificates/system/system-cert2-uuid HTTP/1.1
```

HTTP Strict Transport Security 사용

HSTS(HTTP Strict Transport Security)는 지정된 기간 동안 HTTP 연결이 아닌 보안 HTTPS 연결만 허용합니다. HSTS를 사용하기 전에 HSTS 필수 조건을 파악하고 HSTS가 사용으로 설정된 브라우저 동작을 이해하고 인증 기관에서 서명한 인증서를 설치하십시오.

주 - 인증서를 유효하고 적절한 상태로 유지하지 못할 경우 HSTS 보안의 이점이 무효화되거나 브라우저가 어플라이언스에 연결하지 못할 수 있습니다.

다음 예에 표시된 것처럼 HSTS가 사용으로 설정된 상태로 유지되는 기본 최대 시간은 63072000초(2년)입니다.

요청 예:

```
GET /api/setting/v2/security HTTP/1.1
Host: alice.example.com:215
Authorization: Basic Tm8gcGVla2luZyE=
Accept: application/json
```

결과 예:

```
HTTP/1.1 200 OK
Date: Fri, 14 May 2021 19:22:27 GMT
Content-Type: application/json; charset=utf-8
X-Zfssa-Api-Version: 2.0
X-Zfssa-Setting-Api: 2.0
Content-Length: 109
```

```
{
  "Security settings": {
    "href": "/api/setting/v2/security",
    "hsts_enable": false,
    "hsts_max_age": 63072000
  }
}
```

이 어플라이언스에 대해 HSTS를 사용으로 설정하려면 `hsts_enable` 등록 정보를 `true`로 설정합니다.

```
PUT /api/setting/v2/security HTTP/1.1
Host: alice.example.com:215
Content-Type: application/json

{"hsts_enable": true}
```


RESTful API 경고 서비스

하드웨어 및 소프트웨어 결함과 같은 중요한 어플라이언스 이벤트가 경보를 트리거합니다. 경보는 로그에 나타나며 기타 경고 조치(예: 전자메일 전송 또는 데이터 세트 재개)를 수행하도록 구성할 수도 있습니다.

RESTful API 경고 서비스를 통해 사용자정의 경고 조치(이벤트 경고에 대한 응답) 및 사용자정의 Analytics 통계 임계값 경보를 만들 수 있습니다.

경보 서비스 명령

다음 표는 경고 서비스 명령을 보여 줍니다. 단일 경고 이벤트에 대한 경고 조치(응답)를 2개 이상 정의할 수 있습니다. 예를 들어, 2개의 다른 그룹으로 전자메일을 전송하고, syslog 메시지를 작성하고, 단일 경고에 대한 응답으로 데이터 세트를 모두 재개할 수 있습니다. *actions-###* 객체는 경고 이벤트 및 경고에 대한 모든 응답의 세트입니다. *action-###* 객체는 응답 중 하나입니다.

표 7 경고 서비스 명령

요청	경로 <i>/api/alert/v{1 2}</i> 에 추가	설명
GET	<i>/api/alert/v{1 2}</i> 만 사용	경보 서비스 명령을 나열합니다.
GET	<i>/actions</i>	모든 경고 조치 객체를 나열합니다.
POST	<i>/actions</i>	새 경고 조치를 만듭니다.
GET	<i>/actions/actions-###</i>	지정된 경고 조치 등록 정보를 나열합니다.
PUT	<i>/actions/actions-###</i>	지정된 경고 조치 객체를 수정합니다.
DELETE	<i>/actions/actions-###</i>	지정된 조치 객체를 삭제합니다.
POST	<i>/actions/actions-###</i>	새 경고 조치 작업을 만듭니다.
GET	<i>/actions/actions-###/action-###</i>	지정된 경고 조치 작업 등록 정보를 나열합니다.
PUT	<i>/actions/actions-###/action-###</i>	지정된 경고 조치 작업 객체를 수정합니다.
POST	<i>/postalert</i>	사용자정의 경보를 게시합니다.
DELETE	<i>/actions/actions-###/action-###</i>	지정된 경고 조치 작업 객체를 삭제합니다.
GET	<i>/thresholds</i>	모든 임계값 경고 객체를 나열합니다.
POST	<i>/thresholds</i>	새 임계값 경보를 만듭니다.
GET	<i>/thresholds/threshold-alert-uuid</i>	지정된 임계값 경고 등록 정보를 나열합니다.

요청	경로 <code>/api/alert/v{1 2}</code> 에 추가	설명
PUT	<code>/thresholds/threshold-alert-uuid</code>	지정된 임계값 경보 객체를 수정합니다.
DELETE	<code>/thresholds/threshold-alert-uuid</code>	지정된 임계값 경보 객체를 삭제합니다.
GET	<code>/events</code>	새 경보 이벤트를 수신합니다.

경보 조치

경보 조치는 이벤트 경보에 대한 응답입니다. 경보 조치를 만들려면 하나 이상의 이벤트를 지정하고, 해당 이벤트(예: 전자메일 전송 또는 워크플로우 실행)에 대한 경보가 전송될 때 수행할 조치를 하나 이상 지정합니다. 특정 이벤트 경보에 대해 둘 이상의 경보 조치를 지정할 수 있습니다.

`category` 등록 정보는 경보 조치가 수행될 이벤트의 범주를 지정합니다. 각 범주에는 하나 이상의 이벤트가 포함됩니다. 경보 조치를 만들거나 나열하는 경우 각 범주 내 이벤트가 나열됩니다. 기본적으로 범주 내 모든 이벤트에 대해 경보 조치가 수행됩니다. 이벤트 하위 세트에 대해서만 경보 조치를 수행해야 할 경우 이벤트로 인해 경보 조치가 수행되지 않아야 함을 나타내도록 등록 정보 값을 `false`로 변경합니다.

다음 표에서는 지정 가능한 이벤트 범주에 대해 설명합니다.

표 8 경보 조치 이벤트 범주

범주	설명
<code>ad</code>	Active Directory 또는 SMB Kerberos 클라이언트 인증 성능이 저하되었습니다.
<code>all</code>	상위 레벨 이벤트(예: 모든 경보 또는 결함, 서비스 경보 및 하드웨어 결함)입니다.
<code>analytics</code>	상위 레벨 이벤트(예: 데이터 세트 자동 일시 중지 경고, 메모리 합계 초과 및 사용량 초과)입니다. 특정 Analytics 통계에 대한 이벤트는 "임계값 경보" [56] 를 참조하십시오.
<code>appliance_software</code>	소프트웨어 업데이트를 방해하거나 커널 패닉을 반환하는 이벤트입니다.
<code>cloud_snapshot</code>	
<code>cluster</code>	연결 장애와 피어 오류 등의 클러스터 이벤트.
<code>custom</code>	워크플로우에서 지정된 사용자정의 이벤트에 대한 경보 조치입니다. "사용자정의 경보" [54] 를 참조하십시오.
<code>hardware</code>	어플라이언스 부트 및 하드웨어 구성 변경.
<code>hardware_faults</code>	하드웨어 결함입니다.
<code>ndmp</code>	NDMP TAR/DUMP 백업 및 복원 시작 및 완료 이벤트
<code>backup</code>	
<code>restore</code>	
<code>network</code>	네트워크 포트, 데이터 링크, IP 인터페이스의 이벤트 및 장애.

범주	설명
scrk	번들 업로드 이벤트 지원.
replication	이벤트 및 오류 송수신.
replication_source	
replication_target	
smf	소프트웨어 서비스 오류 이벤트.
shadow	마이그레이션 오류 또는 마이그레이션 완료입니다.
thresholds	"임계값 경고" [56]에 설명된 대로 기존 임계값 이벤트 경보에 조치를 추가할 수 있습니다.
zfs_pool	스크러빙 및 핫 스페이스 활성화 등의 스토리지 풀 이벤트.

handler 등록 정보는 지정된 이벤트가 발생할 때 수행할 조치의 유형을 지정합니다. 다음 표에서와 같이 handler에 대한 값이 지정되면 대부분 이에 따라 추가 등록 정보를 설정해야 합니다.

표 9 경보 조치 응답 유형

응답 유형(handler)	처리기 등록 정보	응답 유형 설명
email	address	지정된 수신자에게 지정된 제목의 전자메일을 전송합니다.
	subject	수신자를 목록이 아닌 하나만 입력합니다. 개별 수신자 여럿에게 전송하려면 이 이벤트 경보에 대한 추가 경고 조치를 만듭니다. SMTP 서비스를 사용하여 전자메일 전송 방법을 구성합니다. " 서비스 나열 " [143]을 참조하십시오.
snmp_trap	없음	경보 세부정보를 포함하는 SNMP 트랩을 전송합니다. SNMP 서비스를 사용하여 SNMP 트랩 대상을 구성할 수 있습니다. " 서비스 나열 " [143]을 참조하십시오.
syslog	없음	경보 세부정보를 포함하는 시스템 메시지를 하나 이상의 원격 시스템으로 전송합니다. syslog 서비스를 사용하여 syslog 대상을 구성합니다. " 서비스 나열 " [143]을 참조하십시오. syslog 메시지 전송에 대한 자세한 내용은 " Syslog Configuration in Oracle ZFS Storage Appliance Administration Guide, Release OS8.8.x "을 참조하십시오.
resume_dataset	dataset	Analytics 데이터 세트를 재개합니다. 데이터 세트 재개 및 일시 중지는 일시적 성능 문제를 진단하고 데이터 세트를 계속 사용으로 설정하는 것이 적절하지 않은 경우 등에 유용할 수 있습니다. 자세한 내용은 " About Analytics Datasets in Oracle ZFS Storage Appliance Analytics Guide, Release OS8.8.x "를 참조하십시오.
suspend_dataset	dataset	Analytics 데이터 세트를 일시 중지합니다.
resume_worksheet	worksheet	Analytics 워크시트를 재개합니다. 워크시트 재개 및 일시 중지는 데이터 세트 재개 및 일시 중지와 동일한 이유로 유용할 수 있습니다. 워크시트에는 다양한 데이터 세트가 포함될 수 있습니다.

응답 유형(handler)	처리기 등록 정보	응답 유형 설명
		자세한 내용은 “Worksheet Graphs and Plots” in Oracle ZFS Storage Appliance Analytics Guide, Release OS8.8.x 을 참조하십시오.
suspend_worksheet	worksheet	Analytics 워크시트를 일시 중지합니다.
execute_workflow	workflow	지정된 워크플로우를 실행합니다. 경보 조치로 적격한 워크플로우에 대한 자세한 내용은 “Using Workflows for Alert Actions” in Oracle ZFS Storage Appliance Administration Guide, Release OS8.8.x 을 참조하십시오.

모든 경보 조치 나열

모든 경보 조치를 나열하는 경우 각 경보 조치에 대해 이벤트 범주 및 해당 범주 내 각 이벤트가 나열됩니다. 경보 조치에 대해 정의된 응답을 표시하려면 [“단일 경보 조치 나열” \[51\]](#)을 참조하십시오.

경보 조치를 가져오는 요청의 예:

```
GET /api/alert/v1/actions HTTP/1.1
Authorization: Basic Tm8gcGVla2luZyE=
Host: zfs-storage.example.com:215
Accept: application/json
```

결과 예:

```
HTTP/1.1 200 OK
X-Zfssa-Appliance-API: 1.0
Content-Type: application/json
Content-Length: 1395

{
  "actions": [
    {
      "category": "smf",
      "uuid": "actions-uuid1",
      "failed_services": true,
      "degraded_services": true,
      "repaired_services": false,
      "actions": "actions-000",
      "href": "/api/alert/v2/actions/actions-000"
    },
    {
      "category": "analytics",
      "uuid": "actions-uuid2",
      "analytics_datasets_auto-suspend_notify": false,
      "analytics_datasets_auto-suspend_warning": false,
      "analytics_memory_total_exceeded": true,
      "analytics_memory_total_normal": false,
      "analytics_usage_exceeded": true,
      "analytics_usage_normal": false,
      "actions": "actions-001",
      "href": "/api/alert/v2/actions/actions-001"
    }
  ]
}
```

단일 경보 조치 나열

나열할 특정 경보 조치를 지정하는 경우 해당 경보 조치에 대해 이벤트 범주와 해당 범주 내 각 이벤트, 각 응답 또는 조치가 나열됩니다.

다음 경보 조치의 응답 세 가지는 모두 `true` 이벤트 중 하나가 발생할 때 수행됩니다.

요청 예:

```
GET /api/alert/v1/actions/actions-000 HTTP/1.1
```

결과 예:

```
HTTP/1.1 200 OK
X-Zfssa-Appliance-API: 1.0
Content-Type: application/json
Content-Length: 331

{
  "actions": {
    "href": "/api/alert/v2/actions/actions-000",
    "category": "smf",
    "uuid": "actions-uuid1",
    "failed_services": true,
    "degraded_services": true,
    "repaired_services": false,
    "action-000": {
      "handler": "email",
      "address": "admin@example.com",
      "subject": "failed or degraded service",
      "href": "/api/alert/v2/actions/actions-000/action-000"
    },
    "action-001": {
      "handler": "email",
      "address": "it-team@example.com",
      "subject": "failed or degraded service",
      "href": "/api/alert/v2/actions/actions-000/action-001"
    },
    "action-002": {
      "handler": "syslog",
      "href": "/api/alert/v2/actions/actions-000/action-002"
    }
  }
}
```

경보 조치 만들기

경보 조치를 만들 때는 이 사용자정의 응답을 정의 중인 이벤트의 범주인 `category` 등록 정보에 대한 값을 지정해야 합니다. `category` 값의 목록은 표 8. “경보 조치 이벤트 범주”를 참조하십시오.

요청 예:

```
POST /api/alert/v1/actions HTTP/1.1
Host: zfs-storage.example.com:215
```

```
X-Auth-Session: uerqghq84vbdv
Content-Type: application/json
Content-Length: 30
```

```
{"category": "smf"}
```

결과 예:

결과는 지정된 이벤트 범주 내 모든 이벤트를 나열합니다. 기본적으로 범주 내 모든 이벤트로 인해 응답 조치가 수행됩니다(모두 `true`로 설정됨).

```
HTTP/1.1 201 Created
X-Zfssa-Appliance-API: 1.0
Content-Type: application/json
Content-Length: 118
Location: /api/alert/v1/actions/actions-000
```

```
{
  "actions": {
    "href": "/api/alert/v2/actions/actions-000",
    "category": "smf",
    "uuid": "actions-uuid",
    "failed_services": true,
    "degraded_services": true,
    "repaired_services": true
  }
}
```

경보 조치 수정

지정된 이벤트 범주 내 일부 이벤트로 인해 응답 조치가 수행되지 않아야 할 경우 해당 이벤트를 나타내는 등록 정보를 `false`로 설정합니다.

다음 예에서는 실패하거나 성능이 저하된 서비스와 복구된 서비스에 대해 다른 응답을 정의할 수 있습니다.

요청 예:

```
PUT /api/alert/v1/actions/actions-000 HTTP/1.1
Host: zfs-storage.example.com:215
X-Auth-Session: uerqghq84vbdv
Content-Type: application/json
Content-Length: 30
```

```
{"repaired_services": false}
```

결과 예:

```
HTTP/1.1 202 Accepted
X-Zfssa-Appliance-API: 1.0
Content-Type: application/json
Content-Length: 195
```

```
{
  "actions": {
    "href": "/api/alert/v2/actions/actions-000",
    "category": "smf",
```

```

    "uuid": "actions-uuid",
    "failed_services": true,
    "degraded_services": true,
    "repaired_services": false
  }
}

```

이벤트에 대한 응답 지정

기본적으로 이벤트 경보는 경보 로그에 기록됩니다. 앞선 예에서는 경보 로그에 게시하는 것 이외에도 응답을 정의할 이벤트를 지정했습니다. 해당 이벤트에 대한 응답을 정의하려면 특정 경보 조치의 handler 등록 정보 값을 지정합니다. handler 값의 목록은 [표 9. "경보 조치 응답 유형"](#)을 참조하십시오.

요청 예:

이 예에서는 actions-000 경보에 대해 admin으로 전자메일을 전송하는 경보 조치를 만듭니다.

```

POST /api/alert/v1/actions/actions-000 HTTP/1.1
Host: zfs-storage.example.com:215
X-Auth-Session: uerqghq84vbdv
Content-Type: application/json
Content-Length: 68

```

```

{"handler": "email", "address": "admin@example.com", "subject": "failed or degraded service"}

```

결과 예:

```

HTTP/1.1 201 Created
X-Zfssa-Appliance-API: 1.0
Content-Type: application/json
Content-Length: 177
Location: /api/alert/v1/actions/actions-000/action-000

```

```

{
  "action": {
    "href": "/api/alert/v1/actions/actions-002/action-000",
    "handler": "email",
    "address": "admin@example.com",
    "subject": "failed or degraded service"
  }
}

```

동일한 이벤트에 대한 추가 응답을 지정하려면 동일한 경보 조치에 대해 POST 요청을 다시 실행하고 다른 처리기를 지정하거나, 동일한 처리기를 지정하고 처리기에 대해 다른 인수를 지정합니다. 한 요청에서 handler 등록 정보를 2개 이상 지정하면 마지막 등록 정보를 제외한 모든 handler 등록 정보가 무시됩니다.

다음 예가 짧게 줄어듭니다. 해당 요청은 ["단일 경보 조치 나열" \[51\]](#)에서와 같이 /api/alert/v1/actions/actions-000/action-001 및 /api/alert/v1/actions/actions-000/action-002를 만듭니다.

```

POST /api/alert/v1/actions/actions-002 HTTP/1.1
...

```

```
{"handler": "email", "address": "it-team@example.com", "subject": "failed or degraded service"}

POST /api/alert/v1/actions/actions-002 HTTP/1.1
...
{"handler": "syslog"}
```

이벤트에 대한 응답 수정

응답을 수정하려면 수정할 응답에 대한 HREF를 지정합니다.

```
PUT /api/alert/v1/actions/actions-000/action-001 HTTP/1.1
Host: zfs-storage.example.com:215
X-Auth-Session: uerqghq84vbdv
Content-Type: application/json
Content-Length: 28

{"address": "it-group@example.com"}
```

이벤트에 대한 응답 삭제

응답을 삭제하려면 삭제할 응답에 대한 HREF를 지정합니다.

```
DELETE /api/alert/v1/actions/actions-000/action-000 HTTP/1.1
Host: zfs-storage.example.com:215
X-Auth-Session: uerqghq84vbdv

HTTP/1.1 204 No Content
```

경보 조치 삭제

경보 조치를 삭제하려면 삭제할 경보 조치에 대한 HREF를 지정합니다.

요청 예:

```
DELETE /api/alert/v1/actions/actions-003 HTTP/1.1
Authorization: Basic Tm8gcGVla2luZyE=
Host: zfs-storage.example.com:215
```

결과 예:

```
HTTP/1.1 204 No Content
X-Zfssa-Appliance-API: 1.0
```

사용자정의 경보

사용자정의 경보는 사용자정의 이벤트에 대한 경보 조치입니다. category의 값에 대해 custom을 지정합니다.

사용자정의 이벤트는 워크플로우에서 정의됩니다. 미리 정의된 이벤트 중에서 선택하는 대신, 워크플로우에서 정의된 이벤트를 기술하는 다음 등록 정보를 지정합니다.

표 10 사용자정의 경보 등록 정보

등록 정보	유형	설명
severity	String	선택사항입니다. 경보가 발생하게 만든 이벤트의 심각도입니다. 유효한 값은 Minor, Major 또는 Critical입니다.
description	String	필수입니다. 경보를 발생시킨 이벤트에 대한 설명입니다.
response	String	선택사항입니다. 시스템이 이 이벤트의 영향을 줄이기 위해 수행할 조치에 대한 설명입니다.
impact	String	선택사항입니다. 이 이벤트가 어플라이언스에 끼치는 영향에 대한 설명입니다.
recommended_action	String	선택사항입니다. 관리자가 이 이벤트의 영향을 줄이기 위해 수행해야 할 조치에 대한 설명입니다.

사용자정의 경보 생성 및 게시에 필요한 권한 부여 구성

사용자정의 경보를 만들려면 사용자에게 alert 범위의 allow_configure 권한 부여가 있어야 합니다. 사용자정의 경보를 게시하려면 사용자에게 alert 범위의 allow_post 권한 부여가 있어야 합니다. [“Authorizations Required to Use Custom Alerts” in Oracle ZFS Storage Appliance Administration Guide, Release OS8.8.x](#)를 참조하십시오.

사용자정의 경보 만들기

category의 값에 대해 custom을 지정하고, 표 10. “사용자정의 경보 등록 정보”에서 최소한 description 등록 정보에 대해 값을 설정합니다.

요청 예:

```
POST /api/alert/v1/actions
Accept: application/json
Content-Type: application/json
```

```
{"category": "custom", "severity": "Minor", "description": "Custom alert description",
"response": "What the system will do", "impact": "What happened to the appliance",
"recommended_action": "What the administrator should do"}
```

결과 예:

출력에서 경보의 uuid를 기록해 둡니다. 이 정보는 경보 게시에 필요합니다.

```
{
  "actions": {
    "href": "/api/alert/v1/actions/actions-004",
    "category": "custom",
    "severity": "Minor",
    "description": "Custom alert description",
```

```

        "response": "What the system will do",
        "impact": "What happened to the appliance",
        "recommended_action": "What the administrator should do",
        "uuid": "custom-alert-uuid"
    }
}

```

“[경보 조치 수정](#)” [52]에 설명된 대로 다른 경보를 수정하는 것과 동일한 방식으로 이 사용자정의 경보를 수정합니다. severity, description, response, impact 또는 recommended_action 등록 정보의 값을 추가 또는 변경하면 됩니다.

“[이벤트에 대한 응답 지정](#)” [53] 및 “[이벤트에 대한 응답 수정](#)” [54]에 설명된 대로 다른 경보에 대한 응답을 지정하는 것과 동일한 방식으로 이 사용자정의 경보에 대한 응답을 지정합니다.

“[이벤트에 대한 응답 삭제](#)” [54] 및 “[경보 조치 삭제](#)” [54]에 설명된 대로 다른 경보에 대한 것과 동일한 방식으로 경보 응답 또는 사용자정의 경보를 삭제합니다.

사용자정의 경보 게시

게시할 사용자정의 경보의 UUID를 제공해야 합니다. UUID는 경보를 만들 때와 경보를 나열할 때 표시됩니다. “[사용자정의 경보 만들기](#)” [55] 및 “[단일 경보 조치 나열](#)” [51]을 참조하십시오.

category 값이 custom인 경보만 게시할 수 있습니다.

경보의 UUID와 함께 경보에서 지정되지 않은 등록 정보(표 10. “[사용자정의 경보 등록 정보](#)”에 나열됨)를 지정해야 합니다. 경보가 게시될 때 경보에서 지정된 등록 정보에 새 값이 제공될 수 있습니다.

요청 예:

```

POST /api/alert/v1/postalert
Accept: application/json
Content-Type: application/json

```

```

{"uuid": "custom-alert-uuid"}

```

결과 예:

```

{
  "uuid": "posted-alert-uuid"
}

```

임계값 경고

임계값 경고는 특정 Analytics 통계에 대해 임계값이 정의된 사용자정의 경보이며, 통계 값이 해당 임계값을 벗어나는 경우 경보 조치가 실행됩니다. [Oracle ZFS Storage Appliance Analytics Guide, Release OS8.8.x](#)를 참조하십시오.

다음 표에서는 Analytics 통계 지정, 해당 통계에 대한 임계값 정의, 해당 임계값 경보에 대해 경보 조치가 실행될 시기 정의를 위해 설정할 등록 정보를 설명합니다.

표 11 임계값 경고 등록 정보

등록 정보	유형	설명
statname	AnalyticsStatistics	필수입니다. 모니터할 Analytics 통계입니다.
limit	PositiveInteger	필수입니다. 경보를 트리거하는 임계값입니다. 초당 바이트, 작업, 액세스 또는 요청의 퍼센트나 개수입니다.
type	ChooseOne	임계값(limit)과 현재 통계 값을 비교하는 방법입니다. <ul style="list-style-type: none"> ■ normal – 현재 통계 값이 임계값을 초과합니다. 이는 기본값입니다. ■ inverted – 현재 통계 값이 임계값 미만입니다.
minpost	Duration	경보 조치가 실행되기 전 통계 값이 임계값 조건으로 유지되어야 할 기간(초)입니다. 기본값은 5분입니다.
days	ChooseOne	해당 경보를 전송할 일 수(all days, weekdays 또는 weekends)입니다. 기본값은 all입니다.
window_start	TimeOfDay	이 경보 조치를 실행할 시간 범위입니다. 30분 증분 간격으로 00:00부터 23:30 UTC까지의 시간을 지정합니다. 조건이 충족될 때 이 경보 조치를 실행하려면 시작 시간 또는 종료 시간으로 none을 지정합니다. 기본값은 none(window_start의 경우) 및 00:00(window_end의 경우)입니다.
window_end		
frequency	Duration	통계 값이 임계값 조건으로 유지되는 동안 경보 조치 재실행 사이의 기간(초)입니다. 기본값은 5분입니다.
minclear	Duration	후속 경보 조치가 실행되기 전 통계 값이 임계값 조건에서 벗어나야 할 기간(초)입니다. 기본값은 5분입니다.

임계값 경고 나열

다음 요청은 구성된 임계값 경보를 모두 나열합니다.

요청 예:

```
GET /api/alert/v1/thresholds HTTP/1.1
Authorization: Basic Tm8gcGV1a2luZyE=
Host: zfs-storage.example.com:215
Accept: application/json
```

결과 예:

```
HTTP/1.1 200 OK
Date: Tue, 15 Oct 2019 10:38:40 GMT
X-Zfssa-Appliance-API: 1.0
Content-Type: application/json
Content-Length: 689
```

```
{
  "thresholds": [
    {
      "uuid": "threshold-uuid1",
```

```

        "statname": "cpu.utilization",
        "type": "normal",
        "limit": 80,
        "minpost": 300,
        "days": "weekdays",
        "window_start": "08:00",
        "window_end": "19:30",
        "frequency": 300,
        "minclear": 300,
        "threshold": "threshold-000",
        "href": "/api/alert/v1/thresholds/threshold-uuid1"
    },
    {
        "uuid": "threshold-uuid2",
        "statname": "cap.meta_percentused[pool]",
        "type": "normal",
        "limit": 85,
        "minpost": 300,
        "days": "all",
        "window_start": "none",
        "window_end": "00:00",
        "frequency": 0,
        "minclear": 0,
        "threshold": "threshold-001",
        "href": "/api/alert/v1/thresholds/threshold-uuid2"
    }
]
}

```

다음 요청을 사용하여 지정된 임계값 경보에 대해서만 모든 등록 정보를 나열할 수 있습니다.

```
GET /api/alert/v1/thresholds/threshold-uuid HTTP/1.1
```

임계값 경보 만들기

이 예에서는 초당 데이터 링크 바이트가 100000KB를 초과하는 이벤트의 임계값 경보를 만듭니다. 기타 모든 등록 정보에는 기본값이 있습니다.

사용자정의 임계값 경보를 만들려면 사용자에게 alert 범위의 allow_configure 권한 부여가 있어야 합니다.

요청 예:

```

POST /api/alert/v1/thresholds HTTP/1.1
Host: zfs-storage.example.com
X-Auth-User: root
X-Auth-Key: password
Content-Type: application/json
Content-Length: 50

{"statname": "datalink.kilobytes", "limit": 100000}

```

결과 예:

```

HTTP/1.1 201 Created
X-Zfssa-Appliance-API: 1.0
Content-Type: application/json
Content-Length: 321

```

Location: `/api/alert/v1/thresholds/threshold-uuid`

```
{
  "threshold": {
    "href": "/api/alert/v1/thresholds/threshold-uuid",
    "uuid": "threshold-uuid",
    "statname": "datalink.kilobytes",
    "type": "normal",
    "limit": 100000,
    "minpost": 300,
    "days": "all",
    "window_start": "none",
    "window_end": "00:00",
    "frequency": 300,
    "minclear": 300
  }
}
```

임계값 경보 수정

이 명령을 사용하여 지정된 임계값 경보의 등록 정보를 수정할 수 있습니다.

요청 예:

```
PUT /api/alert/v1/thresholds/threshold-uuid HTTP/1.1
Authorization: Basic Tm8gcGVla2luZyE=
Host: zfs-storage.example.com:215
```

```
{"days": "weekdays", "minpost": 120}
```

결과 예:

```
HTTP/1.1 202 Accepted
X-Zfssa-Appliance-API: 1.0
Content-Type: application/json
Content-Length: 326
```

```
{
  "threshold": {
    "href": "/api/alert/v1/thresholds/threshold-uuid",
    "uuid": "threshold-uuid",
    "statname": "datalink.kilobytes",
    "type": "normal",
    "limit": 100000,
    "minpost": 120,
    "days": "weekdays",
    "window_start": "none",
    "window_end": "00:00",
    "frequency": 300,
    "minclear": 300
  }
}
```

임계값 경보 삭제

지정된 임계값 경보를 삭제합니다.

요청 예:

```
DELETE /api/alert/v1/thresholds/threshold-uuid HTTP/1.1
Authorization: Basic Tm8gcGVla2luZyE=
Host: zfs-storage.example.com:215
```

결과 예:

```
HTTP/1.1 204 No Content
X-Zfssa-Appliance-API: 1.0
```

Analytics Services

Analytics를 통해 다양한 통계를 그래프에 실시간으로 표시하고 나중에 검색하도록 데이터를 기록할 수 있습니다. 장기 모니터링과 단기 분석을 모두 수행할 수 있습니다. Analytics는 DTrace를 사용하여 운영체제 스택의 여러 계층을 상세하게 분석할 수 있도록 동적으로 사용자 정의 통계를 만듭니다.

Analytics 명령

<http://hostname/api/analytics/v{1|2}>에서 다음 Analytics Services를 사용할 수 있습니다.

표 12 Analytics 명령

요청	경로 /analytics/v{1 2}에 추가	설명
GET	/analytics/v{1 2}만 사용	Analytics Services 정보를 나열합니다.
POST	/worksheets	새 Analytics 데이터 세트를 만듭니다.
GET	/worksheets/worksheet	지정된 Analytics 데이터 세트 등록 정보를 가져옵니다.
GET	/worksheets	모든 Analytics 데이터 세트 객체를 나열합니다.
PUT	/worksheets/worksheet	지정된 Analytics 데이터 세트 객체를 수정합니다.
DELETE	/worksheets/worksheet	지정된 워크시트 객체를 삭제합니다.
PUT	/worksheets/worksheet/suspend	모든 워크시트 데이터 세트를 일시중단합니다.
PUT	/worksheets/worksheet/resume	모든 워크시트 데이터 세트를 재개합니다.
POST	/worksheets/worksheet/datasets	새 워크시트 데이터 세트를 만듭니다.
GET	/worksheets/worksheet/datasets/dataset	지정된 워크시트 데이터 세트 등록 정보를 가져옵니다.
GET	/worksheets/worksheet/datasets	모든 워크시트 데이터 세트 객체를 나열합니다.
PUT	/worksheets/worksheet/datasets/dataset	지정된 워크시트 데이터 세트 객체를 수정합니다.
DELETE	/worksheets/worksheet/datasets/dataset	지정된 데이터 세트 객체를 삭제합니다.
POST	/datasets	새 Analytics 데이터 세트를 만듭니다.
GET	/datasets/dataset	지정된 Analytics 데이터 세트 등록 정보를 가져옵니다.
GET	/datasets	모든 Analytics 데이터 세트 객체를 나열합니다.
PUT	/datasets/dataset	지정된 Analytics 데이터 세트 객체를 수정합니다.

요청	경로 /analytics/v{1 2}에 추가	설명
DELETE	/datasets/dataset	지정된 데이터 세트 객체를 삭제합니다.
PUT	/datasets	모든 데이터 세트를 일시중단하거나 재개합니다.
PUT	/datasets/dataset/data	이 데이터 세트를 저장합니다(저장되지 않은 경우).
DELETE	/datasets/dataset/data	이 데이터 세트에서 지정된 [단위]의 데이터를 제거합니다.
GET	/settings	Analytics 설정을 나열합니다.
PUT	/settings	Analytics 설정을 수정합니다.

Analytics 설정

다음 표에서 설명되는 등록 정보를 통해 모든 분석 데이터를 수집하고, 데이터 보유 시간(시)을 설정하며, 호스트 이름 조회 정책을 설정할 수 있습니다.

등록 정보	설명
retain_second_data	초 단위 데이터에 대한 시간의 보존 간격
retain_minute_data	분 단위 데이터에 대한 시간의 보존 간격
retain_hour_data	시간 단위 데이터에 대한 시간의 보존 간격
hostname_lookup	호스트 이름 조회 정책

설정 가져오기

이 명령은 Analytics 등록 정보의 현재 값을 가져옵니다.

요청 예:

```
GET /api/analytics/v1/settings HTTP/1.1
Authorization: Basic Tm8gcGVla2luZyE=
Host: zfs-storage.example.com:215
Accept: application/json
```

결과 예:

```
HTTP/1.1 200 OK
Content-Type: application/json
Content-Length: 131
X-Zfssa-Analytics-API: 1.0

{
  "settings": {
    "href": "/api/analytics/v1/settings",
    "retain_hour_data": 600,
    "retain_minute_data": 400,
    "retain_second_data": 200,
```

```

    "hostname_lookup": true
  }
}

```

설정 수정

설정 수정 명령은 데이터 보존 값 및 호스트 이름 조회 정책과 같은 Analytics 설정을 수정하는 데 사용됩니다.

요청 예:

```

PUT /api/analytics/v1/settings HTTP/1.1
Authorization: Basic Tm8gcGVla2luZyE=
Host: zfs-storage.example.com:215
Content-Type: application/json
Content-Length: 60

```

```

{"retain_hour_data":600, "retain_minute_data":400, "retain_second_data":200,
 "hostname_lookup":true}

```

결과 예:

```

HTTP/1.1 202 Accepted
Content-Type: application/json
Content-Length: 101
X-Zfssa-Analytics-API: 1.0

```

```

{
  "settings": {
    "href": "/api/analytics/v1/settings",
    "retain_hour_data": 600,
    "retain_minute_data": 400,
    "retain_second_data": 200,
    "hostname_lookup": true
  }
}

```

Analytics 워크시트

워크시트는 통계가 그래프로 표시되는 BUI 화면입니다. 동시에 여러 가지 통계를 그릴 수 있으며 워크시트에 제목을 지정하고 나중에 보도록 저장할 수 있습니다. 워크시트를 저장하면 열려 있는 모든 통계에 대해 아카이브 작업이 자동으로 실행됩니다. 즉, 열려 있는 모든 통계를 읽고 아카이브하는 작업이 영원히 계속됩니다. 워크시트 명령은 BUI에서 사용할 수 있는 워크시트를 관리하는 데 사용할 수 있습니다.

다음 표에서는 Analytics 워크시트에서 사용되는 등록 정보를 보여줍니다.

등록 정보	설명
ctime	이 워크시트가 생성된 시간 및 날짜

등록 정보	설명
mtime	이 워크시트가 마지막으로 수정된 시간 및 날짜
name	이 워크시트의 이름
owner	이 워크시트의 소유자
uuid	이 워크시트에 대한 범용 고유 식별자

워크시트 나열

현재 구성된 모든 Analytics 워크시트를 나열합니다.

요청 예:

```
GET /api/analytics/v1/worksheets HTTP/1.1
Authorization: Basic Tm8gcGVla2luZyE=
Host: zfs-storage.example.com:215
Accept: application/json
```

결과 예:

```
HTTP/1.1 200 OK
Content-Type: application/json
Content-Length: 237
X-Zfssa-Analytics-API: 1.0

{
  "worksheets": [{
    "href": "/api/analytics/v1/worksheets/ab59bcbc...",
    "uuid": "ab59bcbc-080a-cf1a-98c9-9f485bc3a43d"
  }, {
    "href": "/api/analytics/v1/worksheets/bb3ee729...",
    "uuid": "bb3ee729-080a-cf1a-98c9-9f485bc3a43d"
  }]
}
```

Analytics 워크시트 가져오기

단일 Analytics 워크시트를 가져옵니다.

요청 예:

```
GET /api/analytics/v1/worksheets/ab59bcbc-080a-cf1a-98c9-9f485bc3a43d
HTTP/1.1
Authorization: Basic Tm8gcGVla2luZyE=
Host: zfs-storage.example.com:215
Accept: application/json
```

결과 예:

```
HTTP/1.1 200 OK
Content-Type: application/json
Content-Length: 237
```

```
X-Zfssa-Analytics-API: 1.0

{
  "worksheet": {
    "ctime": "Thu Jun 13 2013 02:17:14 GMT+0000 (UTC)",
    "href": "/api/analytics/v1/worksheets
             /ab59bcbc-080a-cf1a-98c9-9f485bc3a43d",
    "mtime": "Sun Jun 23 2013 16:22:01 GMT+0000 (UTC)",
    "name": "myworksheet",
    "owner": "root",
    "uuid": "ab59bcbc-080a-cf1a-98c9-9f485bc3a43d"
  }
}
```

워크시트 만들기

새 Analytics 워크시트를 만듭니다.

요청 예:

```
POST /api/analytics/v1/worksheets HTTP/1.1
Authorization: Basic Tm8gcGVla2luZyE=
Content-Type: application/json
Content-Length: 26
```

```
{"name": "myworksheet"}
```

결과 예:

```
HTTP/1.1 201 Created
Content-Length: 280
Location: /api/analytics/v1/worksheets/bb3ee729-4480-4609-89b2-fae2dc016bec
```

```
{
  "worksheet": {
    "uuid": "bb3ee729-4480-4609-89b2-fae2dc016bec",
    "name": "myworksheet",
    "owner": "root",
    "ctime": "Fri Aug 23 2013 20:35:00 GMT+0000 (UTC)",
    "mtime": "Fri Aug 23 2013 20:35:00 GMT+0000 (UTC)",
    "href": "/api/analytics/v1/worksheets
             /bb3ee729-4480-4609-89b2-fae2dc016bec"
  }
}
```

워크시트 이름 바꾸기

저장된 워크시트의 이름을 바꿉니다.

요청 예:

```
PUT /api/analytics/v1/worksheets/a442e761-4048-4738-b95f-be0824d7ed09
Authorization: Basic Tm8gcGVla2luZyE=
Content-Type: application/json
Content-Length: 26
```

```
{"name": "test"}
```

결과 예:

```
HTTP/1.1 202 Accepted
Date: Tue, 20 Dec 2016 00:33:06 GMT
Server: TwistedWeb/192.0.2
Content-Length: 279
X-Zfssa-Analytics-API: 1.1
X-Zfssa-API-Version: 1.0
Content-Type: application/json; charset=utf-8
```

```
{
  "worksheet": {
    "href": "/api/analytics/v1/worksheets/a442e761-4048-4738-b95f-be0824d7ed09",
    "uuid": "a442e761-4048-4738-b95f-be0824d7ed09",
    "name": "test",
    "owner": "root",
    "ctime": "Wed Dec 14 2016 03:58:28 GMT+0000 (UTC)",
    "mtime": "Tue Dec 20 2016 00:25:57 GMT+0000 (UTC)"
  }
}
```

워크시트 삭제

Analytics 워크시트를 삭제합니다. 이 예에서는 워크시트 이름이 워크시트 식별자로 사용되지 만 href에서 식별된 uuid도 사용될 수 있습니다. 이 명령의 동작은 워크시트를 삭제하는 CLI 명령의 동작과 일치합니다.

요청 예:

```
DELETE /api/analytics/v1/worksheets/name=myworksheet HTTP/1.1
Authorization: Basic Tm8gcGVla2luZyE=
Content-Type: application/json
Content-Length: 26
```

결과 예:

```
HTTP/1.1 204 No Content
X-Zfssa-Analytics-API: 1.0
```

워크시트 데이터 세트 나열

지정된 워크시트의 모든 데이터 세트를 나열합니다.

다음 표에서는 데이터 세트 구성에서 사용되는 등록 정보를 보여줍니다.

등록 정보	설명
name	이 데이터 세트에 대한 기본 통계의 이름

등록 정보	설명
drilldowns	현재 강조 표시된 드릴다운(있는 경우)
seconds	이 데이터 세트에 대해 표시되는 시간(초)

요청 예:

```
GET /api/analytics/v1/worksheets/name=myworksheet/datasets HTTP/1.1
Authorization: Basic Tm8gcGV1a2luZyE=
Host: zfs-storage.example.com:215
Accept: application/json
```

워크시트 데이터 세트 추가

워크시트 데이터 세트를 만듭니다.

요청 예:

```
POST /api/analytics/v1/worksheets/name=myworksheet/datasets HTTP/1.1
Authorization: Basic Tm8gcGV1a2luZyE=
Content-Type: application/json
Content-Length: 26
```

```
{"name": "nfs4.ops", "seconds": 300}
```

결과 예:

```
HTTP/1.1 201 Created
Content-Type: application/json
X-Zfssa-Analytics-API: 1.0
Location: /api/analytics/v1/worksheets/name=me/datasets/nfs4.ops
Content-Length: 162
```

```
{
  "dataset": {
    "href": "/api/analytics/v1/worksheets/name=me/datasets/dataset-008",
    "name": "nfs4.ops",
    "width": 0,
    "drilldowns": [],
    "seconds": 300,
    "time": ""
  }
}
```

워크시트 데이터 세트 수정

기존 워크시트 데이터 세트를 수정합니다.

요청 예:

```
PUT /api/analytics/v1/worksheets/name=myworksheet/datasets/dataset-008
```

```

HTTP/1.1
Authorization: Basic Tm8gcGVla2luZyE=
Content-Type: application/json
Content-Length: 26
    
```

```

{"seconds": 60}
    
```

결과 예:

```

HTTP/1.1 202 Accepted
Content-Type: application/json
Content-Length: 161
X-Zfssa-Analytics-API: 1.0
    
```

```

{
  "dataset": {
    "href": "/api/analytics/v1/worksheets/name=me/datasets/dataset-008",
    "name": "nfs4.ops",
    "width": 0,
    "drilldowns": [],
    "seconds": 60,
    "time": ""
  }
}
    
```

Analytics 데이터 세트

Analytics 데이터 세트에는 다음 등록 정보가 사용됩니다. `suspended`를 제외한 모든 등록 정보는 변경할 수 없습니다.

등록 정보	설명
<code>name</code>	이 데이터 세트에 대한 기본 통계의 이름
<code>grouping</code>	이 통계가 속한 그룹
<code>explanation</code>	기본 통계의 설명
<code>incore</code>	인코어 데이터 세트 데이터의 바이트
<code>size</code>	온디스크 데이터 세트 데이터의 바이트
<code>suspended</code>	데이터 세트가 현재 일시 중지되어 있는지 여부를 나타내는 부울
<code>activity</code>	보류 중인 데이터 세트 작동 플래그

사용 가능한 데이터 세트:

- `ad.avglatency`
- `ad.avglatency[op]`
- `ad.avglatency[result]`
- `ad.binds`
- `ad.binds[hostname]`

- ad.binds[result]
- ad.ops
- ad.ops[op]
- ad.ops[result]
- arc.accesses[hit/miss]
- arc.l2_accesses[hit/miss]
- arc.l2_size
- arc.size
- arc.size[component]
- cpu.utilization
- cpu.utilization[mode]
- dnlc.accesses[hit/miss]
- fc.bytes
- fc.ops
- ftp.kilobytes
- http.reqs
- io.bytes
- io.bytes[op]
- io.disks[utilization=95][disk]
- io.ops
- io.ops[disk]
- io.ops[op]
- iscsi.bytes
- iscsi.ops
- metacap.bytesused
- metacap.percentused
- ndmp.diskkb
- nfs2.ops
- nfs2.ops[op]
- nfs3.ops
- nfs3.ops[op]
- nfs4.ops
- nfs4.ops[op]
- nfs4-1.ops
- nfs4-1.bytes
- nic.kilobytes
- nic.kilobytes[device]
- nic.kilobytes[direction]

- sftp.kilobytes
- smb.ops
- smb.ops[op]

데이터 세트 나열

구성된 모든 Analytics 데이터 세트를 나열합니다.

요청 예:

```
GET /api/analytics/v1/datasets HTTP/1.1
Authorization: Basic Tm8gcGVla2luZyE=
Host: zfs-storage.example.com:215
Accept: application/json
```

결과 예:

```
HTTP/1.1 200 OK
Content-Type: application/json
Content-Length: 237
X-Zfssa-Analytics-API: 1.0
```

```
{
  "datasets": [{
    "dataset": "dataset-000",
    "href": "/api/analytics/v1/datasets/arc.accesses[hit/miss]",
    "name": "arc.accesses[hit/miss]"
  }, {
    "dataset": "dataset-001",
    "href": "/api/analytics/v1/datasets/arc.l2_accesses[hit/miss]",
    "name": "arc.l2_accesses[hit/miss]",
  }, {
    "dataset": "dataset-002",
    "href": "/api/analytics/v1/datasets/arc.l2_size",
    "name": "arc.l2_size",
  }, {
    "dataset": "dataset-003",
    "href": "/api/analytics/v1/datasets/arc.size",
    "name": "arc.size",
  }, {
    "dataset": "dataset-004",
    "href": "/api/analytics/v1/datasets/arc.size[component]",
    "name": "arc.size[component]",
  }, {
    ...
  }]
}
```

데이터 세트 가져오기

지정된 데이터 세트에서 등록 정보를 가져옵니다.

요청 예:

```
GET /api/analytics/v1/datasets/nfs4.ops HTTP/1.1
Authorization: Basic Tm8gcGVla2luZyE=
Host: zfs-storage.example.com:215
Accept: application/json
```

결과 예:

```
HTTP/1.1 200 OK
Content-Type: application/json
Content-Length: 237
X-Zfssa-Analytics-API: 1.0
```

```
{
  "dataset": {
    "activity": "none",
    "dataset": "dataset-030",
    "explanation": "NFSv4 operations per second",
    "grouping": "Protocol",
    "href": "/api/analytics/v1/datasets/nfs4.ops",
    "incore": 296128,
    "name": "nfs4.ops",
    "size": 53211540,
    "suspended": false
  }
}
```

데이터 세트 만들기

새 데이터 세트를 만듭니다.

요청 예:

```
POST /api/analytics/v1/datasets HTTP/1.1
X-Auth-User: root
X-Auth-Key: password
Content-Type: application/json
Content-Length: 26
```

```
{"statistic": "test.sine"}
```

결과 예:

```
HTTP/1.1 201 Created
Content-Type: application/json
Content-Length: 200
Location: /api/analytics/v1/datasets/test.sine
```

```
{
  "dataset": {
    "href": "/api/analytics/v1/datasets",
    "name": "test.sine",
    "grouping": "Test",
    "explanation": "sine per second",
    "incore": 34752,
    "size": 31912,
    "suspended": false,
    "activity": "none"
  }
}
```

데이터 세트 수정

데이터 세트 수정 명령은 단일 데이터 세트의 데이터 수집을 일시 중단하거나 재개하는 데 사용됩니다.

일시 중지 요청 예:

```
POST /api/analytics/v1/datasets/nfs4.ops
{"suspended":true}
```

재개 요청 예:

```
POST /api/analytics/v1/datasets/nfs4.ops
{"suspended":false}
```

결과 예:

```
HTTP/1.1 202 Accepted
Content-Type: application/json
Content-Length: 228
X-Zfssa-Analytics-API: 1.0

{
  "dataset" {
    ...
    "suspended": false
  }
}
```

데이터 세트 삭제

데이터 세트를 삭제합니다.

요청 예:

```
DELETE /api/analytics/v1/datasets/test.sine HTTP/1.1
```

결과 예:

```
HTTP/1.1 204 No Content
X-Zfssa-Analytics-API: 1.0
```

데이터 세트 저장

데이터 세트를 저장합니다.

요청 예:

```
PUT /api/analytics/v1/datasets/nfs4.ops/data
```

결과 예:

HTTP/1.1 202 Accepted

데이터 세트 데이터 제거

다음 표에서는 데이터 세트 제거에서 사용되는 질의 매개변수를 보여줍니다.

매개변수	설명
granularity	단위를 제거합니다. 데이터 세트 내의 데이터는 second, minute 또는 hour 단위 값에서 제거될 수 있습니다.
endtime	지정된 endtime 이전에 수집된 데이터를 제거합니다. 이 시간 값의 형식은 "질의 매개변수: start" [21]를 참조하십시오.

요청 예:

다음 예에서는 nfs4.ops 데이터 세트에서 초당, 분당, 시간당 데이터를 모두 삭제합니다. 매일, 매주, 매월 또는 매년 수집된 데이터는 데이터 세트에 유지됩니다.

```
DELETE /api/analytics/v1/datasets/nfs4.ops/data?granularity=hour
```

결과 예:

HTTP/1.1 204 No Content

요청 예:

다음 예에서는 nfs4.ops 데이터 세트에서 지정된 endtime 이전에 수집된 초당, 분당, 시간당 데이터를 모두 삭제합니다.

```
DELETE /api/analytics/v1/datasets/nfs4.ops/data?granularity=hour&endtime=20130910T00:00:00
```

데이터 세트 데이터 가져오기

활성 분석 데이터 세트에서 데이터를 반환합니다. 초 단위 데이터 검색 및 세분화된 단위의 데이터 검색이 모두 지원됩니다.

다음 표에서는 데이터 세트 데이터를 가져오기 위한 시간 기반 질의 매개변수를 보여줍니다.

매개변수	설명
start	샘플 데이터 수집을 시작할 시간 또는 데이터 수집을 시작할 샘플 인덱스입니다. start 시간 값은 특정 시간일 수도 있고, 키워드 now일 수도 있습니다. 특정 시간 값의 형식은 "질의 매개변수: start" [21]를 참조하십시오. 기본 start 시간은 현재 시간에서 seconds 값을 뺀 값입니다.

매개변수	설명
seconds	샘플 데이터를 수집할 시간(초)입니다. 기본값은 1입니다. span 및 granularity 매개변수가 지정된 경우 seconds 매개변수는 무시됩니다.
span	샘플 데이터를 수집할 기간(minute, hour, day, week, month 또는 year)입니다.
granularity	지정된 지속 시간 내에서 데이터 포인트 평균이 제공되는 단위(minute, hour, day, week, month 또는 year)입니다.

시작 시간이 미래일 수는 없습니다. 데이터를 수집할 시간(초)이 현재 시간 이후로 넘어갈 경우 서버에서 데이터를 반환하기 전에 각 샘플을 기다립니다.

세분화된 단위로 데이터를 검색하려면 span 및 granularity 매개변수의 조합을 사용합니다. span 및 granularity가 사용되는 경우 seconds 매개변수는 무시됩니다. span 또는 granularity가 잘못 입력되는 경우 요청이 무시되며 seconds 매개변수가 대신 사용됩니다. 잘못된 요청이거나 지원되지 않는 요청인 경우 "Input span and granularity are not supported(입력 지속 시간 및 단위는 지원되지 않습니다)."라는 오류 메시지가 표시됩니다.

span 매개변수와 granularity 매개변수는 다음과 같은 방법으로 결합할 수 있습니다.

- span이 minute인 경우 granularity는 minute일 수만 있습니다.
- span이 hour인 경우 granularity는 minute 또는 hour일 수 있습니다.
- span이 day인 경우 granularity는 minute, hour 또는 day일 수 있습니다.
- span이 week인 경우 granularity는 hour, day 또는 week일 수 있습니다.
- span이 month인 경우 granularity는 day, week 또는 month일 수 있습니다.
- span이 year인 경우 granularity는 week, month 또는 year일 수 있습니다.

다음 표는 반환되는 데이터 세트 데이터 등록 정보를 보여줍니다.

등록 정보	설명
data	샘플 데이터의 배열
sample	샘플 데이터의 인덱스 번호
startTime	sample이 반환된 시간
min	지정된 단위 내 초당 최소값
max	지정된 단위 내 초당 최대값

2초 동안의 라이브 데이터를 수집하는 요청의 예:

```
GET /api/analytics/v1/datasets/io.ops[op]/data?start=now&seconds=2 HTTP/1.1
Authorization: Basic Tm8gcGVla2luZyE=
Host: zfs-storage.example.com:215
Accept: text/x-yaml
```

결과 예:

```
HTTP/1.1 200 OK
```

```

Content-Type: text/x-yaml
X-Zfssa-Analytics-API: 1.0
Transfer-Encoding: chunked

{
  "data": [
    {
      "sample": 457642682,
      "data": {
        "data": [
          {
            "key": "write",
            "value": 199
          }
        ],
        "value": 199
      },
      "startTime": "20200818T18:43:47",
      "samples": 457642683
    },
    {
      "sample": 457642683,
      "data": {
        "data": [
          {
            "key": "write",
            "value": 289
          }
        ],
        "value": 289
      },
      "startTime": "20200818T18:43:48",
      "samples": 457642684
    }
  ]
}

```

지속 시간 1주 내 7일 동안의 데이터를 수집하는 요청의 예:

```

GET /api/analytics/v1/datasets/io.ops[op]/data?
start=20200811T15:00:00&granularity=day&span=week
HTTP/1.1
Authorization: Basic Tm8gcGVla2luZyE=
Host: zfs-storage.example.com:215
Accept: text/x-yaml

```

다음 요청은 io.ops[op] 데이터 세트가 "작업 유형별로 분석된 초당 I/O 작업"(읽기 또는 쓰기) 임을 보고합니다.

```
GET /api/analytics/v1/datasets/io.ops[op]
```

결과 예. 간략히 나타내기 위해 7개 출력 샘플 중 5개가 생략됩니다.

```

HTTP/1.1 200 OK
Content-Type: text/x-yaml
X-Zfssa-Analytics-API: 1.0
Transfer-Encoding: chunked

```

```

{
  "data": [
    {
      "sample": 457197423,
      "data": {

```

```

    "max": 3156,
    "data": [
      {
        "max": 588,
        "key": "read",
        "value": 6,
        "min": 0
      },
      {
        "max": 3156,
        "key": "write",
        "value": 45,
        "min": 0
      }
    ],
    "value": 52,
    "min": 0
  },
  "startTime": "20200811T15:00:00",
  "samples": 457644011
},
{
  "sample": 457283823,
  "data": {
    "max": 3675,
    "data": [
      {
        "max": 588,
        "key": "read",
        "value": 6,
        "min": 0
      },
      {
        "max": 3675,
        "key": "write",
        "value": 45,
        "min": 0
      }
    ]
  },
  "value": 52,
  "min": 0
},
"startTime": "20200812T15:00:23",
"samples": 457644011
}
]
}

```

sample 값을 start 값으로 사용할 수도 있습니다. 다음 요청의 결과는 지정된 1초 샘플에서 가져온 데이터입니다.

GET /api/analytics/v1/datasets/io.ops[op]/data?start=457642682 HTTP/1.1

하드웨어 서비스

이 절에서는 하드웨어 클러스터, 새시 및 구성요소의 관리에 대해 설명합니다.

클러스터

`cluster` 명령은 클러스터링을 설정하고 클러스터화된 리소스를 관리합니다.

표 13 클러스터 명령

요청	경로 <code>/hardware/v{1 2}</code> 에 추가	설명
GET	<code>/cluster</code>	클러스터 등록 정보 및 리소스를 나열합니다.
GET	<code>/cluster/resources/resource-path</code>	지정된 클러스터 리소스에 대한 등록 정보를 나열합니다.
PUT	<code>/cluster/resources/resource-path</code>	지정된 클러스터 리소스를 수정합니다.
PUT	<code>/cluster/failback</code>	클러스터 피어에 지정된 모든 리소스를 페일백합니다.
PUT	<code>/cluster/takeover</code>	클러스터 피어에 지정된 모든 리소스를 인계합니다.
PUT	<code>/cluster/unconfig</code>	클러스터화된 어플라이언스를 독립형 모드로 구성 해제합니다.
GET	<code>/cluster/links</code>	클러스터 카드 링크 상태를 표시합니다.
PUT	<code>/cluster/setup</code>	초기 클러스터 설정을 수행합니다.

클러스터 등록 정보 나열

`cluster` 명령은 현재 클러스터 구성 상태 및 리소스 등록 정보를 나열합니다.

요청 예:

```
GET /api/hardware/v2/cluster HTTP/1.1
Authorization: Basic Tm8gcGV1a2luZyE=
Host: zfs-storage.example.com:215
Accept: application/json
```

결과 예:

```

HTTP/1.1 200 OK
X-Zfssa-Appliance-API: 1.0
Content-Type: application/json
Content-Length: 529
X-Zfssa-API: 1.0

{
  "cluster": {
    "state": "AKCS_OWNER",
    "description": "Active (takeover completed)",
    "peer_asn": "d23331e6-41f4-6a15-ac09-a4353e33b43a",
    "peer_hostname": "peer-1",
    "peer_state": "AKCS_STRIPPED",
    "peer_description": "Ready (waiting for failback)",
    "resources": [
      {
        "owner": "peer-1",
        "type": "private",
        "user_label": "peer-1",
        "details": [
          "ipaddr"
        ],
        "href": "/api/hardware/v2/cluster/resources/net/vnic1"
      },
      {
        "owner": "peer-1",
        "type": "singleton",
        "user_label": "",
        "details": [
          "8.03T"
        ],
        "href": "/api/hardware/v2/cluster/resources/zfs/cas1"
      },
      {
        "owner": "peer-2",
        "type": "singleton",
        "user_label": "",
        "details": [
          "18.7T"
        ],
        "href": "/api/hardware/v2/cluster/resources/zfs/cas2"
      }
    ]
  }
}

```

다음 예에서와 같이 `cluster` 명령으로 나열되는 리소스 중 하나의 `href` 등록 정보를 사용하여 특정 클러스터 리소스에 대해서만 등록 정보를 나열할 수 있습니다.

```
GET /api/hardware/v2/cluster/resources/net/vnic1 HTTP/1.1
```

클러스터 리소스 수정

`cluster` 명령으로 나열되는 리소스 중 하나의 `href` 등록 정보와 함께 `PUT` 요청을 사용하여 해당 클러스터 리소스에 대한 등록 정보를 설정할 수 있습니다.

클러스터 링크 상태

`cluster/links` 명령은 클러스터 카드의 현재 링크 상태를 반환합니다.

초기 클러스터 설정을 수행하기 전에 모든 링크가 AKCIOS_ACTIVE 상태가 되도록 `cluster/links`를 사용합니다. 연결이 AKCIOS_ACTIVE 상태가 아닐 경우 다른 시스템이 다시 시작/재부트 중이거나 링크가 케이블을 통해 제대로 연결되지 않았거나 클러스터 케이블이 커넥터를 고정시키고 있지 않는 것일 수 있습니다. 클러스터 케이블 연결 지침은 [Oracle ZFS Storage Appliance 케이블 연결 설명서의 "클러스터 케이블 연결"](#)을 참조하십시오.

`cluster/links` 명령의 출력은 다른 컨트롤러 모델의 출력과 다릅니다. ZS9-2 컨트롤러의 경우 두 클러스터 I/O 링크에 대해 상태가 반환됩니다. 기타 모든 컨트롤러의 경우 세 클러스터 I/O 링크에 대해 상태가 반환됩니다. 클러스터 I/O 링크에 대한 설명은 ["Cluster Interconnect I/O in Oracle ZFS Storage Appliance Administration Guide, Release OS8.8.x"](#)를 참조하십시오.

요청 예:

```
GET /api/hardware/v2/cluster/links HTTP/1.1
Authorization: Basic Tm8gcGVla2luZyE=
Host: zfs-storage.example.com:215
Accept: application/json
```

Oracle ZFS Storage ZS9-2에 대한 결과 예:

```
HTTP/1.1 200 OK
X-Zfssa-Appliance-API: 1.0
Content-Type: application/json
Content-Length: 84

{
  "links": {
    "lio_dev/i40e0 = AKCIOS_ACTIVE\n
    lio_dev/i40e1 = AKCIOS_ACTIVE"
  }
}
```

ZS7-2 또는 ZS5-2 컨트롤러에 대한 결과 예:

```
HTTP/1.1 200 OK
X-Zfssa-Appliance-API: 1.0
Content-Type: application/json
Content-Length: 181

{
  "links": "\n\tclustron_ng3:0/clustron_uart:0 = AKCIOS_ACTIVE
\n\tclustron_ng3:0/clustron_uart:1 = AKCIOS_ACTIVE
\n\tclustron_ng3:0/dlpi:0 = AKCIOS_ACTIVE\n\n"
}
```

다른 컨트롤러는 유사한 `links` 출력을 보여줍니다. `/clustron` 또는 `/dlpi` 앞에 오는 부분만 다릅니다. 예를 들어, 이전 예의 `clustron3_ng3:0`은 ZS5-4 컨트롤러의 경우 `clustron3:0`, ZS4-4 컨트롤러의 경우 `clustron2:0`, ZS3-2 컨트롤러의 경우 `clustron2_embedded:0`이 됩니다.

클러스터 관리 명령

클러스터 관리에는 `failback`, `takeover` 및 `unconfig`가 포함됩니다. 성공 시 두 명령은 HTTP 상태 202(Accepted)를 반환합니다. 클러스터가 명령을 수락하는 올바른 상태가 아닌 경우 HTTP 상태 409(Conflict)가 반환됩니다.

피어 실패가 감지될 때마다 인계가 자동으로 시도됩니다. 관리자도 인계를 수행할 수 있습니다.

페일백은 관리자가 수행해야 합니다. 페일백은 비동기 작업입니다. REST 클라이언트가 `failback` 명령을 전송하면 요청이 성공적으로 수신된 경우 HTTP 상태 202가 반환됩니다. 페일백 진행 상황을 모니터링하려는 클라이언트는 경보를 수신하거나 클러스터 상태를 폴링할 수 있습니다.

인계 및 페일백에 대한 자세한 내용은 [“Cluster Takeover and Failback” in Oracle ZFS Storage Appliance Administration Guide, Release OS8.8.x](#)을 참조하십시오.

요청 예:

```
PUT /api/hardware/v2/cluster/failback HTTP/1.1
```

클러스터 노드 구성을 해제하면 독립형 작업에 대한 노드가 구성됩니다. 일반적으로 클러스터 노드 구성 해제는 단독으로 수행하지 마십시오. 클러스터 노드 구성 해제는 소거식입니다. 구성 해제는 `unconfig` 명령 이상의 작업과 관련이 있습니다. 자세한 내용은 [“Unconfiguring a Cluster Node” in Oracle ZFS Storage Appliance Administration Guide, Release OS8.8.x](#)를 참조하십시오.

클러스터 설정

설정은 초기 클러스터 구성의 1단계입니다. 자세한 내용은 [“Upgrading a Standalone Appliance to a Clustered Configuration \(BUI\)” in Oracle ZFS Storage Appliance Administration Guide, Release OS8.8.x](#)를 참조하십시오.

`cluster/setup` 명령은 시스템에 대한 초기 클러스터 구성을 수행합니다. `nodename` 및 `password` 등 록 정보에 대한 값을 지정합니다. 설정이 성공하면 HTTP 상태 202 Accepted가 반환됩니다.

다음 조건이 모두 존재하지 않을 경우 `cluster/setup` 명령이 실패합니다.

- 모든 클러스터 링크가 AKCIOS_ACTIVE 상태입니다. [“클러스터 링크 상태” \[79\]](#)를 참조하십시오.
- 피어의 전원이 켜져 있지만 피어가 구성되어 있지 않습니다.

주 - 초기 클러스터 구성 설정은 완료하는 데 몇 분 정도 걸릴 수 있습니다.

요청 예:

```
PUT /api/hardware/v2/cluster/setup HTTP/1.1
Authorization: Basic Tm8gcGV1a2luZyE=
Host: zfs-storage.example.com:215
Accept: application/json

{"nodename": "zfs-storage-2", "password": "password"}
```

새시

하드웨어 명령은 어플라이언스 하드웨어 새시 및 구성요소 목록을 가져오는 데 사용됩니다.

표 14 하드웨어 명령

요청	경로 /hardware/v{1 2}에 추가	설명
GET	/chassis	하드웨어 새시를 나열합니다.
GET	/chassis/chassis	지정된 하드웨어 새시 등록 정보를 가져옵니다.
PUT	/chassis/chassis	지정된 하드웨어 새시 등록 정보를 수정합니다.
GET	/chassis/chassis/fru_type	하드웨어 새시 구성요소를 나열합니다.
GET	/chassis/chassis/fru_type/fru	지정된 새시 구성요소 등록 정보를 가져옵니다.
PUT	/chassis/chassis/fru_type/fru	하드웨어 새시 구성요소 등록 정보를 수정합니다.

새시 나열

새시 가져오기 명령은 인수를 사용하지 않고 시스템 새시 객체 목록을 반환합니다. 성공한 명령에 대해 HTTP 상태 200(OK)이 반환됩니다.

등록 정보	유형	설명
name	string	새시 이름
model	string	새시 모델 번호
manufacturer	string	새시 제조업체
serial	string	새시 일련 번호
revision	string	새시 개정 레벨
part	string	새시 교체 부품 번호
type	string	새시 스토리지 유형
faulted	boolean	결함 표시기
uuid	string	새시 UUID 식별자

요청 예:

```
GET /api/hardware/v1/Chassis HTTP/1.1
Host: zfs-storage.example.com:215
Accept: application/json
```

결과 예:

```
HTTP/1.1 200 OK
Content-Length: 788
Content-Type: application/json
X-Zfssa-Appliance-API: 1.0
```

```
{
  "hardware": [{
    "faulted": false,
    "href": "/api/hardware/v1/chassis/chassis-000",
    "manufacturer": "Oracle",
    "model": "Oracle ZFS Storage ZS3-2",
    "name": "hostname",
    "rpm": "--",
    "serial": "1211FM200C",
    "type": "system"
  }, {
    "faulted": false,
    "href": "/api/hardware/v1/chassis/chassis-001",
    "locate": false,
    "manufacturer": "Oracle",
    "model": "Oracle Storage DE2-24C",
    "name": "1235FM4002",
    "part": "7046842",
    "path": 2,
    "revision": "0010",
    "rpm": 7200,
    "serial": "1235FM4002",
    "type": "storage"
  }, {
    "faulted": false,
    "href": "/api/hardware/v1/chassis/chassis-002",
    "locate": false,
    "manufacturer": "Oracle",
    "model": "Oracle Storage DE2-24P",
    "name": "50050cc10c206b96",
    "part": "7046836",
    "path": 2,
    "revision": "0010",
    "rpm": 10000,
    "serial": "50050cc10c206b96",
    "type": "storage"
  }
  ]
}
```

새시 구성요소 가져오기

이 명령은 지정된 새시 내 모든 하드웨어 구성요소를 반환합니다. 성공한 명령에 대해 HTTP 상태 200(OK)이 반환됩니다.

요청 예:

```
GET /api/hardware/v1/chassis/chassis-001 HTTP/1.1
Host: zfs-storage.example.com:215
```

Accept: application/json

결과 예:

HTTP/1.1 200 OK

Content-Type: application/json

```
{
  "chassis": {
    "type": "storage"
    "faulted": false,
    "href": "/api/hardware/v1/chassis/chassis-001",
    "locate": false,
    "manufacturer": "Oracle",
    "model": "Oracle Storage DE2-24C",
    "name": "1235FM4002",
    "part": "7046842",
    "path": 2,
    "revision": "0010",
    "rpm": 7200,
    "serial": "1235FM4002",
    "disk": [{
      "device": "c0t5000CCA01A76A2B8d0",
      "faulted": false,
      "href": "/api/hardware/v1/chassis/chassis-001/disk/disk-000",
      "interface": "SAS",
      "label": "HDD 0",
      "locate": false,
      "offline": false,
      "readytoremove": false,
      "manufacturer": "HITACHI",
      "model": "H7230AS60SUN3.0T",
      "pathcount": 2,
      "present": true,
      "revision": "A310",
      "rpm": 7200,
      "serial": "001210R37LVD-----YHJ37LVD",
      "size": 3000592982016,
      "type": "data",
      "use": "peer"
    }, {
      "href": "/api/hardware/v1/chassis/chassis-001/disk/disk-001",
      ...
    }, {
      "href": "/api/hardware/v1/chassis/chassis-001/disk/disk-002",
      ...
    },
    ... {
      "href": "/api/hardware/v1/chassis/chassis-001/disk/disk-023",
      ...
    }
  ]},
  "fan": [
    {
      "href": "/api/hardware/v1/chassis/chassis-001/fan/fan-000",
      ...
    },
    ... {
      "href": "/api/hardware/v1/chassis/chassis-001/fan/fan-007",
    },
  ],
  "psu": [
    {
      "href": "/api/hardware/v1/chassis/chassis-001/psu/psu-000",
      ...
    },
    {
      "href": "/api/hardware/v1/chassis/chassis-001/psu/psu-001",
    },
  ],
}
```

```

        "href": "/api/hardware/v1/chassis/chassis-001/psu/psu-002",
    }, {
        "href": "/api/hardware/v1/chassis/chassis-001/psu/psu-003",
    }],
    "slot": [{
        "href": "/api/hardware/v1/chassis/chassis-001/slot/slot-000",
    }, {
        "href": "/api/hardware/v1/chassis/chassis-001/slot/slot-001",
    }],
    }
}

```

하드웨어 구성요소 가져오기

이 명령은 단일 하드웨어 구성요소의 등록 정보를 반환합니다. 성공한 명령에 대해 HTTP 상태 200(OK)이 반환됩니다. 응답 객체는 다음 표에 표시된 구성요소 등록 정보를 포함합니다.

offline, readytoremove 및 use 등록 정보는 풀의 디스크에만 적용됩니다.

등록 정보	유형	설명
device	string	FRU 장치 ID
faulted	boolean	FRU에 결함이 발생했는지 여부
interface	string	FRU 인터페이스 유형
label	string	FRU 위치 레이블
locate	boolean	FRU 위치 LED가 켜져 있는지 여부
manufacturer	string	FRU 제조업체
model	string	FRU 모델
offline	boolean	디스크가 오프라인인지 여부
pathcount	integer	Disk Shelf에 대한 총 I/O 경로 수입니다
present	boolean	FRU 존재 표시기
readytoremove	boolean	결함 후 디스크 드라이브의 분리가 준비되었는지 여부
revision	string	FRU의 펌웨어 또는 하드웨어 개정
rpm	number	플래터 RPM(디스크 전용)
serial	string	FRU 일련 번호
size	number	FRU 크기(용량)
type	string	구성요소 유형
use	string	구성요소 사용법 열거

요청 예:

```

GET /api/hardware/v1/chassis/chassis-001/disk/disk-011 HTTP/1.1
Host: zfs-storage.example.com:215
Accept: application/json

```

결과 예:

```

HTTP/1.1 200 OK
Content-Type: application/json

{
  "disk": {
    "device": "c0t5000CCA01A764FB0d0",
    "faulted": false,
    "href": "/api/hardware/v1/chassis/chassis-001/disk/disk-011",
    "interface": "SAS",
    "label": "HDD 11",
    "locate": false,
    "offline": false,
    "readytoremove": false,
    "manufacturer": "HITACHI",
    "model": "H7230AS60SUN3.0T",
    "pathcount": 2,
    "present": true,
    "revision": "A310",
    "rpm": 7200,
    "serial": "001210R322ED-----YHJ322ED",
    "size": 3000592982016,
    "type": "data",
    "use": "peer"
  }
}

```

구성요소 등록 정보 수정

PUT 요청은 선택된 하드웨어 구성 요소에 대한 등록 정보를 설정하는 데 사용할 수 있습니다. 성공한 요청은 HTTP 상태 201(Accepted)은 물론 JSON 형식의 구성 요소 등록 정보도 반환합니다.

요청 예:

```

PUT /api/hardware/v1/chassis/chassis-001/disk/disk-011 HTTP/1.1
Host: zfs-storage.example.com:215
X-Auth-User: root
X-Auth-Key: password
Accept: application/json
Content-Type: application/json
Content-Length: 16

```

```
{"locate": true}
```

JSON 응답 예:

```

HTTP/1.1 202 Accepted
X-Zfssa-Appliance-API: 1.0
Content-Length: 403
Content-Type: application/json

{
  "disk": {
    "href": "/api/hardware/v1/chassis/chassis-001/disk/disk-011",
    ...
    "locate": true
  }
}

```

```
} }
```

로그 명령

로그 명령은 CLI "유지 관리 로그" 메뉴 아래에서 사용할 수 있는 로그를 관리합니다. 개별 서비스 로그 정보는 서비스 API를 참조하십시오.

로그 관리 명령

다음 표에서는 로그 관리 명령 호출 방법을 보여줍니다.

표 15 로그 관리 명령

요청	경로 <code>/api/log/v{1 2}</code> 에 추가	설명
GET	<code>/api/log/v{1 2}</code> 만 사용	로그 서비스 명령을 나열합니다.
GET	<code>/logs</code>	모든 로그 유형을 나열합니다.
GET	<code>/logs/?start=index/time&limit=entry limit</code>	선택된 범위에 대한 로그 항목을 가져옵니다.
GET	<code>/logs/alert</code>	모든 경고 로그를 나열합니다.
GET	<code>/logs/alert?start=index/time&limit=entry limit</code>	선택된 범위에 대한 로그 항목을 가져옵니다.
GET	<code>/collect</code>	모든 로그 항목의 모음을 다운로드합니다.
GET	<code>/collect?start=index/time&limit=entry limit</code>	선택된 범위에서 로그 항목 모음을 다운로드합니다.

로그 나열

이 명령은 어플라이언스에서 사용할 수 있는 모든 로그를 나열합니다. 각 로그는 로그의 항목 수 및 마지막 항목의 시간 기록을 반환합니다.

주 - depth 질의 매개변수와 match_property-name=value 질의 매개변수는 지원되지 않습니다.

요청 예:

```
GET /api/log/v1/logs HTTP/1.1
Authorization: Basic Tm8gcGV1a2luZyE=
Host: zfs-storage.example.com:215
Accept: application/json
```

결과 예:

```

HTTP/1.1 200 OK
X-Zfssa-Appliance-API: 1.0
Content-Type: application/json
Content-Length: 532
X-Zfssa-API: 1.0

{
  "logs": [
    {
      "href": "/api/log/v1/logs/fault",
      "name": "faults",
      "size": 16,
      "updated": "20130614T22:51:48"
    },
    {
      "href": "/api/log/v1/logs/audit",
      "name": "audits",
      "size": 460149,
      "updated": "20130730T22:10:41"
    },
    {
      "href": "/api/log/v1/logs/alert",
      "name": "alerts",
      "size": 13054,
      "updated": "20130728T00:06:10"
    },
    {
      "href": "/api/log/v1/logs/phone-home",
      "name": "phone-home",
      "size": 249,
      "updated": "20130730T03:22:35"
    },
    {
      "href": "/api/log/v1/logs/system",
      "name": "system",
      "size": 344,
      "updated": "20130724T03:21:55"
    }
  ]
}

```

로그 항목 가져오기

지정된 어플라이언스 로그에서 로그 항목을 반환할 수 있습니다. 각 로그 항목은 로그별 내용 등록 정보와 함께 항목의 날짜/시간을 반환합니다.

주 - 로그 수에 따라 메모리 제약 조건으로 인해 이제 이전 로그 항목을 사용할 수 없게 될 수도 있습니다. 이 동일한 제한은 BUI 및 CLI에서도 적용됩니다. 시스템 로그를 모두 가져오려면 ["로그 관리 명령" \[87\]](#)에 설명된 collect 함수를 사용하십시오.

매개변수	설명
start= <i>index</i>	지정된 인덱스/시간에서 로그 반환을 시작합니다.

매개변수	설명
limit=number	반환된 로그 항목 수를 제한합니다.

시작 인덱스는 생성된 첫번째 로그를 반환하는 값 0으로 기본 설정됩니다. 음수 값 및 로그 크기보다 크거나 같은 값은 허용되지 않습니다. 또한 시작 인덱스는 시간 문자열일 수 있습니다 (예: 20130724T03:21:55).

주 - REST에서는 UTC 시간만 허용됩니다. 현재 시간에서 1개월보다 오래된 시간 값은 허용되지 않습니다. 이전 로그 검색은 시작 값에 대한 인덱스 번호를 사용해야 합니다. 제한 값은 지정된 요청에 대해 반환되는 로그 수를 제한합니다. 지정된 제한 값을 초과하지 않는 값이 반환됩니다.

요청 예:

```
GET /api/log/v1/logs/audit?limit=4&start=1000 HTTP/1.1
Authorization: Basic Tm8gcGVla2luZyE=
Host: zfs-storage.example.com:215
Accept: application/json
```

결과 예:

```
HTTP/1.1 200 OK
X-Zfssa-Appliance-API: 1.0
Content-Type: application/json
X-Zfssa-API: development
Transfer-Encoding: chunked

{
  "logs": [
    {
      "address": "192.0.2.0",
      "annotation": "",
      "summary": "User logged in",
      "timestamp": "20131022T22:54:19",
      "user": "root"
    }, {
      "address": "192.0.2.0",
      "annotation": "",
      "summary": "Destroyed share \"zfs-storage-1:tst.volumes.py.34111.project/tst.volumes.py.34111.lun.7\"",
      "timestamp": "20131022T22:52:34",
      "user": "root"
    }, {
      "summary": "Joined workgroup \"RESTTESTWG\"",
      "timestamp": "20131022T22:54:23",
      "user": "<system>"
    }, {
      "address": "192.0.2.0",
      "annotation": "",
      "summary": "User logged in",
      "timestamp": "20131022T22:54:19",
      "user": "root"
    }
  ]
}
```

로그 다운로드

로그 다운로드 명령은 모든 시스템 로그를 포함하는 gzip으로 압축된 tar 파일을 반환합니다. 파일 배치 이름이 logs.tar.gz로 설정됩니다. 데이터가 실시간으로 만들어지고 스트리밍되므로 다운로드를 재개할 수 없습니다.

로그 다운로드

하나의 로그 유형만 다운로드하려는 경우 표에서와 같이 해당 이름을 collect 리소스에 추가할 수 있습니다. 로그 텍스트는 클라이언트로 다시 스트리밍됩니다. gzip 압축이 요청되는 경우 텍스트 스트림이 gzip으로 압축됩니다. 기타 압축 유형은 지원되지 않으며 무시됩니다.

네트워크 명령

이 절에서 설명하는 네트워크 명령은 네트워크 주소 및 장치를 보고 네트워크 데이터 링크, 인터페이스 및 경로를 구성하는 데 사용됩니다.

네트워크 구성

네트워크 구성 기능을 사용하면 물리적 네트워크 포트를 통해 링크 통합, VNIC(가상 NIC), VLAN(가상 LAN) 및 IPMP(IP Network Multipathing) 그룹을 비롯하여 다양한 고급 네트워킹 설정을 만들 수 있습니다. 그런 다음 이러한 추상화를 위한 IPv4 및 IPv6 주소를 원하는 만큼 정의하여 시스템의 다양한 데이터 서비스에 연결하는 데 사용할 수 있습니다.

시스템의 네트워크 구성에는 다음 4가지 구성요소가 있습니다.

- **장치** – 물리적 네트워크 연결 또는 IPoIB(IP on InfiniBand) 분할 영역에 해당하는 물리적 네트워크 포트입니다.
- **데이터 링크** – 패킷을 보내고 받기 위한 기본 구성체입니다. 데이터 링크는 장치(물리적 네트워크 포트) 또는 IB 분할 영역과 1:1로 대응할 수 있습니다. 또는 사용자가 다른 장치 및 데이터 링크로 구성된 통합, VLAN 및 VNIC 데이터 링크를 정의할 수도 있습니다.
- **인터페이스** – IP 구성 및 주소 지정에 위한 기본 구성체입니다. 각 IP 인터페이스는 단일 데이터 링크와 연관되거나 다른 인터페이스로 구성된 IPMP 그룹으로 정의됩니다.
- **경로 지정** – 시스템에서 IP 패킷의 방향을 지정하는 방법을 제어하는 IP 경로 지정 구성입니다.

이 모델에서는 네트워크 장치가 사용 가능한 하드웨어를 나타내며, 구성 가능한 설정이 없습니다. 데이터 링크는 계층 2 엔티티이며, 이러한 네트워크 장치에 LACP와 같은 설정을 적용하도록 만들어야 합니다. 인터페이스는 데이터 링크를 통해 제공되는 IP 설정이 포함된 계층 3 엔티티입니다. 이 모델에서는 네트워크 인터페이스 설정이 두 부분(계층 2 설정을 위한 데이터 링크와 계층 3 설정을 위한 인터페이스)으로 구분되었습니다.

네트워크 데이터 링크

네트워크 데이터 링크 명령은 어플라이언스에 대한 데이터 링크 관리를 제공합니다. 데이터 링크 리소스를 나열, 수정, 생성 및 삭제할 수 있습니다.

표 16 네트워크 데이터 링크 명령

요청	경로 /network/v{1 2}에 추가	설명
POST	/datalinks	새 네트워크 데이터 링크를 만듭니다.
GET	/datalinks/datalink	지정된 네트워크 데이터 링크 등록 정보를 가져옵니다.
GET	/datalinks	모든 네트워크 데이터 링크 객체를 나열합니다.
PUT	/datalinks/datalink	지정된 네트워크 데이터 링크 객체를 수정합니다.
DELETE	/datalinks/datalink	지정된 데이터 링크 객체를 삭제합니다.

표 17 물리적 장치 데이터 링크 등록 정보

등록 정보	유형	설명
class	String	"device"("immutable")
label	NetworkLabel	레이블
links	ChooseOne	링크 ["igb1", "igb0", "ixgbe2", "ixgbe3", "igb4", "igb3", "ixgbe1", "igb2", "igb5"]
jumbo	부울	점보 프레임 사용 ["true", "false"]("deprecated")
mtu	PositiveInteger	최대 전송 단위(MTU)
speed	ChooseOne	링크 속도 ["auto", "10", "100", "1000", "10000"]
duplex	ChooseOne	링크 이중 ["auto", "half", "full"]

표 18 VNIC 장치 데이터 링크 등록 정보

등록 정보	유형	설명
class	String	"vnic" ("immutable")
label	NetworkLabel	레이블
links	ChooseOne	링크 ["ixgbe0"]
mtu	PositiveInteger	최대 전송 단위(MTU)
id	VLAN	VLAN ID

표 19 VLAN 장치 데이터 링크 등록 정보

등록 정보	유형	설명
class	String	"vlan"("immutable")
label	NetworkLabel	레이블
links	ChooseOne	링크 ["ixgbe0"]
mtu	PositiveInteger	최대 전송 단위(MTU)
id	VLAN	VLAN ID

표 20 통합 기반 장치 데이터 링크 등록 정보

등록 정보	유형	설명
class	String	"aggregation"("immutable")
label	NetworkLabel	레이블
links	ChooseN	링크 ["igb1", "igb0", "ixgbe2", "ixgbe3", "igb4", "igb3", "ixgbe1", "igb2", "igb5"]
jumbo	부울	정보 프레임 사용 ["true", "false"]("deprecated")
mtu	PositiveInteger	최대 전송 단위(MTU)
policy	ChooseOne	정책 ["L2", "L3", "L4", "L2+L3", "L2+L4", "L3+L4"]
mode	ChooseOne	모드 ["active", "passive", "off"]
timer	ChooseOne	타이머 ["short", "long"]
key	Integer	통합 키("immutable")

표 21 IP 분할 영역 기반 장치 데이터 링크 등록 정보

등록 정보	유형	설명
class	String	"partition"("immutable")
label	NetworkLabel	레이블
links	ChooseOne	링크
pkey	Pkey	분할 영역 키
linkmode	ChooseOne	링크 모드 ["cm", "ud"]

네트워크 데이터 링크 나열

어플라이언스의 구성된 모든 데이터 링크를 나열합니다. 데이터 링크 목록의 각 객체는 데이터 링크 등록 정보와 함께 단일 데이터 링크 리소스에 대한 작업을 가져오는 href를 포함합니다.

요청 예:

```
GET /api/network/v1/datalinks HTTP/1.1
Host: zfs-storage.example.com:215
Accept: application/json
```

JSON 데이터 예:

```
{
  "datalinks": [
    {
      "href": "/api/network/v1/datalinks/ixgbe0",
      ...
    },
    {
      "href": "/api/network/v1/datalinks/ixgbe1",
      ...
    },
    {
      "href": "/api/network/v1/datalinks/ixgbe2",
      ...
    },
    {
      "href": "/api/network/v1/datalinks/ixgbe3",
      ...
    }
  ]
}
```

```
    ...
  }
}
```

네트워크 데이터 링크 가져오기

GET 메소드는 데이터 링크 객체 목록과 함께 데이터 링크 등록 정보를 포함하는 JSON 객체를 반환합니다.

```
GET /api/network/v1/datalinks/ixgbe0 HTTP/1.1
Host: zfs-storage.example.com:215
Accept: application/json
```

JSON 데이터 예:

```
{
  "datalink": {
    "class": "device",
    "datalink": "ixgbe0",
    "duplex": "auto",
    "href": "/api/network/v1/datalinks/ixgbe0",
    "jumbo": false,
    "label": "Untitled Datalink",
    "links": [
      "ixgbe0"
    ],
    "mac": "0:21:28:a1:d9:68",
    "mtu": 1500,
    "speed": "auto"
  }
}
```

네트워크 데이터 링크 만들기

POST 명령은 새 데이터 링크를 만듭니다. 새 데이터 링크를 만들 때 필요한 하나의 추가 등록 정보는 만들 데이터 링크의 클래스를 정의하는 `class` 등록 정보입니다. 데이터 링크 클래스는 데이터 링크를 만드는 동안 정의되고 다음 클래스 유형 중 하나일 수 있습니다.

- `device` – 장치 기반 데이터 링크를 만듭니다.
- `vnic` – VNIC 기반 데이터 링크를 만듭니다.
- `vlan` – VLAN 기반 데이터 링크를 만듭니다.
- `aggregation` – 통합 기반 데이터 링크를 만듭니다.
- `partition` – IB 분할 영역 데이터 링크를 만듭니다.

등록 정보는 "구성 네트 데이터 링크" 메뉴에서 사용할 수 있는 동일한 CLI 등록 정보에 매핑됩니다.

요청 예:

```
POST /api/network/v1/datalinks HTTP/1.1
Host: zfs-storage.example.com:215
X-Auth-User: root
```

```
X-Auth-Key: password
Content-Type: application/json
Content-Length: 78
```

```
{
  "class": "device",
  "jumbo": true,
  "links": ["ixgbe2"],
  "label": "TestDataLink"
}
```

결과 예:

```
HTTP/1.1 201 Created
X-Zfssa-Appliance-API: 1.0
Location: /api/network/v1/datalinks/ixgbe2
```

네트워크 데이터 링크 수정

PUT 메소드는 데이터 링크 등록 정보를 수정하는 데 사용됩니다. 데이터 링크 설정에 대한 자세한 내용은 CLI 설명서를 참조하십시오.

요청 예:

```
PUT /api/network/v1/datalinks/ixgbe2 HTTP/1.1
```

```
{"jumbo": true}
```

결과 예:

```
HTTP/1.1 202 Accepted
X-Zfssa-Appliance-API: 1.0
Content-Type: application/json
Content-Length: 219
```

```
{
  "datalink": {
    "href": "/api/network/v1/datalinks/ixgbe2",
    "class": "device",
    "label": "MyDataLink",
    "links": ["ixgbe2"],
    "mac": "0:21:28:a1:d9:6a",
    "mtu": 9000,
    "duplex": "auto",
    "jumbo": true,
    "speed": "auto"
  }
}
```

네트워크 데이터 링크 삭제

이 명령은 시스템에서 데이터 링크를 제거합니다. href 경로를 사용하여 지정된 데이터 링크를 삭제합니다.

요청 예:

DELETE /api/network/v1/datalinks/ixgbe2 HTTP/1.1

결과 예:

HTTP/1.1 204 No Content

네트워크 장치

해당 명령은 시스템의 물리적 네트워크 장치를 나열합니다. 물리적 네트워크 장치에 수정할 수 있는 등록 정보가 없습니다.

표 22 네트워크 장치 명령

요청	경로 /network/v{1 2}에 추가	설명
GET	/devices/device	지정된 네트워크 장치 등록 정보를 가져옵니다.
GET	/devices	모든 네트워크 장치 객체를 나열합니다.

표 23 네트워크 장치 등록 정보

등록 정보	설명
active	장치가 활성 상태인지 여부를 나타내는 부울 플래그
duplex	장치의 이중
factory_mac	공장 MAC 주소
media	장치 매체
speed	장치 속도(Mb/초)
up	장치가 작동하는지 여부를 나타내는 부울 플래그

네트워크 장치 나열

이 명령은 모든 네트워크 장치를 나열합니다.

요청 예:

```
GET /api/network/v1/devices HTTP/1.1
Authorization: Basic Tm8gcGVla2luZyE=
Host: zfs-storage.example.com:215
Accept: application/json
```

결과 예:

```
HTTP/1.1 200 OK
Content-Type: application/json
Content-Length: 412
X-Zfssa-Gns-API: 1.0

{
```

```

    "devices": [{
      "href": "/api/network/v1/devices/ixgbe0",
      ...
    }, {
      "href": "/api/network/v1/devices/ixgbe1",
      ...
    }, {
      "href": "/api/network/v1/devices/ixgbe2",
      ...
    }, {
      "href": "/api/network/v1/devices/ixgbe3",
      ...
    }
  ]
}

```

네트워크 장치 가져오기

이 명령은 단일 네트워크 장치에서 등록 정보를 가져옵니다.

요청 예:

```

GET /api/network/v1/devices/ixgbe0 HTTP/1.1
Authorization: Basic Tm8gcGVla2luZyE=
Host: zfs-storage.example.com:215
Accept: application/json

```

결과 예:

```

HTTP/1.1 200 OK
Content-Type: application/json
Content-Length: 178
X-Zfssa-Gns-API: 1.0

{
  "devices": {
    "active": false,
    "device": "ixgbe0",
    "duplex": "full-duplex",
    "factory_mac": "0:21:28:a1:d9:68",
    "href": "/api/network/v1/devices/ixgbe0",
    "media": "Ethernet",
    "speed": "1000 Mbit/s",
    "up": true
  }
}

```

네트워크 인터페이스

표 24 네트워크 인터페이스 명령

요청	경로 <code>/api/network/v{1 2}</code> 에 추가	설명
POST	<code>/interfaces</code>	새 네트워크 인터페이스를 만듭니다.
GET	<code>/interfaces/interface</code>	지정된 네트워크 인터페이스 등록 정보를 가져옵니다.

요청	경로 <code>/api/network/v{1 2}</code> 에 추가	설명
GET	<code>/interfaces</code>	모든 네트워크 인터페이스 객체를 나열합니다.
PUT	<code>/interfaces/interface</code>	지정된 네트워크 인터페이스 객체를 수정합니다.
DELETE	<code>/interfaces/interface</code>	지정된 인터페이스 객체를 삭제합니다.

표 25 네트워크 인터페이스 등록 정보

등록 정보	설명
<code>admin</code>	이 인터페이스에 대한 관리가 허용되는지 여부를 나타내는 플래그
<code>class</code>	클래스 유형("ip", "ipmp")(만든 후 변경할 수 없음)
<code>curaddr</code>	현재 IP 주소(변경할 수 없음)
<code>enable</code>	이 인터페이스를 사용으로 설정할지 여부를 나타내는 플래그
<code>label</code>	인터페이스에 대한 사용자 레이블
<code>links</code>	이 인터페이스에 대한 네트워크 링크 선택
<code>state</code>	인터페이스의 상태(변경할 수 없음)
<code>v4addr</code>	IPv4 주소
<code>v6dhcp</code>	IPv4 DHCP 플래그
<code>v6addr</code>	IPv6 주소
<code>v6dhcp</code>	IPv6 DHCP 플래그

네트워크 인터페이스 나열

이 명령은 구성된 모든 네트워크 인터페이스를 나열합니다.

요청 예:

```
GET /api/network/v1/interfaces HTTP/1.1
Authorization: Basic Tm8gcGVla2luZyE=
Host: zfs-storage.example.com:215
Accept: application/json
```

결과 예:

```
HTTP/1.1 200 OK
Content-Type: application/json
Content-Length: 369
```

```
{
  "interfaces": [
    {
      "href": "/api/network/v1/interfaces/ixgbe0",
      "v4addr": ["ipaddr-1"]
    },
    {
      "href": "/api/network/v1/interfaces/ixgbe1",
      "v4addr": ["ipaddr-2"]
    }
  ]
}
```

```

        "href": "/api/network/v1/interfaces/ixgbe2",
        "v4addr": ["ipaddr-3"]
      }, {
        "href": "/api/network/v1/interfaces/ixgbe3",
        "v4addr": ["ipaddr-4"]
      }
    ]
  }
}

```

네트워크 인터페이스 가져오기

이 명령은 지정된 네트워크 인터페이스에 대한 등록 정보의 전체 목록을 가져옵니다.

요청 예:

```

GET /api/network/v1/interfaces/ixgbe0 HTTP/1.1
Authorization: Basic Tm8gcGVla2luZyE=
Host: zfs-storage.example.com:215
Accept: application/json

```

결과 예:

```

HTTP/1.1 200 OK
Content-Type: application/json
Content-Length: 292

{
  "interface": {
    "admin": true,
    "class": "ip",
    "curaddr": ["ipaddr-1"],
    "enable": true,
    "href": "/api/network/v1/interfaces/ixgbe0",
    "interface": "ixgbe0",
    "label": "Untitled Interface",
    "links": ["ixgbe0"],
    "state": "up",
    "v4addr": ["ipaddr-1"],
    "v4dhcp": false,
    "v6addr": [],
    "v6dhcp": false
  }
}

```

네트워크 인터페이스 만들기

이 명령은 새 네트워크 인터페이스를 만듭니다.

요청 예:

```

POST /api/network/v1/interfaces HTTP/1.1
Host: zfs-storage.example.com:215
X-Auth-User: root
X-Auth-Key: password
Content-Type: application/json

```

Content-Length: 78

```
{
  "class": "ip",
  "links": ["ixgbe3"],
  "v4addrs": "192.0.2.0/24"
}
```

결과 예:

HTTP/1.1 201 Created
X-Zfssa-Appliance-API: 1.0
Location: /api/network/v1/interfaces/ixgbe3

네트워크 인터페이스 수정

이 명령은 기존 네트워크 인터페이스를 수정합니다.

요청 예:

PUT /api/network/v1/interfaces/ixgbe3 HTTP/1.1

```
{
  "v4addrs": ["192.0.2.0/24"],
  "interface": "Demo Rest"
}
```

결과 예:

HTTP/1.1 202 Accepted
X-Zfssa-Appliance-API: 1.0
Content-Type: application/json
Content-Length: 219

```
{
  "admin": true,
  "class": "ip",
  "curaddrs": ["192.0.2.0/24"],
  "enable": true,
  "href": "/api/network/v1/interfaces/ixgbe3",
  "interface": "ixgbe3",
  "label": "Demo Rest",
  "links": ["ixgbe3"],
  "state": "failed",
  "v4addrs": ["192.0.2.0/24"]
  "v4dhcp": false,
  "v6addrs": [],
  "v6dhcp": false
}
```

네트워크 인터페이스 삭제

이 명령은 기존 네트워크 인터페이스를 삭제합니다.

주 - 인터페이스가 삭제되면 이와 연관된 모든 경로도 제거됩니다.

요청 예:

```
DELETE /api/network/v1/interfaces/ixgbe3 HTTP/1.1
Authorization: Basic Tm8gcGVla2luZyE=
Host: zfs-storage.example.com:215
```

결과 예:

```
HTTP/1.1 204 No Content
```

네트워크 경로

해당 명령은 네트워크 경로를 관리합니다.

표 26 네트워크 경로 관리

요청	경로 <code>/api/network/v{1 2}</code> 에 추가	설명
POST	<code>/routes</code>	새 네트워크 경로를 만듭니다.
GET	<code>/routes/route</code>	지정된 네트워크 경로 등록 정보를 가져옵니다.
GET	<code>/routes</code>	모든 네트워크 경로 객체를 나열합니다.
DELETE	<code>/routes/route</code>	지정된 경로 객체를 삭제합니다.
GET	<code>/routing</code>	네트 경로 지정 등록 정보를 가져옵니다.
PUT	<code>/routing</code>	네트 경로 지정 등록 정보를 수정합니다.

표 27 네트워크 경로 등록 정보 관리

등록 정보	설명
<code>type</code>	"system" 또는 "static"과 같은 경로의 유형(변경할 수 없음)
<code>family</code>	주소 그룹(IPv4 또는 IPv6)
<code>destination</code>	경로 대상 주소
<code>gateway</code>	게이트웨이 주소
<code>interface</code>	네트워크 데이터 링크 인터페이스

각 경로에 대한 href 경로는 CLI에 설정된 경로 ID를 사용하지만 이러한 값은 경로가 수정될 때 변경될 수 있습니다. API는 경로 내 고유한 등록 정보를 사용하여 단일 경로를 선택합니다. 구문은 `routes/route-###`과 비교하여 `routes/name=value`입니다.

경로 나열

어플라이언스에서 만든 모든 네트워크 경로를 나열합니다.

요청 예:

```
GET /api/network/v1/routes HTTP/1.1
Authorization: Basic Tm8gcGVla2luZyE=
Host: zfs-storage.example.com:215
Accept: application/json
```

결과 예:

```
HTTP/1.1 200 OK
Content-Type: application/json
Content-Length: 192
```

```
{
  "routes": [{
    "destination": "ipaddr-0",
    "family": "IPv4",
    "gateway": "ipaddr-1",
    "href": "/api/network/v1/routing/route-000",
    "interface": "ixgbe0",
    "mask": 0,
    "route": "route-000",
    "type": "static"
  }, {
    "destination": "ipaddr-2",
    "family": "IPv4",
    "gateway": "ipaddr-3",
    "href": "/api/network/v1/routes/route-001",
    "interface": "ixgbe0",
    "mask": 24,
    "route": "route-001",
    "type": "system"
  }
]}
```

경로 가져오기

단일 경로에 대한 등록 정보를 가져옵니다.

요청 예:

```
GET /api/network/v1/routes/destination=ipaddr-1 HTTP/1.1
Authorization: Basic Tm8gcGVla2luZyE=
Host: zfs-storage.example.com:215
Accept: application/json
```

결과 예:

```
HTTP/1.1 200 OK
Content-Type: application/json
Content-Length: 192
```

```
{
  "route": {
    "destination": "ipaddr-1",
    "family": "IPv4",
    "gateway": "ipaddr-2",
    "href": "/api/network/v1/routes/route-001",
    "interface": "ixgbe0",
```

```

    "mask": 24,
    "route": "route-001",
    "type": "system"
  }
}

```

경로 추가

새 네트워크 경로를 만듭니다. 경로 href 값은 다른 경로가 시스템에 추가된 경우 변경될 수 있습니다. 반환된 등록 정보가 입력 등록 정보와 동일하므로 만들 때 경로 정보가 반환되지 않습니다. 성공한 만들기는 HTTP 상태 204(Created)를 반환합니다.

정적 경로를 만드는 요청의 예:

```

POST /api/network/v1/routes HTTP/1.1
Authorization: Basic Tm8gcGV1a2luZyE=
Host: zfs-storage.example.com:215
Content-Type: application/json
Content-Length: 164

```

```

{
  "family": "IPv4",
  "destination": "ipaddr-0",
  "mask": "0",
  "gateway": "ipaddr-1",
  "interface": "ixgbe0"
}

```

결과 예:

```

HTTP/1.1 201 Created

```

경로 삭제

기존 네트워크 경로를 삭제합니다.

요청 예:

```

DELETE /api/network/v1/routes/route-001 HTTP/1.1
Authorization: Basic Tm8gcGV1a2luZyE=
Host: zfs-storage.example.com:215

```

결과 예:

```

HTTP/1.1 204 No Content

```


RESTful API 클라우드 서비스

RESTful API 클라우드 서비스를 통해 어플라이언스 관리자는 다음 작업을 수행할 수 있습니다.

- 어플라이언스의 ZFS 스냅샷 데이터를 Oracle Cloud Infrastructure 객체 스토리지에 백업합니다.
- 클라우드에서 사용 가능한 스냅샷 백업을 나열합니다.
- 스냅샷 백업을 새 공유로 어플라이언스에 복원합니다.
- 스냅샷 백업 데이터가 더 이상 필요하지 않은 경우 삭제합니다.

둘 이상의 어플라이언스를 동일한 클라우드 대상에 백업할 수 있습니다. 클라우드 백업은 클라우드 대상에 대한 액세스 권한이 있는 모든 어플라이언스에 새로 복제된 공유로 복원할 수 있습니다.

다음 항목을 비롯하여 클라우드 백업에 대한 자세한 내용은 [“Configuring Cloud Backup” in Oracle ZFS Storage Appliance Administration Guide, Release OS8.8.x](#)을 참조하십시오.

- "표준" 또는 "아카이브" 계층을 비롯한 Oracle Cloud Infrastructure 계정 구성
- 데이터 버킷, 메타데이터 버킷 및 아카이브 버킷 관리
- 클라우드 백업 작업에 대한 클러스터화된 구성 영향

다음 작업을 포함하여 Oracle Cloud Infrastructure 객체 스토리지 서비스에 대한 자세한 내용은 [“Oracle ZFS Storage Appliance RESTful API Support for the Oracle Cloud Infrastructure Object Storage Service” in Oracle ZFS Storage Appliance Object API Guide for Oracle Cloud Infrastructure Object Storage Service Support, Release OS8.8.x](#)를 참조하십시오.

- Oracle Cloud Infrastructure 서비스를 사용으로 설정합니다.
- 사용자 키를 만들거나 삭제합니다. 사용자 키의 작업 권한을 변경합니다.

클라우드 서비스 작업

클라우드 서비스 작업을 사용하여 Oracle Cloud Infrastructure 객체 스토리지에서 어플라이언스 스냅샷 백업을 관리할 수 있습니다. Oracle Cloud Infrastructure 객체 스토리지의 대상

및 백업을 나열하고, 대상을 제거하고, 백업을 삭제하고, 백업을 새 어플라이언스 공유로 복원하고, 클라우드 서비스 작업을 취소 또는 다시 시작할 수 있습니다.

백업을 만들려면 “스냅샷 백업 만들기” [119] in “스냅샷 백업 작업” [118]를 참조하십시오. 백업은 공유 스냅샷의 전체 또는 증분 백업입니다. 스냅샷을 만들려면 “스냅샷 및 복제 작업” [174]을 참조하십시오.

클라우드 서비스 작업 표에서는 다음 매개변수가 사용됩니다.

<i>format</i>	Oracle Cloud Infrastructure 객체 스토리지에서 백업이 저장된 형식입니다. <i>format</i> 의 값은 <i>zfs</i> 또는 <i>tar</i> 입니다. 스냅샷 백업 생성에 대한 <i>format</i> 이 지정되지 않은 경우 <i>zfs</i> 가 기본값입니다. <i>zfs</i> 형식은 파일 시스템과 LUN 스냅샷을 지원하며, <i>tar</i> 형식은 파일 시스템 스냅샷만 지원합니다. <i>zfs</i> 및 <i>tar</i> 형식에 대한 자세한 내용은 “ Creating a Cloud Backup (CLI) in Oracle ZFS Storage Appliance Administration Guide, Release OS8.8.x ”를 참조하십시오.
<i>backup-id</i>	Oracle Cloud Infrastructure 객체 스토리지에 저장된 어플라이언스 스냅샷의 복사본에 대한 식별자입니다. “ 클라우드 백업 나열 ” [112]에서와 같이 백업을 나열할 때 <i>backup-id</i> 의 값은 <i>id</i> 등록 정보의 값입니다. 전체 스냅샷 백업의 경우 <i>backup-id</i> 의 값은 다음 예에서와 같이 <i>pool_id/snapshot_id</i> 입니다. 3e035b7e546e0d02/1cbfdb5ff2259b76 증분 스냅샷 백업의 경우 <i>backup-id</i> 의 값은 다음 예에서와 같이 <i>pool_id/child_snapshot_id-parent_snapshot_id</i> 입니다. 6913a5703bee98dc/46be95ced54e99d9-667f3eb88fd209e1
<i>target-id</i>	백업이 저장되는 Oracle Cloud Infrastructure 객체 스토리지 위치입니다. 지정된 백업은 여러 대상에 저장할 수 있습니다. 즉, 동일한 <i>backup-id</i> 가 다른 <i>target-id</i> 위치에 나타날 수 있습니다. 백업을 나열할 때 <i>target-id</i> 의 값은 <i>target</i> 등록 정보의 값입니다.
<i>job-id</i>	실행 중인 작업의 식별자입니다. “ 작업 나열 ” [116]에서와 같이 작업을 나열할 때 <i>job-id</i> 의 값은 <i>id</i> 등록 정보의 값입니다.

표 28 클라우드 서비스 명령

요청	경로 <i>/api/service/v2/services</i> 에 추가	설명
GET	<i>/cloud</i>	대상, 백업 및 작업에 대한 등록 정보와 요약 데이터를 나열합니다.
PUT	<i>/cloud</i>	등록 정보를 수정합니다.
POST	<i>/cloud/targets</i>	새 대상을 만듭니다.

요청	경로 /api/service/v2/services에 추가	설명
GET	/cloud/targets	대상을 나열합니다.
GET	/cloud/targets/target-id	지정된 대상의 등록 정보를 나열합니다.
PUT	/cloud/targets/target-id	지정된 대상의 등록 정보를 수정합니다.
DELETE	/cloud/targets/target-id	서비스에서 지정된 대상을 제거합니다.
GET	/cloud/backups	모든 형식의 완료된 백업을 모두 나열합니다.
GET	/cloud/backups/format/backup-id/target-id	지정된 백업을 나열합니다.
DELETE	/cloud/backups/format/backup-id/target-id	지정된 백업을 삭제합니다(작업 요청을 제출함).
POST	/cloud/backups/format/backup-id/target-id/restore	지정된 백업을 복원합니다(작업 요청을 제출함).
GET	/cloud/jobs	실행 중인 작업 및 최근에 완료된 작업을 나열합니다.
GET	/cloud/jobs/job-id	지정된 작업의 등록 정보를 나열합니다.
PUT	/cloud/jobs/job-id/cancel	지정된 실행 중인 작업을 취소합니다.
PUT	/cloud/jobs/job-id/restart	지정된 중단된 작업을 다시 시작합니다.

클라우드 서비스 사용

클라우드 서비스를 사용으로 설정하려면 다음 예에서와 같이 `status`를 `enable`로 설정합니다.

요청 예:

```
PUT /api/service/v2/services/cloud HTTP/1.1
Host: hostname:215
Authorization: Basic Tm8gcGV1a2luZyE=
Accept: */*
Content-Type: application/json
Content-Length: 22

{ "<status>": "enable" }
```

클라우드 서비스 로그 파일 보기

다음 요청을 사용하여 클라우드 서비스에 대한 로그 파일을 볼 수 있습니다.

```
GET /api/log/v1/logs/appliance-kit-cloud:default HTTP/1.1
```

클라우드 서비스 등록 정보 나열

요청 예:

```
GET /api/service/v2/services/cloud HTTP/1.1
```

```
Host: hostname:215
Authorization: Basic Tm8gcGVla2luZyE=
Accept: */*
```

결과 예:

```
HTTP/1.1 200 OK
Date: Wed, 24 Jul 2019 20:30:59 GMT
Content-Length: 843
X-Zfssa-Service-API: 2.0
X-Zfssa-API-Version: 2.0
Content-Type: application/json; charset=utf-8
```

```
{
  "service": {
    "href": "/api/service/v2/services/cloud",
    "<status>": "online",
    "tls_version": [
      "TLSv1.1",
      "TLSv1.2"
    ],
    "ciphers": [
      "ECDHE-RSA-AES128-GCM-SHA256",
      "ECDHE-RSA-AES256-GCM-SHA384",
      "DHE-RSA-AES128-GCM-SHA256",
      "DHE-RSA-AES256-GCM-SHA384",
      "AES128-SHA",
      "AES256-SHA",
      "DES-CBC3-SHA"
    ],
    "targets": {
      "href": "/api/service/v2/services/cloud/targets",
      "entries": 2
    },
    "backups": {
      "href": "/api/service/v2/services/cloud/backups",
      "entries": 2548
    },
    "jobs": {
      "href": "/api/service/v2/services/cloud/jobs",
      "entries": 0
    }
  }
}
```

클라우드 서비스 등록 정보 수정

요청 예:

```
PUT /api/service/v2/services/cloud HTTP/1.1
Host: hostname:215
Authorization: Basic Tm8gcGVla2luZyE=
Accept: */*
Content-Type: application/json
Content-Length: 48
```

```
{
  "tls_version": [
    "TLSv1.0", "TLSv1.1", "TLSv1.2"
  ]
}
```

```
}

```

대상 나열

다음 예에서는 모든 대상을 나열합니다.

요청 예:

```
GET /api/service/v2/services/cloud/targets HTTP/1.1
Host: hostname:215
Authorization: Basic Tm8gcGV1a2luZyE=
Accept: */*
```

결과 예:

```
HTTP/1.1 200 OK
Date: Wed, 24 Jul 2019 21:06:18 GMT
Content-Length: 1086
X-Zfssa-Service-API: 2.0
X-Zfssa-API-Version: 2.0
Content-Type: application/json; charset=utf-8

{
  "targets": [
    {
      "bucket": "pl-test",
      "href": "/api/service/v2/services/cloud/targets/target-id1",
      "id": "target-id1",
      "key": true,
      "location": "https://objectstorage.us-ashburn-1.oraclecloud.com",
      "name": "oci-ashburn",
      "online": true,
      "proxy_host": "",
      "proxy_on": false,
      "proxy_password": false,
      "proxy_user": "",
      "tenancy": "ocid1.tenancy.oc1..tenancy-id",
      "user": "ocid1.user.oc1..user-id"
    },
    {
      "bucket": "pl-test2",
      "href": "/api/service/v2/services/cloud/targets/target-id2",
      "id": "target-id2",
      "key": true,
      "location": "https://objectstorage.us-phoenix-1.oraclecloud.com",
      "name": "oci-phoenix",
      "online": true,
      "proxy_host": "www-proxy.example.com:80",
      "proxy_on": true,
      "proxy_password": false,
      "proxy_user": "",
      "tenancy": "ocid1.tenancy.oc1..tenancy-id",
      "user": "ocid1.user.oc1..user-id"
    }
  ]
}
```

다음 예에서는 지정된 대상을 나열합니다.

```
GET /api/service/v2/services/cloud/targets/target-id HTTP/1.1
Host: hostname:215
Authorization: Basic Tm8gcGV1a2luZyE=
Accept: */*
```

대상 만들기

클라우드 대상을 만들려면 다음 매개변수가 필요합니다.

매개변수	설명
<i>key</i>	<i>user-id</i> 에 대한 Oracle Cloud Infrastructure 계정 사용자 키
<i>tenancy-id</i>	Oracle Cloud Infrastructure 계정 테넌시 이름 OCID
<i>user-id</i>	Oracle Cloud Infrastructure 계정 사용자 이름 OCID

다음 예에서는 대상을 만듭니다.

요청 예:

```
POST /api/service/v2/services/cloud/targets HTTP/1.1
Host: hostname:215
Authorization: Basic Tm8gcGVla2luZyE=
```

```
{
  "name": "oci4",
  "location": "https://objectstorage.us-phoenix-1.oraclecloud.com",
  "user": "ocid1.user.oc1..user-id",
  "bucket": "test-bucket3",
  "tenancy": "ocid1.tenancy.oc1..tenancy-id",
  "key": "key",
  "proxy_on": false,
  "readlimit": -1,
  "writelimit": -1
}
```

결과 예:

```
HTTP/1.1 201 Created
Date: Wed, 24 Jul 2019 21:14:39 GMT
Content-Length: 568
X-Zfssa-Service-API: 2.0
Location: /api/service/v2/services/cloud/targets/target-id
X-Zfssa-API-Version: 2.0
Content-Type: application/json; charset=utf-8
```

```
{
  "target": {
    "bucket": "test-bucket3",
    "href": "/api/service/v2/services/cloud/targets/target-id",
    "id": "target-id",
    "key": true,
    "location": "https://objectstorage.us-phoenix-1.oraclecloud.com",
    "name": "oci4",
    "proxy_on": false,
    "readlimit": -1,
    "state": "offline",
    "tenancy": "ocid1.tenancy.oc1..tenancy-id",
    "user": "ocid1.user.oc1..user-id",
    "writelimit": -1
  }
}
```

대상 수정

다음 표에서는 클라우드 대상에 대해 수정 가능한 등록 정보를 보여줍니다.

등록 정보	설명
name	각 어플라이언스에서 고유해야 할 이 클라우드 대상의 이름입니다.
proxy_on	true인 경우 인터넷과의 시스템 통신에 프록시를 사용합니다. proxy_on의 값이 true인 경우 proxy_host에 대한 값을 제공해야 합니다.
proxy_host	프록시 호스트 이름 및 포트 번호입니다.
proxy_password	선택사항입니다. 프록시 암호입니다.
proxy_user	선택사항입니다. 프록시 사용자 이름입니다.
readlimit	선택사항입니다. 클라우드 대상에서 데이터를 읽을 초당 최대 바이트 속도입니다. 클라우드 백업이 클라우드에서 복원될 때 이 값이 사용됩니다. 예를 들어, 값이 4194304인 경우 클라우드 대상에서 읽기 속도가 초당 4MB로 제한됩니다. 값이 -1이면 I/O가 제한되지 않습니다.
writelimit	선택사항입니다. 클라우드 대상에 데이터를 쓸 초당 최대 바이트 속도입니다. 클라우드 �냅샷이 클라우드 대상으로 업로드될 때 이 값이 사용됩니다. 예를 들어, 값이 5242880인 경우 클라우드 대상에 쓰기 속도가 초당 5MB로 제한됩니다. 값이 -1이면 I/O가 제한되지 않습니다.

요청 예:

```
PUT /api/service/v2/services/cloud/targets/target-id HTTP/1.1
Host: hostname:215
Authorization: Basic Tm8gcGV1a2luZyE=
User-Agent: curl/7.54.0
Accept: */*
Content-Type: application/json
Content-Length: 19
{
  "proxy_on": true,
  "proxy_host": "www-proxy.example.com:80",
  "readlimit": 4194304,
  "writelimit": 5242880
}
```

대상 삭제

대상을 삭제하기 전에 다음을 확인합니다.

- 이 대상에 대한 백업이 진행 중인지 여부. “[작업 나열](#)” [116]을 참조하십시오.
- 이 대상에 백업이 있는지 여부. “[클라우드 백업 나열](#)” [112]에서와 같이 target 필터를 사용하여 이 대상에 저장된 백업을 나열합니다.

다음 예에서는 지정된 클라우드 대상을 서비스에서 제거합니다.

요청 예:

```
DELETE /api/service/v2/services/cloud/targets/target-id HTTP/1.1
Host: hostname:215
Authorization: Basic Tm8gcGVla2luZyE=
Accept: */*
```

결과 예:

```
HTTP/1.1 204 No Content
Date: Wed, 24 Jul 2019 21:20:27 GMT
X-Content-Type-Options: nosniff
X-Zfssa-Service-API: 2.0
X-Zfssa-API-Version: 2.0
Content-Type: application/json; charset=utf-8
```

클라우드 백업 나열

다음 질의를 사용하여 모든 형식의 완료된 백업을 모두 나열할 수 있습니다(최신 백업이 먼저 나열됨). 진행 중인 백업에 대한 정보를 가져오려면 ["작업 나열" \[116\]](#)을 참조하십시오.

요청 예:

```
GET /api/service/v2/services/cloud/backups HTTP/1.1
Host: hostname:215
Authorization: Basic Tm8gcGVla2luZyE=
Accept: */*
```

결과 예:

이 예에서 나열된 첫번째 백업은 나열된 두번째 백업의 자식입니다. 첫번째 백업의 `parent` 값은 두번째 백업(`app-data-fullsnap`)의 `dataset` 값과 동일하며, 두번째 백업의 `parent` 값은 널입니다. 자식 백업은 부모와 동일한 `format`이 됩니다. 이 예에서는 자식 백업과 부모 백업이 `tar` 형식입니다.

```
HTTP/1.1 200 OK
Date: Wed, 22 Jan 2020 21:22:40 GMT
X-Zfssa-Service-API: 2.0
X-Zfssa-API-Version: 2.0
Content-Type: application/json; charset=utf-8
Transfer-Encoding: chunked

{
  "backups": [{
    "target": "target-id2",
    "parent": "app-data-fullsnap",
    "started": "2020-01-06T20:03:32Z",
    "completed": "2020-01-06T20:04:36Z",
    "id": "backup-id2",
    "source": "cloudsnap0",
    "href": "/api/service/v2/services/cloud/backups/tar/backup-id2/target-id2",
    "tier": "standard",
    "dataset": "p1/local/default/app-data@app-data-incsnap",
    "format": "tar",
    "size": 3224982536.0
  }, {
    "target": "target-id1",
    "parent": "",
```

```

    "started": "2020-01-06T20:01:16Z",
    "completed": "2020-01-06T20:01:17Z",
    "id": "backup-id1",
    "source": "cloudsnap0",
    "href": "/api/service/v2/services/cloud/backups/tar/backup-id1/target-id1",
    "tier": "standard",
    "dataset": "p1/local/default/app-data@app-data-fullsnap",
    "format": "tar",
    "size": 2149988056.0
  }
}

```

다음 요청을 사용하여 지정된 백업을 나열할 수 있습니다.

```
GET /api/service/v2/services/cloud/backups/format/backup-id/target-id HTTP/1.1
```

클라우드 백업 질의 매개변수 나열

클라우드 백업 목록을 필터링할 수 있도록 다음과 같은 질의 매개변수가 지원됩니다. 모든 매개변수를 동일한 질의에서 사용할 수 있습니다. 표 뒤에 나오는 예에서와 같이 질의 매개변수 사이에 AND 연산(&)을 사용합니다.

표 29 클라우드 백업 나열을 위한 질의 매개변수

등록 정보	설명
start	<p>start의 값은 다음 중 하나입니다.</p> <ul style="list-style-type: none"> 백업 인덱스입니다. 이 인덱스의 값은 0부터 <code>entries</code> 등록 정보(“클라우드 서비스 등록 정보 나열” [107]에 표시되는 요청의 <code>backups</code> 절에 있음)의 값보다 1 작은 수까지입니다. 지정된 인덱스로 선택된 백업 및 지정된 백업이 만들어진 후 만들어진 가장 오래된 백업을 나열합니다. <code>%Y-%m-%dT%H:%M:%SZ</code> 형식의 시간입니다. 이는 <code>started</code> 및 <code>completed</code> 등록 정보의 값에 사용되는 동일한 형식입니다. 이 시간은 현재 시간까지의 모든 실시간일 수 있습니다. 예를 들어, <code>2019-09-00T00:00:00Z</code>는 실시간이 아닙니다. 지정된 시간 또는 그 이후에 만들어진 가장 오래된 백업을 나열합니다.
end	<p>end의 값은 <code>%Y-%m-%dT%H:%M:%SZ</code> 형식의 시간입니다. 이는 <code>started</code> 및 <code>completed</code> 등록 정보의 값에 사용되는 동일한 형식입니다. 이 시간은 현재 시간까지의 모든 실시간일 수 있습니다. 예를 들어, <code>2019-09-00T00:00:00Z</code>는 실시간이 아닙니다.</p> <p>지정된 시간 또는 그 이전에 만들어진 최신 백업을 나열합니다.</p>
limit	지정된 수 이하의 백업을 나열합니다. <code>limit</code> 매개변수에는 기본값이 없습니다.
target	<code>target</code> 등록 정보의 값입니다. 해당 대상의 최신 백업을 나열합니다.
source	<code>source</code> 등록 정보의 값입니다. 해당 소스의 최신 백업을 나열합니다.
dataset	<p>@ 앞에 있는 <code>dataset</code> 등록 정보의 값입니다. 예를 들어, 데이터 세트 <code>p1/local/default/app-data@app-data-incsnap</code>의 경우 <code>p1/local/default/app-data</code> 또는 <code>app-data</code>를 지정합니다.</p> <p>지정된 데이터 세트의 최신 백업을 나열합니다.</p> <p>주 - 데이터 세트 백업에 대한 요청은 이름에 요청된 데이터 세트 이름이 포함된 모든 데이터 세트의 모든 백업을 나열합니다. 예를 들어, <code>p1/local/default/app-data@app-data-incsnap</code>의 백업에 대한 요청은 이름이 <code>app-data-incsnap</code>, <code>app-data-incsnap-1</code> 및 <code>myproj-app-data-incsnap</code>인 데이터 세트의 백업도 반환합니다.</p>
format	<code>format</code> 등록 정보의 값(<code>zfs</code> 또는 <code>tar</code>)입니다. 지정된 백업 형식의 최신 백업을 나열합니다.

다음 예에서는 인덱스 번호가 2000인 백업부터 시작하여 가장 오래된 백업을 나열합니다. 클라우드 서비스 등록 정보 목록의 `backups` 절에서 `entries` 등록 정보의 값이 2865인 경우 다음 예에서는 865개 백업(백업 2000부터 백업 2864까지)을 보여줍니다.

```
GET /api/service/v2/services/cloud/backups?start=2000
```

다음 예에서는 인덱스 번호가 2000인 백업만 나열합니다.

```
GET /api/service/v2/services/cloud/backups?start=2000&limit=1
```

다음 예에서는 지정된 시간 또는 그 이후에 만들어진 가장 오래된 500개의 백업을 나열합니다.

```
GET /api/service/v2/services/cloud/backups?start=2019-07-12T00:00:00Z&limit=500
```

다음 예에서는 지정된 시간 또는 그 이전에 만들어진 최신 500개의 백업을 나열합니다.

```
GET /api/service/v2/services/cloud/backups?end=2019-07-12T00:00:00Z&limit=500
```

다음 예에서는 지정된 `start` 시간 또는 그 이후부터 지정된 `end` 시간 전까지 만들어진 모든 백업을 나열합니다.

```
GET /api/service/v2/services/cloud/backups?start=2019-07-11T00:00:00Z&end=2019-07-12T00:00:00Z
```

다음 예에서는 `tar` 백업 형식으로 이름에 `app-data`가 포함된 모든 데이터 세트의 최신 백업을 나열합니다.

```
GET /api/service/v2/services/cloud/backups?dataset=app-data&format=tar
```

다음 예에서는 `target-id` 대상의 이름에 `app-data`가 있는 모든 데이터 세트의 최신 백업을 나열합니다.

```
GET /api/service/v2/services/cloud/backups?dataset=app-data&target=target-id
```

다음 예에서는 소스가 `cloudsnap0`인 `target-id` 대상의 이름에 `app-data`가 있는 모든 데이터 세트의 최신 백업을 나열합니다.

```
GET /api/service/v2/services/cloud/backups?dataset=app-data&target=target-id&source=cloudsnap0
```

클라우드 백업삭제

클라우드 백업을 삭제하기 전에 다음을 확인합니다.

- 이 백업의 복원이 진행 중인지 여부. “[작업 나열](#)” [116]을 참조하십시오.
- 이 백업에 자식 항목이 있는지 여부. 자식 항목 백업이 있는 백업은 삭제할 수 없습니다. 이 백업의 자식 항목은 삭제할 백업의 `dataset` 값과 동일한 `parent` 값을 가집니다. “[클라우드 백업 나열](#)” [112]을 참조하십시오.

다음 예에서는 지정된 대상의 지정된 백업을 Oracle Cloud Infrastructure 객체 스토리지에서 삭제하려는 작업 요청을 제출합니다. 어플라이언스는 어플라이언스가 액세스할 수 있는 모든

대상의 모든 백업에서 작동할 수 있습니다. 해당 백업이 다른 어플라이언스에서 만들어진 경우에도 마찬가지입니다.

어플라이언스에서 공유 스냅샷 백업을 삭제하는 방법을 보여주는 [“스냅샷 백업 삭제” \[122\]](#)와 이 작업을 비교합니다.

요청 예:

```
DELETE /api/service/v2/services/cloud/backups/format/backup-id/target-id HTTP/1.1
Host: hostname:215
Authorization: Basic Tm8gcGV1a2luZyE=
Accept: */*
```

결과 예:

```
{
  "action": "job-id"
}
```

클라우드 백업 삭제의 진행 상황을 보려면 [“작업 나열” \[116\]](#)을 사용하여 위 *job-id*의 작업을 확인합니다.

클라우드 백업 복원

클라우드 백업은 클라우드 대상에 대한 액세스 권한이 있는 모든 어플라이언스에 새로 복제된 공유로 복원할 수 있습니다.

다음 예에서는 지정된 백업을 복원하려는 작업 요청을 제출합니다. 백업을 복원할 풀 및 프로젝트를 지정하고 새 공유에 대한 이름을 지정합니다. 복원의 진행 상황을 보려면 [“작업 나열” \[116\]](#)을 사용하여 반환된 *job-id*의 작업을 확인합니다.

대상에 대해 `readlimit` 등록 정보가 설정된 경우 대상에서 읽는 초당 바이트가 `readlimit` 이하입니다. [“대상 만들기” \[110\]](#)를 참조하십시오.

요청 예:

```
POST /api/service/v2/services/cloud/backups/format/backup-id/target-id/restore HTTP/1.1
Host: hostname:215
Authorization: Basic Tm8gcGV1a2luZyE=
Accept: */*
Content-Length: 55
```

```
{
  "pool": "p1",
  "project": "default",
  "share": "restore6"
}
```

결과 예:

```
{
  "action": "job-id"
}
```

작업 나열

다음 예에서는 실행 중인 모든 작업 및 최근 완료된 작업을 나열합니다(최신 작업이 먼저 나열 됨).

요청 예:

```
GET /api/service/v2/services/cloud/jobs HTTP/1.1
Host: hostname:215
Authorization: Basic Tm8gcGVla2luZyE=
Accept: */*
```

결과 예:

이 예에서는 두 백업 작업이 동일한 대상에 다른 두 백업을 저장하고 있습니다.

```
HTTP/1.1 200 OK
Date: Wed, 22 Jan 2020 21:37:52 GMT
Content-Length: 983
X-Zfssa-Service-API: 2.0
X-Zfssa-API-Version: 2.0
Content-Type: application/json; charset=utf-8

{
  "jobs": [{
    "href": "/api/service/v2/services/cloud/jobs/job-id2",
    "op": "backup",
    "target": "target-id",
    "targetName": "oci-ashburn",
    "created": "2020-02-06T16:52:42Z",
    "updated": "2020-02-06T16:52:48Z",
    "id": "job-id2",
    "status": "in-progress",
    "rate": 10002432,
    "transferred": 80019456,
    "estimated_size": 43088792088,
    "dataset": "p1/local/default/f-1",
    "backup": "backup-id2",
    "snapshot": "snap3",
    "format": "tar",
    "details": "uploading backup to zfs/backups/tar/backup-id2/000000001"
  }, {
    "href": "/api/service/v2/services/cloud/jobs/job-id1",
    "op": "backup",
    "target": "target-id",
    "targetName": "oci-ashburn",
    "created": "2020-02-06T16:52:28Z",
    "updated": "2020-02-06T16:52:48Z",
    "id": "job-id1",
    "status": "in-progress",
    "rate": 1942,
    "transferred": 3884,
    "estimated_size": 0,
    "dataset": "p1/local/default/f-1",
    "backup": "backup-id1",
    "snapshot": "snap2",
    "format": "zfs",
    "details": "uploading backup to zfs/backups/zfs/backup-id1/000000001"
  }
  ]
}
```

다음 예에서는 지정된 작업을 나열합니다.

```
GET /api/service/v2/services/cloud/jobs/job-id2 HTTP/1.1
Host: hostname:215
Authorization: Basic Tm8gcGV1a2luZyE=
Accept: */*
```

클라우드 백업 작업 질의 매개변수 나열

클라우드 백업 작업 목록 필터링에는 `start` 및 `limit` 질의 매개변수가 지원됩니다.

- `start` 매개변수의 값은 작업 인덱스입니다. 이 인덱스의 값은 0부터 `entries` 등록 정보(“클라우드 서비스 등록 정보 나열” [107]에 표시되는 요청의 `jobs` 절에 있음)의 값보다 1 작은 수까지입니다.
- `limit` 매개변수의 값은 나열할 최대 작업 수입니다.

두 매개변수를 동일한 질의에서 사용할 수 있습니다. 다음 예에서와 같이 질의 매개변수 사이에 AND 연산(&)을 사용합니다.

다음 예에서는 실행 중인 100개의 최신 작업 및 최근 완료된 작업을 나열합니다(최신 작업이 먼저 나열됨).

```
GET /api/service/v2/services/cloud/jobs?limit=100
```

다음 예에서는 실행 중인 가장 오래된 작업 또는 최근에 완료된 작업만 나열합니다.

```
GET /api/service/v2/services/cloud/jobs?start=0&limit=1
```

다음 예에서는 실행 중인 모든 작업 및 작업 인덱스 번호가 4보다 크거나 같은 최근에 완료된 작업을 나열합니다(최신 작업이 먼저 나열됨).

```
GET /api/service/v2/services/cloud/jobs?start=4
```

작업 취소 또는 다시 시작

다음 예에서는 지정된 클라우드 서비스 작업을 취소합니다.

요청 예:

```
PUT /api/service/v2/services/cloud/jobs/job-id/cancel HTTP/1.1
Host: hostname:215
Authorization: Basic Tm8gcGV1a2luZyE=
Accept: */*
```

결과 예:

```
HTTP/1.1 202 Accepted
Date: Wed, 24 Jul 2019 21:50:29 GMT
Content-Length: 0
X-Zfssa-Service-API: 2.0
X-Zfssa-API-Version: 2.0
Content-Type: application/json; charset=utf-8
```

다음 예에서는 지정된 클라우드 서비스 작업을 다시 시작합니다.

요청 예:

```
PUT /api/service/v2/services/cloud/jobs/job-id/restart HTTP/1.1
Host: hostname:215
Authorization: Basic Tm8gcGVla2luZyE=
User-Agent: curl/7.54.0
Accept: */*
```

결과 예:

```
HTTP/1.1 202 Accepted
Date: Wed, 24 Jul 2019 21:51:08 GMT
Content-Length: 0
X-Zfssa-Service-API: 2.0
X-Zfssa-API-Version: 2.0
Content-Type: application/json; charset=utf-8
```

스냅샷 백업 작업

스냅샷 백업 작업을 사용하여 Oracle Cloud Infrastructure 객체 스토리지에서 공유 스냅샷 백업을 관리할 수 있습니다. 스냅샷을 만들려면 [“스냅샷 및 복제 작업” \[174\]](#)을 참조하십시오.

각 스냅샷 백업에는 고유한 식별자가 지정되므로 다른 로컬 시스템에서 동일한 이름을 가진 스냅샷은 동일한 클라우드 대상에 백업할 수 있습니다.

동일한 파일 시스템 스냅샷을 두 가지 다른 형식으로 2개의 클라우드 백업에 사용할 수 있습니다.

로컬 스냅샷은 클라우드에 해당 스냅샷을 백업한 후 삭제할 수 있습니다. 단, 향후 증분 스냅샷을 위해 부모 스냅샷으로 사용될 수 있는 로컬 스냅샷은 보관하십시오.

표 30 스냅샷 백업 명령

요청	경로에 추가:	설명
	경로에 추가: <i>/api/storage/v2/pools/pool/projects/project</i> 이에 더해 다음 중 하나: <ul style="list-style-type: none"> ■ <i>/filesystems/fs</i> ■ <i>/luns/lun</i> 	
GET	<i>/snapshots/snapshot/backups</i>	모든 형식의 스냅샷 백업을 모두 나열합니다.
GET	<i>/snapshots/snapshot/backups/format/backup-id/target-id</i>	지정된 스냅샷 백업을 나열합니다.
DELETE	<i>/snapshots/snapshot/backups/format/backup-id/target-id</i>	지정된 스냅샷 백업을 삭제합니다.
POST	<i>/snapshots/snapshot/backups</i>	새 스냅샷 백업을 만듭니다.

스냅샷 백업 나열

다음 예에서는 파일 시스템 `f-1`에서 모든 형식의 스냅샷 `snap0`에 대한 클라우드 백업을 모두 나열합니다.

요청 예:

```
GET /api/storage/v2/pools/p1/projects/default/filesystems/f-1/snapshots/snap0/backups
HTTP/1.1
Host: hostname:215
Authorization: Basic Tm8gcGVla2luZyE=
User-Agent: curl/7.54.0
Accept: */*
```

결과 예:

```
HTTP/1.1 200 OK
Date: Wed, 07 Jan 2020 20:54:47 GMT
Content-Length: 708
X-Zfssa-Storage-API: 2.0
Content-Type: application/json; charset=utf-8
X-Zfssa-API-Version: 2.0

{
  "backups": [{
    "finished": "2020-01-07T21:02:14Z",
    "format": "tar",
    "href": "/api/storage/v2/pools/p1/projects/default/filesystems/f-1/snapshots/snap0/backups/tar/backup-id/target-id",
    "id": "backup-id",
    "status": "completed",
    "target": "target-id",
    "targetName": "oci-ashburn"
  }]
}
```

다음 요청은 지정된 스냅샷 백업을 나열합니다.

```
GET /api/storage/v2/pools/p1/projects/default/filesystems/f-1/snapshots/snap0/backups/format/backup-id/target-id HTTP/1.1
```

스냅샷 백업 만들기

다음 예에서는 `tar` 형식의 스냅샷 `snap0`에 대한 백업을 만들고 대상 `oci-phoenix`에 백업을 저장합니다. 형식이 지정되지 않은 경우 `zfs` 형식으로 백업이 만들어집니다. 스냅샷 백업의 진행 상황을 보려면 “작업 나열” [116]을 사용하여 반환된 `job-id`의 작업을 확인합니다.

대상에 대해 `writelimit` 등록 정보가 설정된 경우 대상에 쓴 초당 바이트가 `writelimit` 이하입니다. “대상 만들기” [110]를 참조하십시오.

요청 예:

```
POST /api/storage/v2/pools/p1/projects/default/filesystems/f-1/snapshots/snap0/backups
HTTP/1.1
Host: hostname:215
```

```
Authorization: Basic Tm8gcGVla2luZyE=
Accept: */*
{
  "target": "oci-phoenix",
  "format": "tar"
}
```

결과 예:

```
{
  "action": "job-id"
}
```

증분 스냅샷 백업 만들기

증분 스냅샷 백업을 만들려면 `incremental` 등록 정보에 대한 값으로 `true`를 지정하고 비교에 사용할 부모 스냅샷을 지정합니다.

- 부모 스냅샷은 지정된 증분 스냅샷 백업과 동일한 로컬 시스템 및 동일한 클라우드 대상에 존재해야 합니다. 다음 예에서 스냅샷 `snap0`은 로컬 시스템과 클라우드 대상의 파일 시스템 `f-1`에 존재해야 합니다.
- 부모 스냅샷과 증분 파일 시스템 스냅샷은 동일한 형식(`zfs` 또는 `tar`)이어야 합니다.

다음 요청은 `snap0`과 파일 시스템 `f-1`의 현재 상태 간 차이인 파일 시스템 `f-1`의 백업을 만듭니다. 증분 스냅샷 백업 `snap1`은 대상 `oci-ashburn`에 저장됩니다. 증분 스냅샷 백업은 부모 스냅샷 백업과 동일한 형식입니다.

요청 예:

```
POST /api/storage/v2/pools/p1/projects/default/filesystems/f-1/snapshots/snap1/backups
HTTP/1.1
Host: hostname:215
Authorization: Basic Tm8gcGVla2luZyE=
Accept: */*
{
  "target": "oci-ashburn",
  "incremental": true,
  "parent": "snap0"
}
```

결과 예:

```
{
  "action": "job-id"
}
```

증분 스냅샷 백업의 부모 찾기

다음 예에서는 지정된 대상에서 지정된 증분 스냅샷 백업의 부모를 식별합니다. 이 예에서는 대상 `oci-ashburn`에서 스냅샷 `snap2`의 백업 부모가 스냅샷 `snap1` 및 `snap0`입니다. 결과에 따라 이 어플라이언스가 `oci-ashburn` 및 `oci-phoenix` 대상에 대한 액세스 권한을 가지고 있음을 알 수 있

습니다. `oci-ashburn`에 대한 결과가 요청된 것이므로 `oci-phoenix`에 대한 결과는 표시되지 않습니다.

요청 예:

```
POST /api/storage/v2/pools/p1/projects/default/filesystems/f-1/snapshots/snap2/backups?
props=true HTTP/1.1
Host: hostname:215
Authorization: Basic Tm8gcGV1a2luZyE=
Accept: */*
```

```
{
  "target": "oci-ashburn"
}
```

결과 예:

```
HTTP/1.1 200 OK
Date: Wed, 22 Jan 2020 22:02:17 GMT
Content-Length: 316
X-Zfssa-Storage-API: 2.0
Content-Type: application/json; charset=utf-8
X-Zfssa-API-Version: 2.0
```

```
{
  "props": [{
    "choices": [
      "oci-ashburn",
      "oci-phoenix"
    ],
    "data_type": "string",
    "label": "Backup target",
    "name": "target"
  }, {
    "choices": [
      "zfs",
      "tar"
    ],
    "data_type": "string",
    "label": "format",
    "name": "format"
  }, {
    "choices": [
      true,
      false
    ],
    "data_type": "boolean",
    "label": "Incremental",
    "name": "incremental"
  }, {
    "choices": [
      "snap1",
      "snap0"
    ],
    "data_type": "string",
    "label": "Parent",
    "name": "parent"
  }
]
```

다음 예에서는 지정된 증분 스냅샷 백업의 부모를 식별하기 위한 대체 방법으로 POST 대신 GET을 사용합니다. 이 방법에서는 대상을 지정할 필요가 없습니다. 결과에 따라 `oci-phoenix`

대상에 snap2의 부모가 없음을 알 수 있습니다. 따라서 oci-phoenix에서 snap2의 증분 백업을 만들 수 없습니다. oci-phoenix에서 snap2의 전체 백업을 만들 수 있습니다.

요청 예:

```
GET /api/storage/v2/pools/p1/projects/default/filesystems/f-1/snapshots/snap2/targets
HTTP/1.1
Host: hostname:215
Authorization: Basic Tm8gcGVla2luZyE=
Accept: */*
```

결과 예:

```
HTTP/1.1 200 OK
Date: Wed, 07 Jan 2020 22:04:08 GMT
Content-Length: 329
X-Zfssa-Storage-API: 2.0
Content-Type: application/json; charset=utf-8
X-Zfssa-API-Version: 2.0

{
  "targets": [{
    "format": "zfs",
    "href": "/api/storage/v2/pools/p1/projects/default/filesystems/f-1/snapshots/snap2/targets/zfs/target-id1",
    "id": "target-id1",
    "name": "oci-ashburn",
    "parents": [
      "snap0",
      "snap1"
    ]
  }, {
    "format": "tar",
    "href": "/api/storage/v2/pools/p1/projects/default/filesystems/f-1/snapshots/snap2/targets/tar/target-id1",
    "id": "target-id1",
    "name": "oci-ashburn",
    "parents": [
      "snap0",
      "snap1"
    ]
  }, {
    "id": "target-id2",
    "name": "oci-phoenix",
    "parents": [],
    "href": "/api/storage/v2/pools/p1/projects/default/filesystems/f-1/snapshots/snap2/targets/target-id2"
  }
]}
}
```

다음 요청을 사용하여 지정된 스냅샷 백업의 부모를 표시할 수 있습니다.

```
GET /api/storage/v2/pools/p1/projects/default/filesystems/f-1/snapshots/snap2/targets/format/target-id1 HTTP/1.1
```

스냅샷 백업 삭제

다음 예에서는 지정된 스냅샷 백업을 제거합니다. 백업 제거의 진행 상황을 보려면 ["작업 나열" \[116\]](#)을 사용하여 반환된 *job-id*의 작업을 확인합니다.

요청 예:

```
DELETE /api/storage/v2/pools/p1/projects/default/filesystems/f-1/snapshots/snap0/  
backups/format/backup-id/target-id HTTP/1.1  
Host: hostname:215  
Authorization: Basic Tm8gcGV1a2luZyE=  
Accept: */*
```

결과 예:

```
{  
  "action": "job-id"  
}
```


RESTful API 문제 서비스

RESTful API 문제 서비스는 어플라이언스 결함 관리자를 통해 발견한 문제를 보고 관리하는데 사용됩니다.

문제 서비스 명령

표 31 문제 서비스 명령

요청	경로 /problem/v{1 2}에 추가	설명
GET	/problem/v{1 2}만 사용	문제 서비스 명령을 나열합니다.
GET	/problems	모든 현재 문제를 나열합니다.
GET	/problems/problem	지정된 uuid의 문제에 대한 세부 등록 정보를 가져옵니다.
PUT	/problems/problem/markrepaired	지정된 문제 uuid를 복구됨으로 표시합니다.
GET	/suspend_notification	통지 일시 중지 여부를 보여줍니다.
PUT	/suspend_notification/enable	통지를 일시 중지합니다.
PUT	/suspend_notification/disable	통지를 재개합니다.

문제 나열

이 명령은 어플라이언스에서 현재 활성 상태인 모든 문제를 나열합니다. 성공한 명령에 대해 HTTP 상태 200(OK)이 반환됩니다.

요청 예:

```
GET /api/problem/v1/problems HTTP/1.1
Host: zfs-storage.example.com:215
Accept: application/json
```

결과 예:

```
HTTP/1.1 200 OK
Content-Type: application/json
```

```

{
  "problems": [{
    "code": "AK-8003-Y6",
    "description": "The device configuration for JBOD
                  '1204FMD063' is invalid.",
    "impact": "The disks contained within the enclosure
              cannot be used as part of a storage pool.",
    "uuid": "0d30be41-b50d-4d03-ddb4-edb69ee080f8",
    "repairable": false,
    "type": "Defect",
    "timestamp": "2013-2-21 17:37:12",
    "severity": "Major",
    "components": [{
      "certainty": 100,
      "status": "degraded",
      "uuid": "b4fd328f-92d6-4f0e-fb86-e3967a5473e7",
      "chassis": "1204FMD063",
      "label": "hc://:chassis-mfg=SUN
                :chassis-name=SUN-Storage-J4410
                :chassis-part=unknown
                :chassis-serial=1204FMD063
                :fru-serial=1204FMD063
                :fru-part=7041262
                :fru-revision=3529/ses-enclosure=0",
      "revision": "3529",
      "part": "7041262",
      "model": "Sun Disk Shelf (SAS-2)",
      "serial": "1204FMD063",
      "manufacturer": "Sun Microsystems, Inc."
    }]
  }]
}

```

문제 나열

문제 나열 명령은 단일 문제를 나열합니다. 성공한 명령에 대해 HTTP 상태 200(OK)이 반환됩니다.

문제 나열 명령은 단일 문제의 UUID인 `uuid` 입력 매개변수를 사용합니다.

요청 예:

```

GET /api/problem/v1.0/problems/0d30be41-b50d-4d03-ddb4-edb69ee080f8
HTTP/1.1
Host: zfs-storage.example.com:215
Accept: application/json

```

결과 예:

```

HTTP/1.1 200 OK
Content-Type: application/json

{
  "problem": {
    "uuid": "0d30be41-b50d-4d03-ddb4-edb69ee080f8",
    ...
  }
}

```

문제 복구

문제 복구 명령은 문제를 복구됨으로 표시합니다.

문제 복구 명령은 복구됨으로 표시될 문제의 UUID인 `uuid` 입력 매개변수를 사용합니다.

요청 예:

```
PUT /api/problem/v1/problems/0d30be41-b50d-4d03-ddb4-edb69ee080f8/repaired
  HTTP/1.1
Host: zfs-storage.example.com:215
Accept: application/json
```

성공한 응답은 HTTP 상태 202(Accepted)를 반환합니다.

```
HTTP/1.1 202 Accepted
```

문제 통지 일시 중지

어플라이언스를 서비스할 때 허위 오류가 발생할 수 있습니다. 원치 않는 SR 전송이 발생하지 않도록 하려는 경우 서비스를 수행하는 기간 동안 모든 통지를 일시 중지할 수 있습니다.

문제 통지가 일시 중지될 때 발생하는 동작에 대한 설명은 [Oracle ZFS Storage Appliance 고객 서비스 설명서의 “문제 통지 일시 중지”](#)를 참조하십시오.

통지 일시 중지 상태 보기

다음 명령을 사용하여 문제 통지 일시 중지 여부를 확인할 수 있습니다.

요청 예:

```
GET /api/problem/v2/suspend_notification HTTP/1.1
Host: zfs-storage.example.com:215
Accept: application/json
```

결과 예:

`period` 등록 정보는 통지 일시 중지에 남은 시간(분)을 보여줍니다.

```
{
  "suspend_notification": {
    "href": "/api/problem/v2/suspend_notification",
    "suspend_notification": "enabled",
    "period": 472
  }
}
```

통지 일시 중지

다음 명령을 사용하여 문제 통지를 일시 중지할 수 있습니다. 성공한 작업은 HTTP 코드 202 (Accepted)를 반환합니다.

요청 예:

```
PUT /api/problem/v2/suspend_notification/enable HTTP/1.1
Host: zfs-storage.example.com:215
Accept: application/json
```

통지 재개

다음 명령을 사용하여 문제 통지를 재개할 수 있습니다. 성공한 작업은 HTTP 코드 202 (Accepted)를 반환합니다.

요청 예:

```
PUT /api/problem/v2/suspend_notification/disable HTTP/1.1
Host: zfs-storage.example.com:215
Accept: application/json
```

RESTful API SAN 서비스

RESTful API SAN 서비스를 사용하여 어플라이언스를 SAN(Storage Area Network)에 연결할 수 있습니다.

SAN 개요

SAN에는 다음과 같은 기본 구성요소가 있습니다.

- 네트워크 스토리지에 액세스하는 클라이언트
- 네트워크 스토리지를 제공하는 스토리지 어플라이언스
- 클라이언트를 스토리지에 연결하는 네트워크

이 세 가지 구성요소는 네트워크에서 사용되는 프로토콜에 관계없이 동일합니다. 경우에 따라 개시자와 대상 간의 케이블이 네트워크일 수도 있지만 대부분의 경우에는 일종의 전환이 수반됩니다. RESTful API SAN 서비스는 지원되는 각 프로토콜에 대한 4가지 유형의 SAN 리소스를 관리합니다:

- 개시자 – SCSI 세션 시작, SCSI 명령 및 I/O 요청 전송을 처리할 수 있는 응용 프로그램 또는 운영 시스템 엔드포인트입니다. 개시자도 고유 주소 지정 방식을 통해 식별됩니다.
- 개시자 그룹 – 일련의 개시자입니다. 개시자 그룹이 LUN(Logical Unit Number)과 연관되어 있으면 해당 그룹의 개시자만 LUN에 액세스할 수 있습니다.
- 대상 – 개시자가 보낸 SCSI 명령 및 I/O 요청을 처리하는 서비스를 제공하는 스토리지 시스템 엔드포인트입니다. 대상은 스토리지 시스템 관리자가 만들며 고유 주소 지정 방식을 통해 식별됩니다. 구성된 대상은 0개 이상의 논리 장치로 이루어집니다.
- 대상 그룹 – 일련의 대상입니다. LUN은 하나의 특정 대상 그룹에 있는 모든 대상으로 내보내집니다.

SAN 개시자

다음 명령은 SAN 개시자를 관리하는 데 사용됩니다.

이러한 명령은 다음 URI 매개변수를 사용합니다:

<i>protocol</i>	개시자의 NAS 프로토콜(<i>fc</i> , <i>iscsi</i> 또는 <i>srp</i>)
<i>initiator</i>	개시자의 IQN, WWN 또는 EUI

표 32 개시자 명령

요청	경로 <i>/san/v{1 2}</i> 에 추가	설명
GET	<i>/protocol/initiators</i>	지정된 프로토콜(<i>fc</i> , <i>iscsi</i> , <i>srp</i>) 객체에 대한 모든 SAN 개시자를 나열합니다.
GET	<i>/protocol/initiators/initiator</i>	지정된 프로토콜(<i>fc</i> , <i>iscsi</i> , <i>srp</i>) 등록 정보에 대한 지정된 SAN 개시자를 가져옵니다.
POST	<i>/protocol/initiators</i>	지정된 프로토콜(<i>fc</i> , <i>iscsi</i> , <i>srp</i>)에 대한 새 SAN 개시자를 만듭니다.
PUT	<i>/protocol/initiators/initiator</i>	지정된 프로토콜(<i>fc</i> , <i>iscsi</i> , <i>srp</i>) 객체에 대한 지정된 SAN 개시자를 수정합니다.
DELETE	<i>/protocol/initiators/initiator</i>	지정된 개시자 객체를 삭제합니다.

개시자 명령 중 상당수가 다음 표에 나열된 등록 정보를 반환 값으로 사용합니다. 만들기 및 수정 명령도 해당 등록 정보를 입력 값으로 사용합니다.

표 33 개시자 등록 정보

등록 정보	프로토콜	설명
<i>alias</i>	<i>all</i>	이 개시자에 대한 별칭
<i>initiator</i>	<i>fc</i>	이 개시자에 대한 포트 WWN(World Wide Name)
<i>iqn</i>	<i>iscsi</i>	이 개시자에 대한 iSCSI 정규화된 이름
<i>chapuser</i>	<i>iscsi</i>	CHAP(Challenge handshake <i>auth</i> protocol) 사용자 이름
<i>chapsecret</i>	<i>iscsi</i>	CHAP(Challenge handshake <i>auth</i> protocol) 암호
<i>initiator</i>	<i>srp</i>	EUI(Extended Unique Identifier)

개시자 나열

지정된 프로토콜 유형의 어플라이언스에 대해 구성된 모든 개시자를 나열합니다. 응답 본문은 이름이 "initiators"인 개시자 등록 정보의 배열(JSON 형식)을 포함합니다.

iSCSI 개시자를 나열하는 요청의 예:

```
GET /api/san/v1/iscsi/initiators HTTP/1.1
Host: zfs-storage.example.com:215
Accept: application/json
```

결과 예:

```
HTTP/1.1 200 OK
Content-Type: application/json
```

```
{
  "initiators": [{
    "alias": "init-02",
    "href": "/api/san/v1/iscsi/initiators/iqn.zfs-storage.example.com.sun:02:02",
    "initiator": "iqn.zfs-storage.example.com.sun:02:02",
    "chapsecret": "",
    "chapuser": ""
  }, {
    "alias": "init-01",
    "initiator": "iqn.zfs-storage.example.com.sun:02:01",
    "href": "/api/san/v1/iscsi/initiators/iqn.zfs-storage.example.com.sun:02:01",
    "chapsecret": "",
    "chapuser": ""
  }
]}
}
```

개시자 세부정보 가져오기

단일 iSCSI 개시자의 세부정보를 나열합니다. 응답 본문은 iSCSI 개시자 등록 정보를 이름이 "initiator"인 객체(JSON 형식)로 포함합니다.

요청 예:

```
GET /api/san/v1/iscsi/initiators/iqn.zfs-storage.example.com.sun:02:01 HTTP/1.1
Host: zfs-storage.example.com:215
Accept: application/json
```

결과 예:

```
HTTP/1.1 200 OK
Content-Type: application/json

{
  "initiator": {
    "alias": "init-01",
    "href": "/api/san/v1/iscsi/initiators/iqn.zfs-storage.example.com.sun:02:01"
    "initiator": "iqn.zfs-storage.example.com.sun:02:01",
    "chapsecret": "",
    "chapuser": ""
  }
}
```

개시자 만들기

새 iSCSI 개시자를 만듭니다. IQN(iSCSI Qualified Name)을 제공해야 합니다. 요청 본문은 iSCSI 개시자 등록 정보(JSON 형식)를 포함합니다. 응답은 성공 시 HTTP 헤더 및 상태 코드 201(Created)에 새 iSCSI 개시자의 위치 URI를 포함합니다. 응답 본문은 iSCSI 개시자 등록 정보를 이름이 "initiator"인 객체(JSON 형식)로 포함합니다.

요청 예:

```
POST /api/san/v1.0/iscsi/initiators HTTP/1.1
```

```
Host: zfs-storage.example.com:215
Content-Type: application/json
Accept: application/json
```

```
{
  "initiator": "iqn.zfs-storage.example.com.sun:02:02",
  "alias": "init-02"
}
```

결과 예:

```
HTTP/1.1 201 Created
Content-Type: application/json
Content-Length: 181
X-Zfssa-San-API: 1.0
Location: /api/san/v1/iscsi/initiators/iqn.zfs-storage.example.com.sun:02:02
```

```
{
  "initiator": {
    "alias": "init-02",
    "href": "/api/san/v1/iscsi/initiators/iqn.zfs-storage.example.com.sun:02:02",
    "initiator": "iqn.zfs-storage.example.com.sun:02:02",
    "chapsecret": "",
    "chapuser": ""
  }
}
```

개시자 수정

이 명령은 기존 개시자를 수정합니다. 요청 본문은 JSON 형식으로 수정되어야 할 개시자 등록 정보를 포함합니다. 개시자에 대한 IQN은 URI에 제공됩니다. 성공 시 HTTP 상태 202 (Accepted)가 반환됩니다. 응답 본문은 새 iSCSI 개시자 등록 정보를 이름이 `initiator`인 객체 (JSON 형식)로 포함합니다.

요청 예:

```
PUT /api/san/v1/iscsi/initiators/iqn.zfs-storage.example.com.sun:01 HTTP/1.1
Host: zfs-storage.example.com:215
Content-Type: application/json
Accept: application/json
```

```
{
  "alias": "init-01-secure",
  "chapuser": "admin4",
  "chapsecret": "secret"
}
```

결과 예:

```
HTTP/1.1 202 Accepted
Content-Length: 167
Content-Type: application/json
X-Zfs-Sa-Nas-API: 1.0

{
  "initiator": {
    "alias": "init-01-secure",
```

```

        "href": "/api/san/v1/iscsi/initiators/iqn.zfs-storage.example.com.sun:01",
        "iqn": "iqn.zfs-storage.example.com.sun:1",
        "chapsecret": "secret",
        "chapuser": "admin4"
    }
}

```

개시자 삭제

어플라이언스에서 개시자를 제거합니다.

요청 예:

```

DELETE /api/san/v1/iscsi/initiators/iqn.zfs-storage.example.com.sun:01 HTTP/1.1
Host: zfs-storage.example.com:215

```

성공한 삭제는 다음과 같이 HTTP 코드 204(No Content)를 반환합니다.

```
HTTP/1.1 204 No-Content
```

개시자 그룹

iSCSI 개시자 명령은 어플라이언스에서 iSCSI 개시자 및 iSCSI 개시자 그룹을 관리하는 데 사용됩니다. 사용 가능한 명령은 아래 표에 나열되어 있습니다.

이러한 명령은 다음 URI 매개변수를 사용합니다:

protocol 개시자의 NAS 프로토콜(*fc*, *iscsi* 또는 *srp*)

name 개시자 그룹의 이름

각 개시자 그룹은 개시자 그룹의 개시자 목록을 포함하는 *name* 등록 정보 및 *initiators* 등록 정보를 포함합니다.

표 34 개시자 그룹 명령

요청	경로 <i>/san/v{1 2}</i> 에 추가	설명
GET	<i>/protocol/initiator-groups</i>	지정된 프로토콜(<i>fc</i> , <i>iscsi</i> , <i>srp</i>) 객체에 대한 모든 SAN 개시자 그룹을 나열합니다.
GET	<i>/protocol/initiator-groups/name</i>	지정된 프로토콜(<i>fc</i> , <i>iscsi</i> , <i>srp</i>) 등록 정보에 대한 지정된 SAN 개시자 그룹을 가져옵니다.
POST	<i>/protocol/initiator-groups</i>	지정된 프로토콜(<i>fc</i> , <i>iscsi</i> , <i>srp</i>)에 대한 새 SAN 개시자 그룹을 만듭니다.
PUT	<i>/protocol/initiator-groups/name</i>	지정된 프로토콜(<i>fc</i> , <i>iscsi</i> , <i>srp</i>) 객체에 대한 지정된 SAN 개시자 그룹을 수정합니다.

요청	경로 /san/v{1 2}에 추가	설명
DELETE	/protocol/initiator-groups/name	지정된 이름 객체를 삭제합니다.

개시자 그룹 나열

사용 가능한 모든 iSCSI 개시자 그룹을 나열합니다. 성공 시 HTTP 상태 200(OK)이 반환되고 본문은 개시자 그룹 객체 배열을 포함하는 이름이 "groups"인 등록 정보가 있는 JSON 객체를 포함합니다.

요청 예:

```
GET /api/san/v1/iscsi/initiator-groups HTTP/1.1
Host: zfs-storage.example.com:215
Accept: application/json
```

결과 예:

```
HTTP/1.1 200 OK
Content-Type: application/json

{
  "groups": [{
    "href": "/san/v1/iscsi/initiator-groups/p1-initiators-0",
    "initiators": ["iqn.zfs-storage.example.com.sun:0"],
    "name": "p1-initiators-0"
  }, {
    "href": "/san/v1/iscsi/initiator-groups/p1-initiators-1",
    "initiators": ["iqn.zfs-storage.example.com.sun:1"],
    "name": "p1-initiators-1"
  }
]
```

개시자 그룹 세부정보 가져오기

단일 iSCSI 개시자 그룹에서 세부 정보를 가져옵니다. 개시자 그룹 나열 명령에서 반환된 href 등록 정보를 따라가면 이 그룹에 액세스할 수 있습니다.

요청 예:

```
GET /api/san/v1/iscsi/initiator-groups/test-group HTTP/1.1
Host: zfs-storage.example.com:215
Accept: application/json
```

결과 예:

```
HTTP/1.1 200 OK
Content-Type: application/json

{
  "group": {
    "href": "/api/san/v1/iscsi/initiator-groups/test-group"
    "initiators": ["iqn.zfs-storage.example.com.sun:02:01"],
    "name": "test-group"
  }
}
```

```
}
}
```

개시자 그룹 만들기

구성원 없이 iSCSI 개시자 그룹을 만듭니다. 요청 본문은 그룹 이름을 포함하는 단일 `name` 매개 변수가 있는 JSON 객체를 포함합니다.

표 35 개시자 그룹 만들기 등록 정보

등록 정보	유형	설명
<code>name</code>	string	개시자 그룹의 이름
<code>initiators</code>	array	기존 개시자 IQN 등록 정보의 배열

요청 예:

```
POST /api/san/v1/iscsi/initiator-groups HTTP/1.1
Host: zfs-storage.example.com:215
Content-Type: application/json
Content-Length: 64
Accept: application/json
```

```
{
  "name": "group-01",
  "initiators": ["iqn.zfs-storage.example.com.sun:02"]
}
```

결과 예:

```
HTTP/1.1 201 Created
Content-Type: application/json
Location: /api/san/v1/iscsi/initiator-groups/test-group
```

```
{
  "group": {
    "href": "/api/san/v1/iscsi/initiator-groups/test-group",
    "initiators": ["iqn.zfs-storage.example.com.sun:02"],
    "name": "group-01"
  }
}
```

개시자 그룹 삭제

어플라이언스에서 개시자 그룹을 제거합니다.

요청 예:

```
DELETE /api/san/v1.0/iscsi/initiator-groups/group-01 HTTP/1.1
Host: zfs-storage.example.com:215
```

성공한 삭제는 HTTP 상태 204(No Content)를 반환합니다.

HTTP/1.1 204 No-Content

대상

iSCSI 대상 명령은 iSCSI 대상 및 iSCSI 대상 그룹을 관리하는 데 사용됩니다. 사용 가능한 명령은 아래 표에 나열되어 있습니다.

대상 명령은 다음 URI 매개변수를 사용합니다.

protocol SAN 프로토콜(*fc*, *iscsi* 또는 *srp*)

target 대상 ID(IQN, WWN 또는 EUI)

표 36 대상 명령

요청	경로 <i>/san/v{1}2</i> 에 추가	설명
GET	<i>/protocol/targets</i>	지정된 프로토콜(<i>fc</i> , <i>iscsi</i> , <i>srp</i>) 객체에 대한 모든 SAN 대상을 나열합니다.
GET	<i>/protocol/targets/target</i>	지정된 프로토콜(<i>fc</i> , <i>iscsi</i> , <i>srp</i>) 등록 정보에 대한 지정된 SAN 대상을 가져옵니다.
POST	<i>/protocol/targets</i>	지정된 프로토콜(<i>fc</i> , <i>iscsi</i> , <i>srp</i>)에 대한 새 SAN 대상을 만듭니다.
PUT	<i>/protocol/targets/target</i>	지정된 프로토콜(<i>fc</i> , <i>iscsi</i> , <i>srp</i>) 객체에 대한 지정된 SAN 대상을 수정합니다.
DELETE	<i>/protocol/targets/target</i>	지정된 대상 객체를 삭제합니다.

대상 가져오기 명령은 대상 등록 정보를 반환합니다. 대상 만들기 및 수정 명령은 다음 표에 나열된 등록 정보를 입력으로 사용합니다.

표 37 대상 입력 등록 정보

등록 정보	프로토콜	설명
<i>alias</i>	<i>iscsi</i>	사람이 읽을 수 있는 간단한 이름
<i>iqn</i>	<i>iscsi</i>	iSCSI 정규화된 이름
<i>state</i>	<i>iscsi</i>	iSCSI 대상 상태("online", "offline")
<i>auth</i>	<i>iscsi</i>	선택적 인증 유형("none", "chap")
<i>targetchapuser</i>	<i>iscsi</i>	선택적 CHAP 사용자 인증
<i>targetchapsecret</i>	<i>iscsi</i>	선택적 CHAP 암호 인증
<i>interfaces</i>	<i>iscsi</i>	대상을 사용할 수 있는 네트워크 인터페이스 목록
<i>wwn</i>	<i>fc</i>	이 대상에 대한 Worldwide 이름
<i>port</i>	<i>fc</i>	포트의 물리적 위치

등록 정보	프로토콜	설명
mode	fc	이 포트의 모드(개시자 또는 대상)
speed	fc	이 포트의 협상 속도
discovered_ports	fc	검색된 원격 개시자 포트의 수
alias	srp	SRP 대상에 대한 별칭
eui	srp	이 대상의 EUI(Extended Unique Identifier)

다음 등록 정보는 iSCSI 대상 그룹 정보를 가져오는 데 사용됩니다.

표 38 대상 그룹 등록 정보

등록 정보	유형	설명
protocol	string	대상 그룹 프로토콜(FC, iSCSI 또는 SRP)
name	string	iSCSI 대상 그룹 이름
targets	array	iSCSI 대상 IQN 그룹 구성원 목록

대상 나열

어플라이언스에서 사용할 수 있는 지정된 프로토콜의 모든 SAN 대상을 나열합니다.

요청 예:

```
GET /api/san/v1/iscsi/targets HTTP/1.1
Host: zfs-storage.example.com:215
Authorization: Basic Tm8gcGVla2luZyE=
Accept: application/json
```

결과 예:

```
HTTP/1.1 200 OK
Content-Type: application/json
Content-Length: 1337

{
  "size": 7,
  "targets": [{
    "alias": "tst.volumes.py.12866.target",
    "href": "/api/san/v1/iscsi/targets/iqn.zfs-storage.example.com.sun:02:72b6fa9a-96c4-e511-db19-aadb9bac2052",
    "iqn": "iqn.zfs-storage.example.com.sun:02:72b6fa9a-96c4-e511-db19-aadb9bac2052",
    ...
  }, {
    "alias": "tst.volumes.py.96238.target",
    "href": "/api/san/v1/iscsi/targets/iqn.zfs-storage.example.com.sun:02:31d26d2e-6aa0-6054-fe58-8b1fb508b008",
    "iqn": "iqn.zfs-storage.example.com.sun:31d26d2e-6aa0-6054-fe58-8b1fb508b008",
    ...
  }
  ...]
}
```

대상 세부정보 가져오기

단일 대상에서 등록 정보를 가져옵니다. "iqn" 등록 정보 또는 "alias=*alias*"를 사용하여 대상을 선택할 수 있습니다.

요청 예:

```
GET /api/san/v1/iscsi/targets/alias=test-target HTTP/1.1
Host: zfs-storage.example.com:215
Authorization: Basic Tm8gcGVla2luZyE=
Accept: application/json
```

결과 예:

```
HTTP/1.1 200 OK
Content-Type: application/json
Content-Length: 251

{
  "target": {
    "alias": "test-target",
    "auth": "none",
    "href": "/api/san/v1/iscsi/targets/alias=test-target",
    "interfaces": ["ixgbe0"],
    "iqn": "iqn.zfs-storage.example.com.sun:02:31d26d2e-6aa0-6054-fe58-8b1fb508b008",
    "targetchapsecret": "",
    "targetchapuser": ""
  }
}
```

대상 만들기

새 대상을 만듭니다. 요청 본문은 새 iSCSI 대상 그룹의 이름인 단일 `name` 등록 정보가 있는 JSON 객체를 포함합니다.

요청 예:

```
POST /api/san/v1/iscsi/targets HTTP/1.1
Host: zfs-storage.example.com:215
Authorization: Basic Tm8gcGVla2luZyE=
Content-Type: application/json
Content-Length: 23
Accept: application/json
```

```
{"alias": "test-target"}
```

결과 예:

```
HTTP/1.1 201 Created
Content-Type: application/json
Content-Length: 233
X-Zfssa-San-API: 1.0
Location: /api/san/v1/iscsi/targets/iqn.zfs-storage.example.com.sun:02:31d26d2e-6aa0-6054-fe58-8b1fb508b008

{
```

```

    "target": {
      "href": "/api/san/v1/iscsi/targets/iqn.zfs-
storage.example.com.sun:02:31d26d2e-6aa0-6054-fe58-8b1fb508b008",
      "alias": "test-target",
      "iqn": "iqn.zfs-storage.example.com.sun:02:31d26d2e-6aa0-6054-fe58-8b1fb508b008",
      "auth": "none",
      "targetchapuser": "",
      "targetchapsecret": "",
      "interfaces": ["ixgbe0"]
    }
  }
}

```

대상 수정

기존 iSCSI 대상을 수정합니다. 요청 본문은 수정되는 iSCSI 대상 등록 정보를 포함하는 JSON 객체를 포함합니다. 성공 시 HTTP 상태 202(Accepted)가 반환됩니다. 응답 본문은 JSON 객체로 인코딩된 대상에 대한 결과 iSCSI 대상 등록 정보를 포함합니다.

요청 예:

```

PUT /api/san/v1/iscsi/targets/alias=test-target HTTP/1.1
Host: zfs-storage.example.com:215
Authorization: Basic Tm8gcGVla2luZyE=
Content-Type: application/json
Content-Length: 54
Accept: application/json

```

```

{"targetchapsecret":"secret", "auth":"chap",
 "targetchapuser":"admin5"}

```

결과 예:

```

HTTP/1.1 202 Accepted
Content-Type: application/json
Content-Length: 189
X-Zfssa-San-API: 1.0

```

```

{
  "target": {
    "href": "/api/san/v1/iscsi/targets/alias=test-target",
    "auth": "chap",
    "targetchapsecret": "secret",
    "alias": "test-target",
    "iqn": "iqn.zfs-storage.example.com.sun:02:31d26d2e-6aa0-6054-fe58-8b1fb508b008",
    "targetchapuser": "admin5",
    "interfaces": ["ixgbe0"]
  }
}

```

대상 삭제

시스템에서 SAN 대상을 제거합니다.

요청 예:

```
DELETE /api/san/v1/iscsi/targets/iqn.zfs-storage.example.com.sun:02:e7e688b1 HTTP/1.1
Host: zfs-storage.example.com:215
Authorization: Basic Tm8gcGVla2luZyE=
```

성공한 삭제는 다음과 같이 HTTP 코드 204(No Content)를 반환합니다.

```
HTTP/1.1 204 No-Content
```

대상 그룹

대상 그룹은 대상의 모음입니다. 대상 그룹 명령은 아래 표에 나열되어 있습니다.

대상 그룹 명령은 다음 URI 매개변수를 사용합니다.

protocol 개시자의 NAS 프로토콜(*fc*, *iscsi* 또는 *srp*)

target-group 대상 그룹의 이름

표 39 대상 그룹 명령

요청	경로 <i>/san/v{1 2}</i> 에 추가	설명
GET	<i>/protocol/target-groups</i>	지정된 프로토콜(<i>fc</i> , <i>iscsi</i> 또는 <i>srp</i>) 객체에 대한 모든 SAN 대상 그룹을 나열합니다.
GET	<i>/protocol/target-groups/target-group</i>	지정된 프로토콜(<i>fc</i> , <i>iscsi</i> 또는 <i>srp</i>) 등록 정보에 대한 지정된 SAN 대상 그룹을 가져옵니다.
POST	<i>/protocol/target-groups</i>	지정된 프로토콜(<i>fc</i> , <i>iscsi</i> 또는 <i>srp</i>)에 대한 새 SAN 대상 그룹을 만듭니다.
PUT	<i>/protocol/target-groups/target-group</i>	지정된 프로토콜(<i>fc</i> , <i>iscsi</i> 또는 <i>srp</i>) 객체에 대한 지정된 SAN 대상 그룹을 수정합니다.
DELETE	<i>/protocol/target-groups/target-group</i>	지정된 대상-그룹 객체를 삭제합니다.

대상 그룹 나열

어플라이언스에 대해 사용할 수 있는 모든 대상 그룹을 나열합니다. 성공 시 HTTP 상태 200 (OK)이 반환되고 본문은 대상 그룹 객체 배열을 포함하는 이름이 *groups*인 등록 정보가 있는 JSON 객체를 포함합니다.

요청 예:

```
GET /api/san/v1/iscsi/target-groups
Host: zfs-storage.example.com:215
Authorization: Basic Tm8gcGVla2luZyE=
Accept: application/json
```

결과 예:

```

HTTP/1.1 200 OK
Content-Type: application/json
Content-Length: 237

{
  "groups": [{
    "href": "/api/san/v1/iscsi/target-groups/test-group",
    "name": "test-group",
    "targets": [
      "iqn.zfs-storage.example.com.sun:02:31d26d2e-6aa0-6054-fe58-8b1fb508b008"
    ]
  }, {
    "href": "/api/san/v1/iscsi/target-groups/alt-group",
    ...
  }]
}

```

대상 그룹 가져오기

단일 대상 그룹을 가져옵니다. 요청은 대상 그룹 이름인 단일 URI 매개변수를 사용합니다. 응답 본문은 대상 그룹 등록 정보를 포함하는 이름이 `group`인 JSON 객체 등록 정보를 포함합니다.

요청 예:

```

GET /api/san/v1/iscsi/target-groups/test-group
Host: zfs-storage.example.com:215
Authorization: Basic Tm8gcGVla2luZyE=
Accept: application/json

```

결과 예:

```

HTTP/1.1 200 OK
Content-Type: application/json

{
  "group": {
    "href": "/api/san/v1/iscsi/target-groups/test-group",
    "name": "test-group",
    "targets": [
      "iqn.zfs-storage.example.com.sun:02:0d5a0ed8-44b6-49f8-a594-872bf787ca5a"
    ]
  }
}

```

대상 그룹 만들기

새 iSCSI 대상 그룹을 만듭니다. 요청 본문은 새 그룹의 이름인 단일 `name` 등록 정보가 있는 JSON 객체입니다.

요청 예:

```

POST /api/san/v1/iscsi/target-groups HTTP/1.1
Host: zfs-storage.example.com:215
Authorization: Basic Tm8gcGVla2luZyE=

```

```
Accept: application/json
Content-Type: application/json
Content-Length: 97

{"name": "test-group",
 "targets": ["iqn.zfs-storage.example.com.sun:02:31d26d2e-6aa0-6054-fe58-8b1fb508b008"]}
```

결과 예:

```
HTTP/1.1 201 Created
Content-Type: application/json
Content-Length: 154
X-Zfssa-San-API: 1.0
Location: /api/san/v1/iscsi/target-groups/test-group

{
  "group": {
    "href": "/api/san/v1/iscsi/target-groups/test-group",
    "name": "test-group",
    "targets": [
      "iqn.zfs-storage.example.com.sun:02:31d26d2e-6aa0-6054-fe58-8b1fb508b008"
    ]
  }
}
```

대상 그룹 삭제

기존 대상 그룹을 삭제합니다.

요청 예:

```
DELETE /api/san/v1.0/iscsi/target-groups/test-group
```

성공한 삭제는 HTTP 상태 204(No Content)를 반환합니다.

```
HTTP/1.1 204 No-Content
```

서비스 명령

서비스 RESTful API는 어플라이언스에서 실행 중인 소프트웨어 서비스를 나열하고 관리하는데 사용됩니다.

서비스 명령

다음 서비스 명령을 사용할 수 있습니다.

표 40 서비스 명령

요청	경로 <code>/service/v{1 2}</code> 에 추가	설명
GET	<code>/service/v{1 2}</code> 만 사용	서비스 명령을 나열합니다.
GET	<code>/services</code>	모든 서비스를 나열합니다.
GET	<code>/services/service</code>	지정된 서비스에 대한 구성 및 상태를 가져옵니다.
PUT	<code>/services/service</code>	지정된 서비스에 대한 구성을 수정합니다.
PUT	<code>/services/service/enable</code>	지정된 서비스를 사용으로 설정합니다.
PUT	<code>/services/service/disable</code>	지정된 서비스를 사용 안함으로 설정합니다.

서비스 나열

이 명령은 스토리지 어플라이언스에서 사용할 수 있는 구성 가능한 서비스 목록을 해당 사용 상태와 함께 반환합니다. 성공한 명령에 대해 HTTP 상태 200(OK)이 반환됩니다.

요청 예:

```
GET /api/service/v1/services HTTP/1.1
Host: zfs-storage.example.com:215
Accept: application/json
```

결과 예. 간략히 나타내기 위해 항목은 대부분 생략됩니다.

```
HTTP/1.1 200 OK
Content-Type: application/json; charset=utf-8
Transfer-Encoding: chunked
```

```
X-Zfssa-Service-API: 1.0

{
  "services": [
    {
      "<status>": "disabled",
      "href": "/api/service/v1/services/ad",
      "name": "ad",
      "log": {
        "href": "/api/log/v1/logs/appliance-kit-adstat:default",
        "size": 2
      }
    },
    {
      "<status>": "online",
      "href": "/api/service/v1/services/nfs",
      "name": "nfs",
      "log": {
        "href": "/api/log/v1/logs/appliance-kit-nfsconf:default",
        "size": 8
      }
    },
    {
      "<status>": "online",
      "href": "/api/service/v1/services/ssh",
      "name": "ssh",
      "log": {
        "href": "/api/log/v1/logs/network-ssh:default",
        "size": 134
      }
    }
  ]
}
```

서비스 가져오기

이 명령은 해당 상태 및 구성을 포함하여 단일 서비스의 세부 정보를 가져옵니다.

요청 예:

```
GET /api/service/v1/services/ndmp HTTP/1.1
Host: zfs-storage.example.com:215
Accept: application/json
```

결과 예:

```
HTTP/1.1 200 OK
Content-Type: application/json

{
  "service": {
    "cram_md5_password": "",
    "cram_md5_username": "",
    "dar_support": true,
    "default_pools": [],
    "drive_type": "sysv",
    "href": "/api/service/v1/services/ndmp",
    "ignore_ctime": false,
    "name": "ndmp",
```

```

        "restore_fullpath": false,
        "status": "online",
        "tcp_port": 10000,
        "version": 4,
        "zfs_force_override": "off",
        "zfs_token_support": false
    }
}

```

서비스 상태 변경

이 명령은 지정된 서비스의 상태를 변경합니다. 다음 URI 매개변수가 사용됩니다.

<i>service</i>	서비스 이름
<i>state</i>	새 서비스 상태(enable 또는 disable)

요청 예:

```

PUT /api/service/v1/services/replication/enable HTTP/1.1
Host: zfs-storage.example.com:215
Accept: application/json

```

성공한 응답은 HTTP 상태 202(Accepted)를 반환합니다.

또한 JSON 요청을 서비스에 전송하여 서비스를 사용으로 설정하거나 사용 안함으로 설정할 수 있습니다.

JSON을 사용하는 요청의 예:

```

PUT /api/service/v1/services/replication HTTP/1.1
Host: zfs-storage.example.com:215
Accept: application/json
Content-Type: application/json
Content-Length: 22

```

```

{"<status>": "enable"}

```

서비스를 사용 안함으로 설정하려면 다음 JSON을 전송합니다.

```

{"<status>": "disable"}

```

서비스 구성 수정

지정된 서비스에 대한 구성 등록 정보는 헤더에 정의된 새 등록 정보 값과 함께 `PUT` 요청을 전송하여 수정할 수 있습니다. 일부 서비스는 하위 리소스를 포함할 수 있으며 이러한 서비스는 하위 리소스에 정의된 `href`를 따라가면 수정할 수 있습니다. 성공한 응답은 HTTP 상태 202 (Accepted)를 반환합니다.

다음 예에서는 LDAP 서비스에 대한 서버 목록을 재정렬하고 서비스 목록이 우선 순위대로 나타나도록 지정합니다. LDAP 구성에 대한 자세한 내용은 [“LDAP Configuration” in Oracle ZFS Storage Appliance Administration Guide, Release OS8.8.x](#)을 참조하십시오.

현재 상태:

```
GET /api/service/v1/services/ldap HTTP/1.1
Host: zfs-storage.example.com:215
Accept: application/json

HTTP/1.1 200 OK
Content-Type: application/json

{
  "service":{
    "href":"/api/service/v1/services/ldap",
    "<status>":"online",
    "servers":[
      "ldap-server2.us.example.com:484",
      "ldap-server1.us.example.com:636"
    ],
    "use_server_order":false,
    "proxy_dn":"",
    "proxy_password":false,
    "base_dn":"dc=us,dc=oracle,dc=com",
    "search_scope":"one",
    "cred_level":"proxy",
    "auth_method":"simple",
    "use_tls":false,
    "user_search":[
    ],
    "user_mapattr":[
    ],
    "user_mapobjclass":[
    ],
    "group_search":[
    ],
    "group_mapattr":[
    ],
    "group_mapobjclass":[
    ],
    "netgroup_search":[
    ],
    "netgroup_mapattr":[
    ],
    "netgroup_mapobjclass":[
    ],
    "server-000":{
      "host":"ldap-server2.us.example.com",
      "port":484,
      "status": "online",
      "last_seen": "142s",
      "rtt": "70.285ms",
      "err_msg": "",
      "href":"/api/service/v1/services/ldap/server-000"
    },
    "server-001":{
      "host":"ldap-server1.us.example.com",
      "port":636,
      "status": "online",
      "last_seen": "142s",
      "rtt": "126.013ms",
      "err_msg": ""
    }
  }
}
```

```

        "href":"/api/service/v1/services/ldap/server-001"
      }
    }
  }
}

```

요청 예:

```

PUT api/service/v1/services/ldap HTTP/1.1
Host: zfs-storage.example.com:215
Content-Type: application/json

```

```

{
  "servers": ["ldap-server1.us.example.com:636,ldap-server2.us.example.com:484"],
  "use_server_order":true
}

```

결과 예:

```

HTTP/1.1 202 Accepted
Content-Length: 1295
Content-Type: application/json; charset=utf-8
X-Zfssa-Service-API: 1.0

```

```

{
  "service":{
    "href":"/api/service/v1/services/ldap",
    "<status>":"online",
    "servers":[
      "ldap-server1.us.example.com:636",
      "ldap-server2.us.example.com:484"
    ],
    "use_server_order":true,
    "proxy_dn":"",
    "proxy_password":false,
    "base_dn":"dc=us,dc=oracle,dc=com",
    "search_scope":"one",
    "cred_level":"proxy",
    "auth_method":"simple",
    "use_tls":false,
    "user_search":[
    ],
    "user_mapattr":[
    ],
    "user_mapobjclass":[
    ],
    "group_search":[
    ],
    "group_mapattr":[
    ],
    "group_mapobjclass":[
    ],
    "netgroup_search":[
    ],
    "netgroup_mapattr":[
    ],
    "netgroup_mapobjclass":[
    ],
    "server-000":{
      "host":"ldap-server1.us.example.com",
      "port":636,
      "status": "online",
      "last_seen": "142s",
      "rtt": "126.013ms",
      "err_msg": ""
    }
  }
}

```

```

        "href": "/api/service/v1/services/ldap/server-000"
    },
    "server-001": {
        "host": "ldap-server2.us.example.com",
        "port": 484,
        "status": "online",
        "last_seen": "142s",
        "rtt": "70.285ms",
        "err_msg": "",
        "href": "/api/service/v1/services/ldap/server-001"
    }
}
}
}

```

서비스 리소스

일부 서비스에 하위 리소스가 있습니다. 각 서비스 또는 서비스 명령 목록에 대해 반환된 데이터를 확인하여 사용할 수 있는 하위 리소스를 검토합니다.

표 41 서비스 하위 리소스 명령

요청	경로	설명
GET	<i>/services/service/resource</i>	서비스 하위 리소스 나열
PUT	<i>/services/service/resource/href</i>	하위 리소스 수정
POST	<i>/services/service/resource</i>	새 하위 리소스를 만듭니다.
DELETE	<i>/services/service/resource/href</i>	하위 리소스를 삭제합니다.

이러한 명령 각각은 다른 RESTful API 명령과 동일한 패턴을 사용합니다. 여기서 GET은 지정된 하위 리소스 유형을 나열하거나 가져오는 데 사용되고, POST는 새 하위 리소스 유형을 만드는 데 사용되고, PUT은 하위 리소스를 수정하는 데 사용되고, DELETE는 지정된 하위 리소스를 삭제하는 데 사용됩니다.

하위 리소스와 각 하위 리소스에 대해 사용할 수 있는 등록 정보 및 명령 목록은 CLI "구성 서비스" 설명서를 참조하십시오.

RESTful API 스토리지 서비스

RESTful API 스토리지 서비스는 구성을 보고 스토리지 풀, 프로젝트, 파일 시스템 및 LUN의 측면을 관리하는 데 사용됩니다. 또한 스냅샷 및 복제를 관리합니다.

스토리지 풀 작업

Oracle ZFS 스토리지 어플라이언스에서 NAS는 모든 LUN 및 파일 시스템에서 동일한 데이터 중복 특성을 결정짓는 풀에서 구성되며, 풀 작업은 어플라이언스 스토리지 구성을 가져오는데 사용됩니다.

표 42 스토리지 풀 명령

요청	경로 <code>/api/storage/v{1 2}</code> 에 추가	설명
GET	<code>/pools</code>	모든 스토리지 풀을 나열합니다.
GET	<code>/pools/pool</code>	스토리지 풀 세부정보를 가져옵니다.
POST	<code>/pools</code>	새 스토리지 풀을 구성합니다.
PUT	<code>/pools/pool</code>	풀에 스토리지를 추가하거나 풀에서 스토리지를 제거합니다.
PUT	<code>/pools/pool/scrub</code>	지정된 풀에서 데이터 스크러빙을 시작합니다.
DELETE	<code>/pools/pool/scrub</code>	지정된 풀에서 데이터 스크러빙 작업을 중지합니다.
DELETE	<code>/pools/pool</code>	지정된 스토리지 풀을 구성 해제합니다.

풀 나열

이 명령은 시스템에서 모든 스토리지 풀의 등록 정보를 나열합니다. 성공한 명령에 대해 HTTP 상태 200(OK)이 반환됩니다. HTTP 본문은 각 풀을 설명하는 JSON 객체 목록을 포함합니다. 등록 정보 이름은 다음 표에 나와 있습니다.

주 - `depth` 질의 매개변수와 `match_property-name=value` 질의 매개변수는 지원되지 않습니다.

표 43 스토리지 풀 등록 정보

등록 정보	유형	설명
asn	string	풀을 소유하는 어플라이언스의 일련 번호
name	string	대상 풀 이름
owner	string	풀을 소유하는 시스템의 호스트 이름
peer	string	클러스터화된 시스템에서 어플라이언스 클러스터의 파트너 헤드
profile	string	데이터 장치 프로파일
scrub_schedule	string	예약된 풀 스크러빙 작업 간 일 수입니다. 또는 예약된 풀 스크러빙을 사용 안함으로 설정합니다. 허용되는 값 및 추가 풀 스크러빙 등록 정보는 " 풀 스크러빙 " [154]을 참조하십시오.
state	string	풀 상태: online, offline, exported

요청 예:

```
GET /api/storage/v1/pools HTTP/1.1
Host: zfs-storage.example.com:215
Accept: application/json
```

결과 예:

```
HTTP/1.1 200 OK
Content-Type: application/json

{
  "pools": [
    {
      "profile": "mirror3",
      "name": "m1",
      "peer": "peer-hostname",
      "state": "online",
      "owner": "system-hostname",
      "asn": "appliance-serial-number",
      "scrub_schedule": "30 days"
    },
    {
      "profile": "raidz1",
      "name": "r1",
      "peer": "peer-hostname",
      "state": "online",
      "owner": "system-hostname",
      "asn": "appliance-serial-number",
      "scrub_schedule": "30 days"
    }
  ]
}
```

풀 가져오기

이 명령은 풀에 대한 스토리지 사용량 정보와 함께 단일 스토리지 풀에서 등록 정보를 반환합니다. 성공한 명령에 대해 HTTP 상태 200(OK)이 반환됩니다.

요청 예:

```
GET /api/storage/v1/pools/p1 HTTP/1.1
Host: zfs-storage.example.com:215
Accept: application/json
```

결과 예:

```
HTTP/1.1 200 OK
Content-Type: application/json
```

```
{
  "pool": {
    "profile": "raidz1",
    "name": "p1",
    "usage": {
      "available": 57454799311352.0,
      "compression": 1.0,
      "dedupratio": 672791,
      "free": 57454799311352.0,
      "total": 74732430950400.0,
      "usage_child_reservation": 0.0,
      "usage_data": 16011663438848.0,
      "usage metasize": 0.0,
      "usage metaused": 0.0,
      "usage_replication": 1693675705344.0,
      "usage_reservation": 0.0,
      "usage_snapshots": 123913627136.0,
      "usage_total": 17829252771328.0,
      "used": 17829252771328.0
    },
    "peer": "00000000-0000-0000-0000-000000000000",
    "state": "online",
    "owner": "admin1",
    "asn": "2f4aeeb3-b670-ee53-e0a7-d8e0ae410749"
  }
}
```

풀 구성

풀을 구성합니다. 풀을 만드는 데 필요한 매개변수에 대해서는 CLI 구성 스토리지 명령을 참조하십시오. 사용 가능한 등록 정보 이름 및 값을 반환하는 풀을 만드는 dry run 요청을 수행할 수 있습니다. props 질의 매개변수 등록 정보를 true로 설정하면 됩니다.

요청 예:

```
POST /api/storage/v1/pools?props=true HTTP/1.1
Host: zfs-storage.example.com:215
Authorization: Basic Tm8gcGVla2luZyE=
Content-Type: application/json
Accept: application/json
```

```
{
  "name": "p1",
}
```

결과 예:

```
HTTP/1.1 200 OK
```

```
Content-Type: application/json

"props": [{
  "choices": ["custom" ],
  "label": "Chassis 0",
  "name": "0",
  "type": "ChooseOne"
}, {
  "choices": ["custom"],
  "label": "Chassis 1",
  "name": "1",
  "type": "ChooseOne"
}, {
  "choices": [0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12],
  "label": "Chassis 1 data",
  "name": "1-data",
  "type": "ChooseOne"
}, {
  "choices": ["mirror", "mirror3", "raidz1",
    "raidz2", "raidz3_max", "stripe"],
  "label": "Data Profile",
  "name": "profile",
  "type": "ChooseOne"
}
]
```

새시 [1]에서 8개의 디스크를 사용하는 풀을 만드는 요청의 예:

```
POST /api/storage/v1/pools HTTP/1.1
Host: zfs-storage.example.com:215
Authorization: Basic Tm8gcGVla2luZyE=
Content-Type: application/json
Accept: application/json
```

```
{
  "name": "p1",
  "profile": "stripe",
  "1-data": 8
}
```

결과 예:

```
HTTP/1.1 201 Created
Content-Type: application/json

{
  "pool": {
    "asn": "314d252e-c42b-e844-dab1-a3bca680b563",
    "errors": [],
    "name": "p1",
    "owner": "zfs-storage",
    "peer": "00000000-0000-0000-0000-000000000000",
    "profile": "stripe",
    "status": "online",
    "usage": {
      "available": 1194000466944.0,
      "dedupratio": 100,
      "total": 1194000908288.0,
      "used": 441344.0
    }
  }
}
```

풀에 스토리지 추가

이 명령은 풀을 만들거나 구성하는 명령과 유사합니다. 스토리지를 추가는 기존 풀에 추가 스토리지 장치를 추가합니다. 풀에 추가할 필요한 스토리지 장치 수가 포함된 본문과 함께 `href pool/add`를 전송합니다.

요청 예:

```
PUT /api/storage/v1/pools/p1/add HTTP/1.1
Host: zfs-storage.example.com:215
Authorization: Basic Tm8gcGV1a2luZyE=
Content-Type: application/json
Accept: application/json
```

```
{
  "2-data": 8
}
```

결과 예:

```
HTTP/1.1 202 Accepted
```

풀에서 스토리지 제거

이 명령은 풀에 스토리지를 추가하는 명령과 유사합니다. 스토리지를 제거하면 기존 풀에서 캐시 및 로그 스토리지 장치가 제거됩니다. 풀에서 제거할 스토리지 장치 수, 필요한 유형 및 새시 번호가 포함된 본문과 함께 `href pool/remove`를 전송합니다.

요청 예:

```
PUT /api/storage/v1/pools/p1/remove HTTP/1.1
Host: zfs-storage.example.com:215
Authorization: Basic Tm8gcGV1a2luZyE=
Content-Type: application/json
Accept: application/json
```

```
{
  "0-cache" : 2
}
```

결과 예:

```
HTTP/1.1 202 Accepted
```

제거할 수 있는 장치 수를 표시하려면 `props` 질의 매개변수를 `true`로 설정합니다.

요청 예:

```
PUT /api/storage/v1/pools/p1/remove?props=true HTTP/1.1
Host: zfs-storage.example.com
Authorization: Basic Tm8gcGV1a2luZyE=
```

```
Content-Type: application/json
Accept: application/json
```

결과 예:

```
HTTP/1.1 200 OK
Content-Type: application/json
```

```
{
  "props": [
    {
      "choices": [
        "0",
        "1",
        "2"
      ],
      "type": "ChooseOne",
      "name": "0-cache",
      "label": "Chassis 0 cache"
    },
    {
      "choices": [
        "0",
        "1",
        "2"
      ],
      "type": "ChooseOne",
      "name": "1-log",
      "label": "Chassis 1 log"
    }
  ]
}
```

풀 스크러빙

`pool/scrub PUT` 요청을 전송하면 풀 스크러빙 작업이 시작됩니다. `pool/scrub DELETE` 요청을 전송하면 실행 중인 스크러빙 작업이 중지됩니다. 풀 스크러빙에 대한 자세한 내용은 [“Scrubbing a Storage Pool – Manual \(CLI\)” in Oracle ZFS Storage Appliance Administration Guide, Release OS8.8.x](#)을 참조하십시오.

기본적으로 예약된 스토리지 풀 스크러빙이 사용으로 설정되고 30일 간격으로 설정됩니다. `scrub_schedule` 등록 정보는 예약된 풀 스크러빙 작업 간 일 수를 지정합니다. 또는 예약된 풀 스크러빙을 사용 안함으로 설정합니다. `scrub_schedule`의 기본값은 30입니다.

- 예약된 스크러빙을 사용 안함으로 설정하려면(예: 수동 스크러빙을 수행하려는 경우) `scrub_schedule` 등록 정보의 값을 `off`로 설정합니다.
- 예약된 스크러빙 작업 간 일 수를 변경하려면 `scrub_schedule` 등록 정보의 값을 15, 30, 45, 60, 75 또는 90으로 설정합니다.

예약된 풀 스크러빙에 대한 자세한 내용은 [“Scrubbing a Storage Pool – Scheduled \(CLI\)” in Oracle ZFS Storage Appliance Administration Guide, Release OS8.8.x](#)을 참조하십시오.

scrub 객체는 가장 최근 풀 스크러빙(예약됨 또는 수동)을 보고합니다.

- complete 등록 정보가 false인 경우 스크러빙이 계속 실행 중인 것입니다. 발견된 오류 수(errors)와 복구된 오류 수(repaired)가 표시됩니다.
- complete 등록 정보가 true인 경우 스크러빙이 완료된 것입니다. 스크러빙 시작 시간(op_start) 및 스크러빙 종료 시간(last_end) 등이 추가 정보로 표시됩니다. 시간은 GMT입니다.

scrub_started 및 scrub_finished는 zfs_pool 경보 조치 이벤트 범주의 이벤트이며 해당 이벤트에 대한 사용자정의 조치를 지정할 수 있습니다. [RESTful API 경보 서비스 \[47\]](#)를 참조하십시오.

다음 예에서는 스크러빙 후 부분 풀 목록을 보여줍니다.

```
HTTP/1.1 200 OK
Content-Type: application/json
{
  "pool": {
    "status": "online",
    "profile": "mirror:log_stripe:cache_stripe",
    "scrub": {
      "errors": 0,
      "op_start": "20190520T16:09:41",
      "complete": true,
      "seq_resilver": 0,
      "type": "everything",
      "examined": 403968,
      "repaired": 0,
      "last_end": "20190520T16:17:59"
    },
    "scrub_schedule": "30 days",
    "name": "p0",
    "peer": "peer-hostname",
    "owner": "system-hostname",
    "asn": "appliance-serial-number"
  }
}
```

풀 구성 해제

이 명령은 시스템에서 풀을 제거합니다.

풀을 삭제하는 요청:

```
DELETE /api/storage/v1/pools/p1 HTTP/1.1
Host: zfs-storage.example.com:215
Authorization: Basic Tm8gcGV1a2luZyE=
```

결과 예:

```
HTTP/1.0 204 No Content
Date: Fri, 02 Aug 2013 22:31:06 GMT
X-Zfssa-Nas-API: 1.0
Content-Length: 0
```

프로젝트 작업

모든 프로젝트 작업의 범위를 지정된 풀로 설정할 수 있습니다. 모든 프로젝트에서 작동하는 명령은 URI에 /projects를 추가하고 단일 프로젝트에서 작동하는 명령은 /projects/project를 추가합니다.

표 44 프로젝트 명령

요청	경로 /api/storage/v{1 2}에 추가	설명
GET	/projects	모든 프로젝트를 나열합니다.
GET	/pools/pool/projects	프로젝트를 나열합니다.
GET	/pools/pool/projects?snaps=true	스냅샷을 포함하여 모든 프로젝트를 나열합니다.
GET	/pools/pool/projects/project	프로젝트 세부정보를 가져옵니다.
POST	/pools/pool/projects	프로젝트를 만듭니다.
PUT	/pools/pool/projects/project	프로젝트를 수정합니다.
DELETE	/pools/pool/projects/project	프로젝트를 삭제합니다.
GET	/pools/pool/projects/project/usage/groups	프로젝트 그룹 사용량을 가져옵니다.
GET	/pools/pool/projects/project/usage/groups/group	지정된 그룹에 대한 프로젝트 사용량을 가져옵니다.
GET	/pools/pool/projects/project/usage/users	프로젝트 사용자 사용량을 가져옵니다.
GET	/pools/pool/projects/project/usage/users/user	지정된 사용자에 대한 프로젝트 사용량을 가져옵니다.

다음 표는 프로젝트 리소스 내에 있는 편집 가능한 등록 정보의 목록을 보여 줍니다.

표 45 프로젝트 등록 정보

등록 정보	유형	설명
aclinherit	string	ACL 상속 동작("discard", "noallow", "restricted", "passthrough", "passthrough-x", "passthrough-mode-preserve")
aclmode	string	모드 변경 시 ACL 동작("discard", "mask", "passthrough")
atime	boolean	읽기 시 액세스 시간 업데이트 플래그
canonical_name	string	표준 이름
checksum	string	블록 체크섬("fletcher2", "fletcher4", "sha256")
compression	string	데이터 압축 설정("off", "lzb", "gzip-2", "gzip", "gzip-9")
copies	number	추가 복제 복사본 수
creation	datetime	프로젝트(또는 LUN, 파일 시스템) 생성 날짜 및 시간
dedup	boolean	데이터 중복 플래그

등록 정보	유형	설명
default_group	string	프로젝트 기본 파일 시스템 그룹: "other"
default_permissions	string	프로젝트 기본 파일 시스템 권한 "700"
default_sparse	boolean	프로젝트 기본 LUN 스파스 데이터 플래그
default_user	string	프로젝트 기본 파일 시스템 사용자: "nobody"
default_volblocksize	number	프로젝트 기본 LUN 블록 크기: 8192
default_volsize	number	프로젝트 기본 LUN 크기
exported	boolean	내보냄 플래그
logbias	string	동기식 쓰기 바이어스("latency", "throughput")
mountpoint	string	마운트 지점 기본값 공유 "/export/proj-01"
name	string	프로젝트 이름
nbmand	boolean	비블로킹 필수 잠금 플래그
nodestroy	boolean	삭제 금지 플래그
quota	number	바이트 단위의 프로젝트 쿼터 크기
origin	string	원점 복제
pool	string	풀 이름
readonly	boolean	데이터는 true로 설정된 경우 읽기 전용임
recordsize	string	데이터베이스 레코드 크기 "128k"
reservation	number	데이터 예약 크기
rstchown	boolean	소유권 변경 제한 플래그
secondarycache	string	보조 캐시 사용량("all", "metadata", "none")
sharedav	string	HTTP 공유("off", "rw", "ro")
shareftp	string	FTP 공유("off", "rw", "ro")
sharenfs	string	NFS 공유("off", "on", "ro", "rw")
sharesftp	string	SFTP 공유("off", "rw", "ro")
sharesmb	string	SMB/CIFS 공유("off", "rw", "ro")
shareftp	string	TFTP 공유("off", "rw", "ro")
snapdir	string	.zfs/snapshot 가시성("hidden", "visible")
snaplabel	string	예약된 스냅샷 레이블
vscan	boolean	바이러스 검사 플래그

프로젝트 나열

이 명령은 지정된 풀의 모든 프로젝트를 나열합니다. 요청은 스토리지 풀 이름인 단일 URI 매개변수를 사용합니다. 반환된 각 프로젝트는 위에 나열된 수정 가능한 등록 정보 목록은 물론 풀 이름, 생성 시간, 로딩 상태, 복제 작업 및 데이터 사용량도 포함합니다.

주 - depth 질의 매개변수와 match_property-name=value 질의 매개변수는 지원되지 않습니다.

요청 매개변수: `filter` – 프로젝트 내의 등록 정보가 해당 값 안에 동일한 필터 문자열을 포함하도록 요구하는 단순한 문자열 일치 필터입니다.

요청 예:

```
GET /api/storage/v1/pools/p1/projects HTTP/1.1
Host: zfs-storage.example.com:215
Accept: application/json
```

`get`에 성공하면 JSON 형식의 프로젝트 등록 정보 배열과 함께 HTTP 코드 200(OK)이 반환됩니다.

결과 예:

```
HTTP/1.1 200 OK
Content-Type: application/json
```

```
{
  "projects": [{
    "name": "proj-01",
    ...
  }, {
    "name": "proj-02",
    ...
  }
}
```

모든 풀의 모든 프로젝트 목록도 지원됩니다. URI는 `/projects` 경로만 포함합니다.

`backup`을 해당 등록 정보의 일부로 사용하여 모든 프로젝트를 가져오는 요청의 예:

```
GET /projects?filter=backup HTTP/1.1
Host: zfs-storage.example.com:215
Accept: application/json
```

프로젝트 등록 정보 가져오기

이 명령은 지정된 풀의 단일 프로젝트에 대한 등록 정보를 나열합니다. 성공한 `get`은 JSON 형식의 프로젝트 등록 정보와 함께 HTTP 코드 200(OK)을 반환합니다.

`zfs-storage-1` 풀에 이름이 `proj-01`인 프로젝트를 나열하는 요청의 예:

```
GET /api/storage/v1/pools/p1/projects/proj-01 HTTP/1.1
Host: zfs-storage.example.com:215
Accept: application/json
```

결과 예:

```
HTTP/1.1 200 OK
Content-Type: application/json
```

```
{
  "project": {
    "default_volblocksize": 8192.0,
    "logbias": "latency",
    "creation": "20130411T20:02:35",
  }
}
```

```

    "nodelay": false,
    "nodelay": false,
    "dedup": false,
    "sharenfs": "on",
    "sharesmb": "off",
    "default_permissions": "700",
    "mountpoint": "/export",
    "snaplabel": "",
    "id": "042919bb-0882-d903-0000-000000000000",
    "readonly": false,
    "rrsrc_actions": [],
    "compression": "off",
    "shareftp": "",
    "default_sparse": false,
    "snapdir": "hidden",
    "aclmode": "discard",
    "copies": 1,
    "aclinherit": "restricted",
    "shareftp": "",
    "canonical_name": "zfs-storage-1/local/default",
    "recordsize": 131072.0,
    "usage": {
      "available": 1758424767306.0,
      "loading": false,
      "quota": 0.0,
      "snapshots": 0.0,
      "compressratio": 100.0,
      "child_reservation": 0.0,
      "reservation": 0.0,
      "total": 45960.0,
      "data": 45960.0
    },
    "default_volsize": 0.0,
    "secondarycache": "all",
    "collection": "local",
    "exported": true,
    "vscan": false,
    "reservation": 0.0,
    "atime": true,
    "pool": "p1",
    "default_user": "nobody",
    "name": "default",
    "checksum": "fletcher4",
    "default_group": "other",
    "shareftp": "",
    "nbmand": false,
    "sharedav": "",
    "rstchown": true
  }
}

```

프로젝트 만들기

프로젝트 만들기 명령은 지정된 스토리지 풀에 있는 지정된 이름의 프로젝트를 만듭니다. 요청은 스토리지 풀 이름인 단일 URI 매개변수를 사용합니다. 기본 등록 정보가 지정된 새 프로젝트가 반환됩니다.

JSON 본문 요청 매개변수:

- `name` – 프로젝트를 만들려면 프로젝트 이름을 제공해야 합니다.

- 프로젝트 등록 정보 – 프로젝트 등록 정보를 새 프로젝트의 초기 값으로 설정할 수 있습니다.

이름이 `proj-01`인 프로젝트를 만드는 요청의 예:

```
POST /api/storage/v1/pools/p1/projects HTTP/1.1
Host: zfs-storage.example.com:215
Content-Type: application/json
Accept: application/json
```

```
{
  "name": "proj-01",
  "sharenfs": "ro"
}
```

성공한 만들기는 새 프로젝트의 URI를 포함하는 위치 헤더와 함께 HTTP 상태 201(Created)을 반환합니다. 본문은 JSON 형식의 모든 프로젝트 등록 정보를 포함합니다.

결과 예:

```
HTTP/1.1 201 Created
Content-Type: application/json
Location: http://zfs-storage.example.com:215
          /pools/p1/projects/proj-01
```

```
{
  "project": {
    "name": "proj-01",
    "href": "/api/storage/v1/pools/p1/projects/proj-01",
    "mountpoint": "/export/acme/zfs-storage-1",
    ...
  }
}
```

프로젝트 수정

프로젝트 수정 명령은 기존 프로젝트의 속성을 변경합니다. 다음 URI 매개변수가 사용됩니다.

pool 스토리지 풀 이름

project 프로젝트 이름

요청 매개변수: 프로젝트 등록 정보 – 프로젝트 등록 정보를 새 프로젝트의 초기 값으로 설정할 수 있습니다.

프로젝트 이름을 `proj-01`에서 `new-name`으로 변경하는 요청의 예:

```
POST /api/storage/v1/pools/p1/projects/proj-01 HTTP/1.1
Host: zfs-storage.example.com:215
Content-Type: application/json
Accept: application/json
```

```
{
  "name": "new-name",
  "sharenfs": "rw",
  "compression": "gzip-9"
}
```

성공한 응답은 HTTP 상태 202(Accepted)를 반환하고 모든 프로젝트 등록 정보를 나열합니다.

결과 예:

```
HTTP/1.1 201 Created
Content-Type: application/json
Location: /api/storage/v1/pools/p1/projects/new-name
```

```
{
  "project": {
    "name": "new-name",
    "sharenfs": "rw",
    "compression": "gzip-9",
    ...
  }
}
```

프로젝트 삭제

프로젝트 삭제 명령은 지정된 풀의 단일 프로젝트를 제거합니다. 다음 URI 매개변수가 사용됩니다.

pool 스토리지 풀 이름

project 프로젝트 이름

지연된 업데이트 "비동기 데이터 세트 삭제(OS8.7.0)"가 수락된 경우 스토리지 풀에서 회수할 공간을 모니터링하려면 `pools/pool`에 대한 GET 명령을 입력합니다. `async_destroy_reclaim_space` 등록 정보에 대한 공간을 기록해 두십시오. 작업이 완료되면 0이 표시됩니다.

요청 예:

```
DELETE /api/storage/v1/pools/p1/projects/proj-01 HTTP/1.1
Host: zfs-storage.example.com:215
Accept: application/json
```

프로젝트 사용량

Get 요청 프로젝트 사용량 리소스를 통해 프로젝트에 대해 사용자별 또는 그룹별로 사용량 데이터를 가져올 수 있습니다.

파일 시스템 작업

파일 시스템 작업은 파일 시스템 공유를 나열하고 관리합니다. 모든 명령의 범위는 지정된 스토리지 풀 또는 프로젝트로 지정됩니다.

`service_uri/pools/pool/project/project`

표 46 파일 시스템 명령

요청	경로 <code>/api/storage/v{1 2}</code> 에 추가	설명
GET	<code>/filesystems</code>	모든 파일 시스템을 나열합니다.
GET	<code>/pools/pool/projects/project/filesystems</code>	지정된 파일 시스템을 나열합니다.
GET	<code>/pools/pool/projects/project/filesystems?snaps=true</code>	스냅샷을 포함하여 모든 파일 시스템을 나열합니다.
GET	<code>/pools/pool/projects/project/filesystems/filesystem</code>	파일 시스템 세부정보를 가져옵니다.
POST	<code>/pools/pool/projects/project/filesystems</code>	파일 시스템을 만듭니다.
PUT	<code>/pools/pool/projects/project/filesystems/filesystem</code>	파일 시스템을 수정합니다.
DELETE	<code>/pools/pool/projects/project/filesystems/filesystem</code>	파일 시스템을 삭제합니다.
GET	<code>/pools/pool/projects/project/filesystems/filesystem/usage/groups</code>	파일 시스템 그룹 사용량을 가져옵니다.
GET	<code>/pools/pool/projects/project/filesystems/filesystem/usage/groups/group</code>	지정된 그룹에 대한 파일 시스템 사용량을 가져옵니다.
POST	<code>/pools/pool/projects/project/filesystems/filesystem/usage/groups</code>	파일 시스템 그룹 쿼터를 만듭니다.
PUT	<code>/pools/pool/projects/project/filesystems/filesystem/usage/groups/name</code>	파일 시스템 그룹 쿼터를 수정합니다.
GET	<code>/pools/pool/projects/project/filesystems/filesystem/usage/users</code>	파일 시스템 사용자 사용량을 가져옵니다.
GET	<code>/pools/pool/projects/project/filesystems/filesystem/usage/users/user</code>	지정된 사용자에 대한 파일 시스템 사용량을 가져옵니다.
POST	<code>/pools/pool/projects/project/filesystems/filesystem/usage/users</code>	파일 시스템 사용자 쿼터를 만듭니다.
PUT	<code>/pools/pool/projects/project/filesystems/filesystem/usage/users/name</code>	파일 시스템 사용자 쿼터를 수정합니다.
GET	<code>/pools/pool/projects/project/filesystems/filesystem/shadow/errors</code>	새도우 마이그레이션 오류를 나열합니다.

각 파일 시스템은 프로젝트의 등록 정보를 포함하고 있으며 다음 파일 시스템별 등록 정보도 포함하고 있습니다.

표 47 파일 시스템 등록 정보

등록 정보	유형	설명
<code>casesensitivity</code>	string	대소문자 구분 설정(<code>mixed</code> , <code>sensitive</code> 또는 <code>insensitive</code>)
<code>group</code>	string	그룹 이름
<code>normalization</code>	string	정규화
<code>permissions</code>	string	파일 시스템 권한
<code>project</code>	string	프로젝트 이름
<code>quota_snap</code>	boolean	쿼터에 스냅샷을 포함하는 플래그
<code>reservation_snap</code>	boolean	예약에 스냅샷을 포함하는 플래그
<code>shadow</code>	string	데이터 마이그레이션 소스
<code>errors</code>	string	데이터 마이그레이션 오류

등록 정보	유형	설명
sharesmb_name	string	SMB 공유의 이름
source	object	프로젝트 상속 등록 정보
usage	object	파일 시스템 사용량 정보
user	string	공유를 소유하는 사용자 이름
utf8only	boolean	비UTF-8을 거부하는 플래그

파일 시스템 나열

파일 시스템 나열 명령은 지정된 풀 또는 프로젝트의 모든 파일 시스템을 표시합니다.

주 - depth 질의 매개변수와 match_property-name=value 질의 매개변수는 지원되지 않습니다.

요청 매개변수: filter – 프로젝트 내의 등록 정보가 해당 값 안에 동일한 필터 문자열을 포함하도록 요구하는 단순한 문자열 일치 필터입니다.

파일 시스템 나열 명령은 다음 URI 매개변수를 사용합니다.

pool 스토리지 풀 이름

project 프로젝트 이름

요청 예:

```
GET /api/storage/v1/pools/p1/projects/proj-01/filesystems HTTP/1.1
Host: zfs-storage.example.com:215
Accept: application/json
```

성공한 요청은 JSON 형식의 파일 시스템 등록 정보 배열과 함께 HTTP 상태 200(OK)을 반환합니다.

결과 예:

```
HTTP/1.1 200 OK
Content-Type: application/json

{
  "filesystems": [{
    "name": "filesystem-01",
    "project": "proj-01",
    "pool": "p1",
    ...
  }, {
    "name": "filesystem-02",
    "project": "proj-01",
    "pool": "p1",
    ...
  }]
}
```

모든 풀 및 프로젝트의 모든 파일 시스템 목록도 지원됩니다. 이 경우 URI는 `/api/storage/v{1|2}/filesystems`가 됩니다.

"abcd" 문자열을 해당 등록 정보의 일부로 사용하여 모든 파일 시스템을 가져오는 요청의 예:

```
GET /api/storage/v1/filesystems?filter=abcd HTTP/1.1
Host: zfs-storage.example.com:215
Accept: application/json
```

파일 시스템 가져오기

파일 시스템 가져오기 명령은 지정된 풀 또는 프로젝트의 단일 파일 시스템 등록 정보를 반환합니다. 다음 URI 매개변수가 사용됩니다.

<i>pool</i>	스토리지 풀 이름
<i>project</i>	프로젝트 이름
<i>filesystem</i>	파일 시스템 이름

이름이 `proj-01`인 프로젝트를 나열하는 요청의 예:

```
GET /api/storage/v1/pools/p1/projects/proj-01 HTTP/1.1
Host: zfs-storage.example.com:215
Accept: application/json
```

성공한 `get`은 JSON 형식의 파일 시스템 등록 정보와 함께 HTTP 상태 200(OK)을 반환합니다.

결과 예:

```
HTTP/1.1 200 OK
Content-Type: application/json

{
  "filesystem": {
    "logbias": "latency",
    "creation": "20130423T21:30:34",
    "nodeldestroy": false,
    "dedup": false,
    "sharenfs": "on",
    "sharesmb": "off",
    "mountpoint": "/export/mnt1",
    "snaplabel": "",
    "id": "424ca2ec-b3fa-df86-0000-000000000000",
    "readonly": false,
    "rrsrc_actions": [],
    "compression": "off",
    "sharetftp": "",
    "source": {
      "logbias": "default",
      "dedup": "default",
      "sharenfs": "inherited",
      "sharesmb": "off",
      "mountpoint": "inherited",
      "rrsrc_actions": "local",
    }
  }
}
```

```

        "compression": "default",
        "shareftp": "inherited",
        "snapdir": "default",
        "aclmode": "default",
        "copies": "default",
        "aclinherit": "default",
        "shareftp": "inherited",
        "readonly": "default",
        "secondarycache": "default",
        "exported": "inherited",
        "vscan": "default",
        "reservation": "local",
        "atime": "default",
        "recordsize": "default",
        "checksum": "inherited",
        "sharesftp": "inherited",
        "nbmand": "default",
        "rstchown": "default"
    },
    "snapdir": "hidden",
    "aclmode": "discard",
    "copies": 1,
    "aclinherit": "restricted",
    "shareftp": "",
    "canonical_name": "p1/local/default/mnt1",
    "recordsize": 131072.0,
    "usage": {
        "available": 880395477504.0,
        "loading": false,
        "quota": 0.0,
        "snapshots": 18432.0,
        "compressratio": 100.0,
        "reservation": 0.0,
        "total": 50176.0,
        "data": 31744.0
    },
    "secondarycache": "all",
    "collection": "local",
    "exported": true,
    "vscan": false,
    "reservation": 0.0,
    "shadow": "none",
    "atime": true,
    "pool": "p1",
    "quota_snap": true,
    "name": "mnt1",
    "checksum": "fletcher4",
    "project": "default",
    "sharesftp": "",
    "nbmand": false,
    "reservation_snap": true,
    "sharedav": "",
    "rstchown": true,
    "root_acl": {
        "owner@:cc:fd:deny",
        "everyone@:rw:fd:allow",
        "user:admin1:rw:allow",
    }
    "smbshareacl": {
        "owner@:cc:fd:deny",
        "everyone@:rw:fd:allow",
        "user:admin1:rw:allow",
    }
}

```

파일 시스템 만들기

파일 시스템 만들기 명령은 지정된 스토리지 풀 또는 프로젝트에 있는 지정된 이름의 파일 시스템을 만듭니다. 기본 등록 정보가 지정된 새 파일 시스템이 반환됩니다.

다음 URI 매개변수가 사용됩니다.

<i>pool</i>	스토리지 풀 이름
<i>project</i>	프로젝트 이름
<i>filesystem</i>	파일 시스템 이름

요청 매개변수:

- *name* – 새 파일 시스템을 만들려면 파일 시스템 이름을 제공해야 합니다.
- 파일 시스템 등록 정보 – 파일 시스템 등록 정보 또는 프로젝트 등록 정보에 나열된 등록 정보를 초기 값으로 설정할 수 있습니다.

이름이 `share-01`이고 사용자 `admin1`이 소유한 파일 시스템을 만드는 요청의 예:

```
POST /api/storage/v1/pools/p1/projects/proj-01/filesystems HTTP/1.1
Host: zfs-storage.example.com:215
Content-Type: application/json
Accept: application/json

{
  "name": "share-01",
  "root_user": "admin1"
}
```

성공한 만들기는 새 파일 시스템의 URI를 포함하는 위치 헤더와 함께 HTTP 상태 201 (Created)을 반환합니다. 본문은 JSON 형식의 모든 파일 시스템 등록 정보를 포함합니다.

결과 예:

```
HTTP/1.1 201 Created
Content-Type: application/json
Location: /api/storage/v1/pools/p1/projects/proj-01/filesystems/share-01

{
  "filesystem": {
    "name": "share-01",
    "pool": "p1",
    "collection": "local",
    "project": "proj-01",
    "root_user": "admin1"
    ...
  }
}
```

파일 시스템 수정

파일 시스템 수정 명령은 기존 파일 시스템의 속성을 변경합니다. 성공한 응답은 HTTP 상태 202(Accepted)를 반환하고 모든 파일 시스템 등록 정보를 나열합니다.

다음 URI 매개변수가 사용됩니다.

<i>pool</i>	스토리지 풀 이름
<i>project</i>	프로젝트 이름
<i>filesystem</i>	파일 시스템 이름

요청 매개변수: 파일 시스템 등록 정보 – 파일 시스템 또는 프로젝트 등록 정보를 수정할 수 있습니다.

파일 시스템 이름을 `share-01`에서 `new-name`으로 변경하고 소유자를 `nobody`로 변경하는 요청의 예:

```
PUT /api/storage/v1/pools/p1/projects/proj-01/filesystems/share-01 HTTP/1.1
Host: zfs-storage.example.com:215
Content-Type: application/json
Accept: application/json
```

```
{
  "name": "new-name",
  "root_user": "nobody",
}
```

결과 예:

```
HTTP/1.1 202 Accepted
Content-Type: application/json
Location: http://zfs-storage.example.com:215/pools/p1/projects/proj-01/filesystems/share-01
```

```
{
  "filesystem": {
    "name": "new-name",
    "pool": "p1",
    "collection": "local",
    "project": "proj-01",
    "root_user": "nobody"
    ...
  }
}
```

파일 시스템 삭제

파일 시스템 삭제 명령은 지정된 풀 또는 프로젝트의 단일 파일 시스템을 제거합니다.

다음 URI 매개변수가 사용됩니다.

<i>pool</i>	스토리지 풀 이름
-------------	-----------

project 프로젝트 이름

filesystem 파일 시스템 이름

스토리지 풀에서 회수할 공간을 모니터하려면 `pools/pool`에 대한 GET 명령을 입력합니다. `async_destroy_reclaim_space` 등록 정보에 대한 공간을 기록해 두십시오. 작업이 완료되면 0이 표시됩니다.

요청 예:

```
DELETE /api/storage/v1/pools/p1/projects/proj-01/filesystems/share-01 HTTP/1.1
Host: zfs-storage.example.com:215
Accept: application/json
```

성공한 삭제는 HTTP 상태 204(No Content)를 반환합니다.

결과 예:

```
HTTP/1.1 204 No-Content
```

파일 시스템 쿼터 및 사용량

사용자 또는 그룹 쿼터는 각각 POST 또는 PUT 요청을 사용하여 만들거나 수정할 수 있습니다. 파일 시스템 사용 리소스에 대한 GET 요청은 프로젝트에 대한 사용자당 또는 그룹당 사용량 데이터를 가져오는 데 사용됩니다.

LUN 작업

모든 LUN 또는 볼륨 작업의 범위는 지정된 풀 또는 프로젝트로 지정됩니다. 다음 LUN 명령을 사용할 수 있습니다.

표 48 볼륨 명령

요청	경로 <code>/api/storage/v{1 2}</code> 에 추가	설명
GET	<code>/luns</code>	모든 LUN을 나열합니다.
GET	<code>/pools/pool/projects/project/luns</code>	LUN을 나열합니다.
GET	<code>/pools/pool/projects/project/luns?snaps=true</code>	스냅샷을 포함하여 모든 LUN을 나열합니다.
GET	<code>/pools/pool/projects/project/luns/lun</code>	LUN 세부정보를 가져옵니다.
POST	<code>/pools/pool/projects/project/luns</code>	LUN을 만듭니다.
PUT	<code>/pools/pool/projects/project/luns/lun</code>	LUN을 수정합니다.
DELETE	<code>/pools/pool/projects/project/luns/lun</code>	LUN을 삭제합니다.

다음 표에는 LUN 등록 정보가 나와 있습니다. 또한 볼륨은 프로젝트 등록 정보를 상속하거나 대체할 수 있습니다.

표 49 볼륨 등록 정보

등록 정보	유형	설명
assignednumber	number 또는 list of numbers	지정된 LU 번호 다중 개시자 그룹에 제공된 경우 유형은 list of numbers입니다. 다중 개시자 그룹에 제공된 경우 assignednumber 및 initiatorgroups의 순서가 조정됩니다. 예를 들어, assignednumber 목록의 첫번째 항목은 initiatorgroups 목록의 첫번째 항목과 관련이 있습니다.
fixednumber	boolean	LU 번호를 현재 값에 고정하는 플래그.
initiatorgroups	list of strings	개시자 그룹입니다. LUN이 다중 개시자 그룹에 제공된 경우 assignednumber 및 initiatorgroups의 순서가 조정됩니다. 예를 들어, assignednumber 목록의 첫번째 항목은 initiatorgroups 목록의 첫번째 항목과 관련이 있습니다.
lunGUID	string	STMF GUID입니다.
lunumber	number 또는 string	LU 번호 number 또는 auto입니다.
project	string	프로젝트 이름(변경할 수 없음).
source	object	등록 정보의 소스(local 또는 inherited)를 나열합니다.
sparse	boolean	Thin 프로비저닝을 사용으로 설정하는 플래그.
status	string	논리 장치 상태(online 또는 offline)입니다.
targetgroup	string	대상 그룹
usage	object	LUN 사용량 통계 나열
volblocksize	number	볼륨 블록 크기
volsize	number	볼륨 크기
writocache	boolean	쓰기 캐시를 사용으로 설정하는 플래그

프로젝트에서 일부 등록 정보가 상속될 수 있습니다. 소스 객체는 이러한 등록 정보를 각각 나열하고 해당 등록 정보가 LUN에 로컬인지 아니면 프로젝트에서 상속되는지를 식별합니다. 기본적으로 이러한 등록 정보는 프로젝트에 의해 상속됩니다. 이러한 등록 정보가 설정되면 해당 등록 정보가 LUN에 로컬입니다. 소스 객체는 변경할 수 없습니다. 소스를 다시 상속됨으로 변경하려는 경우 등록 정보를 설정 해제할 수 있습니다.

압축을 설정 해제하는 JSON 요청의 예:

```
{"unset": ["compression"]}
```

LUN을 나열합니다.

LUN 나열 명령은 지정된 풀 또는 프로젝트에서 사용할 수 있는 LUN 목록을 반환합니다.

주 - depth 질의 매개변수와 match_property-name=value 질의 매개변수는 지원되지 않습니다.

다음 URI 매개변수가 사용됩니다.

<i>pool</i>	스토리지 풀 이름
<i>project</i>	프로젝트 이름
<i>filesystem</i>	파일 시스템 이름

프로젝트 `proj-01` 내에서 LUN을 나열하는 요청의 예:

```
GET /api/storage/v1/pools/p1/projects/proj-01/luns HTTP/1.1
Host: zfs-storage.example.com:215
Accept: application/json
```

성공한 `get`은 JSON 형식의 LUN 등록 정보와 함께 HTTP 상태 200(OK)을 반환합니다.

결과 예:

```
HTTP/1.1 200 OK
Content-Type: application/json
{
  "luns": [{
    "id": "fa4ac6fb-0bcc-d2e3-0000-000000000000",
    "name": "vol-01"
    ...
  }, {
    "id": "690ae407-7c4d-b5d2-0000-000000000000",
    "name": "vol-01",
    ....
  }
]
```

LUN 가져오기

LUN 가져오기 명령은 지정된 풀 또는 프로젝트의 단일 LUN 등록 정보를 반환합니다.

다음 URI 매개변수가 사용됩니다.

<i>pool</i>	스토리지 풀 이름
<i>project</i>	프로젝트 이름
<i>lun</i>	LUN 이름

이름이 "vol-01"인 LUN을 가져오는 요청의 예:

```
GET /api/storage/v1/pools/p1/projects/proj-01/lun/vol-01 HTTP/1.1
Host: zfs-storage.example.com:215
Accept: application/json
```

성공한 `get`은 JSON 형식의 LUN 등록 정보와 함께 HTTP 상태 200(OK)을 반환합니다.

결과 예:

```

HTTP/1.1 200 OK
Content-Type: application/json

{
  "lun": {
    "logbias": "latency",
    "creation": "20130423T21:31:17",
    "nodestroy": false,
    "dedup": false,
    "rrsrc_actions": [],
    "id": "e3045406-319b-cf7a-0000-000000000000",
    "writecache": false,
    "compression": "off",
    "copies": 1,
    "stmfguid": "600144F0D8E0AE4100005176FDA60001",
    "source": {
      "compression": "default",
      "checksum": "inherited",
      "logbias": "default",
      "dedup": "default",
      "copies": "default",
      "exported": "inherited",
      "rrsrc_actions": "inherited",
      "secondarycache": "default"
    },
    "canonical_name": "p1/local/default/disk1",
    "snaplabel": "",
    "usage": {
      "available": 881469214720.0,
      "loading": false,
      "snapshots": 0.0,
      "compressratio": 100.0,
      "total": 1073758208.0,
      "data": 1073758208.0
    },
    "secondarycache": "all",
    "collection": "local",
    "exported": true,
    "volsize": 1073741824.0,
    "pool": "p1",
    "volblocksize": 8192,
    "checksum": "fletcher4",
    "project": "default",
    "sparse": false
  }
}

```

새 LUN 만들기

이 명령은 새 LUN을 만듭니다. 새 LUN에 대한 크기 또는 복제 소스를 제공해야 합니다.

다음 URI 매개변수가 사용됩니다.

pool 스토리지 풀 이름

project 프로젝트 이름

요청 매개변수:

- `name` – 새 LUN을 만들려면 새 LUN 이름을 제공해야 합니다.
- 볼륨 등록 정보 – LUN 등록 정보 또는 프로젝트 등록 정보에 나열된 등록 정보를 초기 값으로 설정할 수 있습니다.

요청 예:

```
POST /api/storage/v1/pools/p1/projects/proj-01/luns HTTP/1.1
Host: zfs-storage.example.com:215
Accept: application/json
```

Request JSON:

```
{
  name : "vol-001",           // Volume name (required)

  size : 500000,             // New Volume size
  blocksize : 8192,         // New Volume block size
  sparse : true,             // New Volume sparse data flag

  initiatorgroup : 'default', // Initiator group name
  targetgroup : 'default',    // Target group name
  lunumber : 'auto',         // Volume LU number
  status : 'online',         // Initial Status ('online', 'offline')
  fixednumber : false,

  "source": {
    "snapshot_id" : "76b8950a-8594-4e5b-8dce-0dfa9c696358",
    "snapshot": "/pool-001/local/proj-001/snap-001"
  }
}
```

성공한 만들기는 새 LUN의 URI를 포함하는 위치 헤더와 함께 HTTP 상태 201(Created)을 반환합니다. 본문은 JSON 형식의 모든 LUN 등록 정보를 포함합니다.

결과 예:

```
HTTP/1.1 201 Created
Content-Type: application/json
Location: http://zfs-storage.example.com:215
          /pools/p1/projects/proj-01/luns/vol-001
```

```
{
  "lun": {
    "name": "vol-001",
    ...
  }
}
```

LUN 수정

LUN 수정 명령은 기존 LUN의 속성을 변경합니다.

다음 URI 매개변수가 사용됩니다.

pool 스토리지 풀 이름

project 프로젝트 이름

lun LUN 이름

요청 매개변수: 볼륨 등록 정보 – LUN 또는 프로젝트 등록 정보를 수정할 수 있습니다.

LUN 이름을 *vol-01*에서 *new-name*으로 변경하는 요청의 예:

```
POST /api/storage/v1/pools/p1/projects/proj-01/luns/vol-01 HTTP/1.1
Host: zfs-storage.example.com:215
Content-Type: application/json
Accept: application/json
```

```
{
  "name": "new-name",
}
```

성공한 응답은 HTTP 상태 202(Accepted)를 반환하고 모든 LUN 등록 정보를 나열합니다.

결과 예:

```
HTTP/1.1 201 Created
Content-Type: application/json
Location: /api/storage/v1/pools/p1/projects/proj-01/luns/new-name
```

```
{
  "lun": {
    "name": "new-name",
    "pool": "p1",
    "collection": "local",
    "project": "proj-01",
    ...
  }
}
```

LUN 삭제

LUN 삭제 명령은 지정된 풀 또는 프로젝트의 단일 LUN을 제거합니다.

다음 URI 매개변수가 사용됩니다.

pool 스토리지 풀 이름

project 프로젝트 이름

lun LUN 이름

스토리지 풀에서 회수할 공간을 모니터링하려면 *pools/pool*에 대한 GET 명령을 입력합니다. *async_destroy_reclaim_space* 등록 정보에 대한 공간을 기록해 두십시오. 작업이 완료되면 0이 표시됩니다.

요청 예:

```
DELETE /pools/p1/projects/proj-01/luns/lun-01 HTTP/1.1
Host: zfs-storage.example.com:215
Accept: application/json
```

성공한 `get`은 HTTP 상태 204(No Content)를 반환합니다.

결과 예:

HTTP/1.1 204 No-Content

스냅샷 및 복제 작업

모든 스냅샷 작업의 범위는 지정된 풀 또는 프로젝트로 지정됩니다. 또한 스냅샷 작업의 범위를 파일 시스템 또는 LUN 레벨로 설정할 수 있습니다.

- 모든 프로젝트 기반 스냅샷 작업에 대한 URI는 `/api/storage/v{1|2}/pools/pool/projects/project`로 시작합니다.
- 모든 파일 시스템 기반 스냅샷 작업에 대한 URI는 `/api/storage/v{1|2}/pools/pool/projects/project/filesystems/filesystem`으로 시작합니다.
- 모든 LUN 기반 스냅샷 작업에 대한 URI는 `/api/storage/v{1|2}/pools/pool/projects/project/luns/lun`으로 시작합니다.

클라우드에 스냅샷을 백업하거나 어플라이언스에 새 공유로 스냅샷 백업을 복원하려면 [RESTful API 클라우드 서비스 \[105\]](#)를 참조하십시오.

표 50 스냅샷 및 복제 명령

요청	경로 <code>/api/storage/v{1 2}</code> 에 추가	설명
GET	<code>/snapshots</code>	모든 로컬 스냅샷을 나열합니다.
GET	<code>/pools/pool/projects?snaps=true</code>	스냅샷을 포함하여 모든 프로젝트를 나열합니다.
GET	<code>/pools/pool/projects/project/filesystems?snaps=true</code>	스냅샷을 포함하여 모든 파일 시스템을 나열합니다.
GET	<code>/pools/pool/projects/project/luns?snaps=true</code>	스냅샷을 포함하여 모든 LUN을 나열합니다.
GET	<code>/pools/pool/projects/project/snapshots</code>	프로젝트에 대한 모든 스냅샷을 나열합니다.
GET	<code>/pools/pool/projects/project/filesystems/filesystem/snapshots</code>	파일 시스템에 대한 모든 스냅샷을 나열합니다.
GET	<code>/pools/pool/projects/project/luns/lun/snapshots</code>	LUN에 대한 모든 스냅샷을 나열합니다.
GET	<code>/pools/pool/projects/project/snapshots/snapshot</code>	프로젝트 스냅샷 세부정보를 가져옵니다.
GET	<code>/pools/pool/projects/project/filesystems/filesystem/snapshots/snapshot</code>	파일 시스템 스냅샷 세부정보를 가져옵니다.
GET	<code>/pools/pool/projects/project/luns/lun/snapshots/snapshot</code>	LUN 스냅샷 세부정보를 가져옵니다.
POST	<code>/pools/pool/projects/project/snapshots</code>	프로젝트 스냅샷을 만듭니다.
POST	<code>/pools/pool/projects/project/filesystems/filesystem/snapshots</code>	파일 시스템 스냅샷을 만듭니다.

요청	경로 <code>/api/storage/v{1 2}</code> 에 추가	설명
POST	<code>/pools/pool/projects/project/luns/lun/snapshots</code>	LUN 스냅샷을 만듭니다.
PUT	<code>/pools/pool/projects/project/snapshots/snapshot</code>	프로젝트 스냅샷을 수정합니다.
PUT	<code>/pools/pool/projects/project/filesystems/filesystem/snapshots/snapshot</code>	파일 시스템 스냅샷을 수정합니다.
PUT	<code>/pools/pool/projects/project/luns/lun/snapshots/snapshot</code>	LUN 스냅샷을 수정합니다.
PUT	<code>/pools/pool/projects/project/filesystems/filesystem/snapshots/snapshot/clone</code>	파일 시스템 스냅샷을 복제합니다.
PUT	<code>/pools/pool/projects/project/luns/lun/snapshots/snapshot/clone</code>	LUN 스냅샷을 복제합니다.
PUT	<code>/pools/pool/projects/project/filesystems/filesystem/snapshots/snapshot/rollback</code>	데이터를 지정된 파일 시스템 스냅샷으로 롤백합니다.
PUT	<code>/pools/pool/projects/project/lun/lun/snapshots/snapshot/rollback</code>	데이터를 지정된 LUN 스냅샷으로 롤백합니다.
DELETE	<code>/pools/pool/projects/project/snapshots/snapshot</code>	프로젝트 스냅샷을 삭제합니다.
DELETE	<code>/pools/pool/projects/project/filesystems/filesystem/snapshots/snapshot</code>	파일 시스템 스냅샷을 삭제합니다.
DELETE	<code>/pools/pool/projects/project/luns/lun/snapshots/snapshot</code>	LUN 스냅샷을 삭제합니다.
GET	<code>/pools/pool/projects/project/snapshots/snapshot/dependents</code>	프로젝트 스냅샷 종속 항목을 나열합니다.
GET	<code>/pools/pool/projects/project/filesystems/filesystem/snapshots/snapshot/dependents</code>	파일 시스템 스냅샷 종속 항목을 나열합니다.
GET	<code>/pools/pool/projects/project/lun/lun/snapshots/snapshot/dependents</code>	LUN 스냅샷 종속 항목을 나열합니다.
POST	<code>/pools/pool/projects/project/automatic</code>	새 프로젝트 자동 스냅샷 객체를 만듭니다.
POST	<code>/pools/pool/projects/project/automatic?convert=true</code>	새 프로젝트 자동 스냅샷 객체를 만듭니다. 새 일정에 맞지 않는 자동 생성된 기존 스냅샷이 수동 스냅샷으로 변환됩니다. convert 등록 정보를 제외하면 자동 생성된 기존 스냅샷이 삭제됩니다.
GET	<code>/pools/pool/projects/project/automatic/automatic</code>	지정된 프로젝트 자동 스냅샷 등록 정보를 가져옵니다.
GET	<code>/pools/pool/projects/project/automatic</code>	모든 프로젝트 자동 스냅샷 객체를 나열합니다.
PUT	<code>/pools/pool/projects/project/automatic/automatic</code>	지정된 프로젝트 자동 스냅샷 객체를 수정합니다.
PUT	<code>/pools/pool/projects/project/automatic/automatic?convert=true</code>	지정된 프로젝트 자동 스냅샷 일정 객체를 수정합니다. 새 일정에 맞지 않는 자동 생성된 기존 스냅샷이 수동 스냅샷으로 변환됩니다. convert 등록 정보를 제외하면 자동 생성된 기존 스냅샷이 삭제됩니다.
DELETE	<code>/pools/pool/projects/project/automatic/automatic</code>	지정된 자동 객체를 삭제합니다.
DELETE	<code>/pools/pool/projects/project/automatic/automatic?convert=true</code>	지정된 자동 스냅샷 일정 객체를 삭제합니다. 새 일정에 맞지 않는 자동 생성된 기존 스냅샷이 수동 스냅샷으로 변환됩니다.

스냅샷 및 복제 작업

요청	경로 <i>/api/storage/v{1 2}</i> 에 추가	설명
		convert 등록 정보를 제외하면 자동 생성된 기존 스냅샷이 삭제됩니다.
POST	<i>/pools/pool/projects/project/filesystems/filesystem/automatic</i>	새 파일 시스템 자동 스냅샷 객체를 만듭니다.
POST	<i>/pools/pool/projects/project/filesystems/filesystem/automatic?convert=true</i>	새 파일 시스템 자동 스냅샷 객체를 만듭니다. 새 일정에 맞지 않는 자동 생성된 기존 스냅샷이 수동 스냅샷으로 변환됩니다. convert 등록 정보를 제외하면 자동 생성된 기존 스냅샷이 삭제됩니다.
GET	<i>/pools/pool/projects/project/filesystems/filesystem/automatic/automatic</i>	지정된 파일 시스템 자동 스냅샷 등록 정보를 가져옵니다.
GET	<i>/pools/pool/projects/project/filesystems/filesystem/automatic</i>	모든 파일 시스템 자동 스냅샷 객체를 나열합니다.
PUT	<i>/pools/pool/projects/project/filesystems/filesystem/automatic/automatic</i>	지정된 파일 시스템 자동 스냅샷 객체를 수정합니다.
PUT	<i>/pools/pool/projects/project/filesystems/filesystem/automatic/automatic?convert=true</i>	지정된 파일 시스템 자동 스냅샷 일정 객체를 수정합니다. 새 일정에 맞지 않는 자동 생성된 기존 스냅샷이 수동 스냅샷으로 변환됩니다. convert 등록 정보를 제외하면 자동 생성된 기존 스냅샷이 삭제됩니다.
DELETE	<i>/pools/pool/projects/project/filesystems/filesystem/automatic/automatic</i>	지정된 자동 스냅샷 일정 객체를 삭제합니다.
DELETE	<i>/pools/pool/projects/project/filesystems/filesystem/automatic/automatic?convert=true</i>	지정된 파일 시스템 자동 스냅샷 일정 객체를 삭제합니다. 새 일정에 맞지 않는 자동 생성된 기존 스냅샷이 수동 스냅샷으로 변환됩니다. convert 등록 정보를 제외하면 자동 생성된 기존 스냅샷이 삭제됩니다.
POST	<i>/pools/pool/projects/project/luns/lun/automatic</i>	새 LUN 자동 스냅샷을 만듭니다.
POST	<i>/pools/pool/projects/project/luns/lun/automatic?convert=true</i>	새 LUN 자동 스냅샷 일정을 만듭니다. 새 일정에 맞지 않는 자동 생성된 기존 스냅샷이 수동 스냅샷으로 변환됩니다. convert 등록 정보를 제외하면 자동 생성된 기존 스냅샷이 삭제됩니다.
GET	<i>/pools/pool/projects/project/luns/lun/automatic/automatic</i>	지정된 LUN 자동 스냅샷 등록 정보를 가져옵니다.
GET	<i>/pools/pool/projects/project/luns/lun/automatic</i>	모든 LUN 자동 스냅샷 객체를 나열합니다.
PUT	<i>/pools/pool/projects/project/luns/lun/automatic/automatic</i>	지정된 LUN 자동 스냅샷 객체를 수정합니다.
PUT	<i>/pools/pool/projects/project/luns/lun/automatic/automatic?convert=true</i>	지정된 LUN 자동 스냅샷 일정 객체를 수정합니다. 새 일정에 맞지 않는 자동

요청	경로 <code>/api/storage/v{1 2}</code> 에 추가	설명
		생성된 기존 스냅샷이 수동 스냅샷으로 변환됩니다. convert 등록 정보를 제외하면 자동 생성된 기존 스냅샷이 삭제됩니다.
DELETE	<code>/pools/pool/projects/project/luns/lun/automatic/automatic</code>	지정된 LUN 자동 객체를 삭제합니다.
DELETE	<code>/pools/pool/projects/project/luns/lun/automatic/automatic?convert=true</code>	지정된 LUN 자동 스냅샷 일정 객체를 삭제합니다. 새 일정에 맞지 않는 자동 생성된 기존 스냅샷이 수동 스냅샷으로 변환됩니다. convert 등록 정보를 제외하면 자동 생성된 기존 스냅샷이 삭제됩니다.

스냅샷 나열

어플라이언스에서 사용할 수 있는 어플라이언스를 나열합니다. 요청 URI에 따라 목록은 프로젝트, 파일 시스템 또는 LUN 스냅샷을 포함합니다.

표 51 스냅샷 나열 명령 형식

명령	경로 <code>/api/storage/v{1 2}/pools/pool/projects/project</code> 에 추가
프로젝트 스냅샷 나열	<code>/snapshots</code>
파일 시스템 스냅샷 나열	<code>/filesystems/share/snapshots</code>
LUN 스냅샷 나열	<code>/lun/share/snapshots</code>

요청 예:

```
GET /api/storage/v1/pools/p1/projects/default/snapshots
Accept: application/json
```

결과 예:

```
HTTP/1.1 200 OK
Content-Type: application/json
```

```
{
  "snapshots": [{
    "id": "3fbbcccf-d058-4502-8844-6feeffdf4cb5",
    "display_name": "snap-001",
    "display_description": "Daily backup",
    "volume_id": "521752a6-acf6-4b2d-bc7a-119f9148cd8c",
    "status": "available",
    "size": 30,
    "created_at": "2012-02-29T03:50:07Z"
  }, {
    "id": "e479997c-650b-40a4-9dfe-77655818b0d2",
    "display_name": "snap-002",
    "display_description": "Weekly backup",
```

```

        "volume_id": "76b8950a-8594-4e5b-8dce-0dfa9c696358",
        "status": "available",
        "size": 25,
        "created_at": "2012-03-19T01:52:47Z"
    }
}

```

스냅샷 가져오기

단일 스냅샷에 대한 모든 정보를 봅니다. 성공 시 HTTP 상태 200(OK)을 반환합니다.

요청 예:

```

GET /api/storage/v1/pools/p1/projects/default/snapshots/snap-001
Accept: application/json

```

결과 예:

```

HTTP/1.1 200 OK
Content-Type: application/json

{
  "snapshot": {
    "id": "3fbbcccf-d058-4502-8844-6feeffdf4cb5",
    "display_name": "snap-001",
    "display_description": "Daily backup",
    "volume_id": "521752a6-acf6-4b2d-bc7a-119f9148cd8c",
    "status": "available",
    "size": 30,
    "created_at": "2012-02-29T03:50:07Z"
  }
}

```

스냅샷 만들기

스냅샷 만들기 명령은 프로젝트, 파일 시스템 또는 LUN에 대한 스냅샷을 만듭니다.

- 프로젝트 스냅샷 만들기 – POST /pools/pool/projects/project/snapshots
- 파일 시스템 스냅샷 만들기 – POST /pools/pool/projects/project/filesystems/share/snapshots
- 볼륨 스냅샷 만들기 – POST /pools/pool/projects/project/luns/lun/snapshots

요청 예:

```

POST /api/storage/v1/pools/p1/projects/default/snapshots
Content-Type: application/json

```

```

{"name": "initial-backup"}

```

결과 예:

```

HTTP/1.1 201 Created

```

```
Content-Type: application/json
Location: /pools/p1/projects/default/
snapshot/initial-backup
```

```
{
  "snapshot": {
    "name": "initial-backup",
    "numclones": 0,
    "creation": "20130610T21:00:49",
    "collection": "local",
    "project": "default",
    "canonical_name": "zfs-storage-1/local/default@initial-backup",
    "usage": {
      "unique": 0.0,
      "loading": false,
      "data": 145408.0
    },
    "type": "snapshot",
    "id": "a26abd24-e22b-62b2-0000-000000000000",
    "pool": "p1"
  }
}
```

스냅샷 이름 바꾸기

기존 스냅샷의 이름을 바꿉니다.

- 요청 URI – 스냅샷, 현재 스냅샷 이름
- 요청 본문 – 새 스냅샷 이름을 포함하는 name 매개변수의 JSON 객체

요청 예:

```
PUT /api/storage/v1/pools/p1/projects/default/snapshots/initial-snapshot
Content-Type: application/json
Accept: application/json

{"name": "old-snapshot"}
```

결과 예:

```
HTTP/1.1 202 Accepted
Content-Type: application/json
Location: /pools/p1/projects/default/snapshot/initial-backup
```

스냅샷 복제

기존 스냅샷에서 새 파일 시스템 또는 LUN을 만듭니다.

다음 URI 매개변수가 사용됩니다.

pool 소스 풀 이름

project 소스 프로젝트 이름

filesystem 파일 시스템 스냅샷에 대한 소스 공유 이름

lun LUN 스냅샷에 대한 소스 공유 이름

snapshot 소스 스냅샷 이름

파일 시스템 복제:

PUT /pools/pool/projects/project/filesystems/share/snapshots/snapshot/clone

볼륨 복제:

PUT /pools/pool/projects/project/luns/lun/snapshots/snapshot/clone

요청 본문은 다음 등록 정보가 있는 JSON 객체를 포함합니다.

표 52 복제 스냅샷 등록 정보

등록 정보	유형	설명
pool	string	대상 복제 풀 이름
project	string	대상 복제 프로젝트 이름
lun	string	LUN 스냅샷에 대한 대상 LUN 이름

요청 예:

```
PUT /api/storage/v1/pools/p1/projects/default/filesystems/fs01/
snapshots/snap01/clone

{"project": "rest", "share": "snap01clone01", "compression": "gzip-9"}
```

결과 예:

```
HTTP/1.1 201 Created
Content-Length: 2035
X-Zfssa-Storage-API: 1.0
Location: /api/storage/v1/pools/p1/projects/rest/filesystem/snap01clone01
Content-Type: application/json; charset=utf-8

{
  "filesystem": {
    "origin": {
      "project": "default",
      "share": "fs01",
      "snapshot": "snap01",
      "pool": "p1",
      "collection": "local"
    },
    "href": "/api/storage/v1/pools/p1/projects/rest/filesystems/snap01clone01",
    "mountpoint": "/export/snap01clone01",
    "compression": "gzip-9",
    "source": {
```

```

        "compression": "local",
        ...
    },
    ...
    "canonical_name": "zfs-storage-1/local/rest/snap01clone01"
}
}

```

스냅샷 롤백

스냅샷을 롤백하면 스냅샷이 사용되었을 때 소스 파일 시스템 또는 LUN이 해당 상태로 다시 수정됩니다. 성공한 응답은 HTTP 상태 202(Accepted) 및 JSON 형식의 스냅샷 등록 정보를 반환합니다.

다음 URI 매개변수가 사용됩니다.

<i>pool</i>	소스 풀 이름
<i>project</i>	소스 프로젝트 이름
<i>filesystem</i>	파일 시스템 스냅샷에 대한 소스 파일 시스템 이름
<i>lun</i>	LUN 스냅샷에 대한 소스 LUN 이름
<i>snapshot</i>	소스 스냅샷 이름

파일 시스템 스냅샷 롤백:

```
PUT /pools/pool/projects/project/filesystems/filesystem/snapshots/snapshot/rollback
```

LUN 스냅샷 롤백:

```
PUT /pools/pool/projects/project/luns/lun/snapshots/snapshot/rollback
```

요청 예:

```
PUT /api/storage/v1/pools/p1/projects/default/filesystems/fs-01
    /snapshots/initial-backup/rollback
```

결과 예:

```
HTTP/1.1 202 Accepted
Location: /pools/p1/projects/default/filesystems/fs-01/snapshot/fs-01-initial-clone
Content-Type: application/json
```

```
{
  "snapshot": {
    "name": "fs-01-initial-clone",
    "numclones": 0,
    "creation": "20130610T21:00:49",
    "filesystem": "fs-01",

```

```

    "collection": "local",
    "project": "default",
    "canonical_name": "zfs-storage-1/local/default/
      fs-01@fs-01-initial-clone",
    "usage": {
      "unique": 0.0,
      "loading": false,
      "data": 31744.0
    },
    "type": "snapshot",
    "id": "5c9bda07-21c1-2238-0000-000000000000",
    "pool": "p1"
  }
}

```

스냅샷 삭제

스냅샷 `DELETE` 명령은 시스템에서 프로젝트, 파일 시스템 또는 LUN 스냅샷을 삭제합니다.

다음 URI 매개변수가 사용됩니다.

<i>pool</i>	소스 풀 이름
<i>project</i>	소스 프로젝트 이름
<i>filesystem</i>	소스 파일 시스템 이름
<i>lun</i>	LUN 이름
<i>snapshot</i>	소스 스냅샷 이름

프로젝트 스냅샷 삭제:

```
DELETE /api/storage/v1/pools/pool/projects/project/snapshots/snapshot
```

파일 시스템 스냅샷 삭제:

```
DELETE /api/storage/v1/pools/pool/projects/project/filesystems/filesystem/snapshots/snapshot
```

파일 시스템 LUN 삭제:

```
DELETE /api/storage/v1/pools/pool/projects/projectsnapshot
```

스냅샷에 NDMP 보류가 있는 경우 `?confirm=true`를 `DELETE` 명령에 추가합니다. 하지만 이렇게 하면 NDMP 작업에 부정적인 영향이 미칠 수 있습니다. 자세한 내용은 [“NDMP Configuration” in Oracle ZFS Storage Appliance Administration Guide, Release OS8.8.x](#)을 참조하십시오.

요청 예:

```
DELETE /pools/p1/projects/default/filesystems/fs-01/snapshots/initial-backup?confirm=true
```

?confirm=true가 추가되지 않은 경우 결과 예:

스냅샷에 NDMP 보류가 있을 때 ?confirm=true가 추가되지 않은 경우 다음 출력과 함께 명령이 실패합니다(가독성을 위해 인위적으로 줄바꿈이 설정됨).

```
HTTP/1.1 409 Conflict
```

```
{"fault": {"message": "request requires confirm=true to complete (confirmation
needed for scripted command (scripted commands must be prefixed with \"confirm\"
to automatically confirm or \"deny\" to automatically deny) (encountered while
attempting to run command \"confirm destroy snap\"))", "code": 409, "name":
"ERR_CONFIRM_REQUIRED"}}
```

스냅샷 종속 항목 나열

파일 시스템 또는 볼륨에 대한 종속 항목을 나열합니다. 다음 URI 매개변수가 사용됩니다.

pool 시스템 스토리지 풀 이름

project 프로젝트 이름

filesystem 파일 시스템 이름

lun LUN 이름

snapshot 스냅샷 이름

파일 시스템 종속 항목을 나열합니다.

```
GET /api/storage/v1/
pools/pool/projects/project/filesystems/filesystem/snapshots/snapshot/dependents
```

볼륨 종속 항목을 나열합니다.

```
GET /api/storage/v1/pools/pool/projects/project/lun/lun/snapshots/snapshot/dependents
```

요청 예:

```
GET /api/storage/v1/pools/p1/projects/default/filesystems/fs01/snapshots/snap01/dependents
Accept: application/json
```

결과 예:

```
HTTP/1.1 200 OK
X-Zfssa-Storage-API: 1.0
Content-Type: application/json; charset=utf-8
X-Zfssa-API-Version: 1.0
```

```
{
```

```

    "depends": [
      {
        "project": "rest",
        "href": "/api/storage/v1/pools/p1/projects/rest/filesystems/snap01clone01",
        "share": "snap01clone01"
      },
      {
        "project": "rest",
        "href": "/api/storage/v1/pools/p1/projects/rest/filesystems/snap01clone02",
        "share": "snap01clone02"
      },
      {
        "project": "rest",
        "href": "/api/storage/v1/pools/p1/projects/rest/filesystems/snap01clone03",
        "share": "snap01clone03"
      }
    ]
  }
}

```

스키마

사용자정의 스키마 등록 정보를 관리합니다.

표 53 스키마 명령

요청	경로 /api/storage/v{1 2}에 추가	설명
GET	/schema	모든 NAS 스키마 등록 정보 객체를 나열합니다.
GET	/schema/property	지정된 NAS 스키마 등록 정보 등록 정보를 가져옵니다.
POST	/schema	새 NAS 스키마 등록 정보를 만듭니다.
PUT	/schema/property	지정된 NAS 스키마 등록 정보 객체를 수정합니다.
DELETE	/schema/property	지정된 NAS 스키마 등록 정보 객체를 삭제합니다.

사용자정의 등록 정보 이름에 접두어 `custom:`을 추가하여 프로젝트, 파일 시스템 및 LUN에서 각 사용자정의 스키마 등록 정보를 설정할 수 있습니다.

예를 들어, 다음 PUT 본문은 이름이 `priority`인 사용자정의 `int` 등록 정보를 수정합니다.

```
{"custom:priority": 5}
```

표 54 스키마 매개변수

매개변수	설명
property	등록 정보의 이름(변경할 수 없음)
description	등록 정보 설명(브라우저 인터페이스용)
type	유형("String", "Integer", "PositiveInteger", "Boolean", "EmailAddress", "Host")

등록 정보 나열

스키마 등록 정보를 나열합니다.

요청 예:

```
GET /api/storage/v1/schema
```

결과 예:

```
{
  "properties": [{
    "description": "bob",
    "href": "/api/storage/v1/schema/bob",
    "property": "bob",
    "type": "String"
  }, {
    "description": "boo",
    "href": "/api/storage/v1/schema/boo",
    "property": "boo",
    "type": "String"
  }]
}
```

등록 정보 가져오기

스키마 등록 정보를 가져옵니다.

요청 예:

```
GET /api/storage/v1/schema/priority
```

결과 예:

```
{
  "property": {
    "description": "priority",
    "href": "/api/storage/v1/schema/priority",
    "property": "bob",
    "type": "Integer"
  }
}
```

등록 정보 만들기

새 스키마 등록 정보를 만듭니다.

요청 예:

```
POST /api/storage/v1/schema HTTP/1.1
```

```
Host: zfs-storage.example.com:215
Content-Type: application/json
Content-Length: 64
```

```
{"property": "priority", "type": "Integer", "description": "Oh my"}
```

결과 예:

```
HTTP/1.1 201 Created
Content-Length: 89
X-Zfssa-Nas-API: 1.0
Content-Type: application/json
Location: /api/storage/v1/schema/priority
```

```
{
  "property": {
    "href": "/api/storage/v1/schema",
    "type": "Integer",
    "description": "Oh my"
  }
}
```

등록 정보 수정

스키마 등록 정보를 수정합니다.

요청 예:

```
PUT /api/storage/v1/schema/priority
{"description": "My custom priority level"}
```

결과 예:

```
HTTP/1.1 202 Accepted
X-Zfssa-Nas-API: 1.0
Content-Type: application/json
Content-Length: 90

{
  "property": {
    "href": "//api/storage/v1/schema/priority",
    "type": "Integer",
    "description": "My custom priority level"
  }
}
```

등록 정보 삭제

이 명령은 스키마 등록 정보를 삭제합니다.

요청 예:

DELETE /api/storage/v1/schema/me HTTP/1.1

결과 예:

HTTP/1.1 204 No Content

복제

원격 복제를 사용하면 프로젝트 복제 및 어플라이언스 간 공유가 쉬워집니다.

주 - 복제는 Oracle ZFS 스토리지 어플라이언스의 특정 모델에 대해 사용이 허가된 기능이며 복제 RESTful API가 해당 기능을 관리합니다. 서비스는 URI `https://hostname:215/api/storage/v{1|2}/replication`에서 사용할 수 있습니다. 라이선스에 대한 자세한 내용은 Oracle SLA(소프트웨어 라이선스 계약) 및 통합된 소프트웨어 옵션을 사용하는 하드웨어 시스템에 대한 자격 및 소프트웨어 릴리스의 라이선스 정보 사용자 설명서를 참조하십시오.

복제 RESTful API는 다음 리소스를 관리합니다:

- 복제 서비스 – 복제 작업을 관리하는 서비스입니다.
- 복제 대상 – 다른 어플라이언스 피어(소스)에서 복제된 데이터를 받고 저장하는 어플라이언스 피어입니다. 이 용어는 어플라이언스를 다른 어플라이언스에 복제할 수 있도록 하는 어플라이언스의 구성 객체를 의미하기도 합니다.
- 복제 작업 – 프로젝트 또는 공유, 대상 어플라이언스 및 정책 옵션(업데이트 보내기 빈도, 데이터를 전송 중에 암호화할지 여부 등 포함)을 지정하는 소스 어플라이언스에 대한 구성 객체입니다.
- 복제 패키지 – 대상측에서 작업에 해당되는 개념입니다. 특정 소스에서 특정 작업의 일부분으로 복제된 데이터를 관리하는 대상 어플라이언스에 대한 구성 객체입니다. 소스 어플라이언스의 각 작업은 대상 어플라이언스에서 정확하게 하나의 패키지와 연관되며 그 반대의 경우도 마찬가지입니다. 둘 중 한 객체가 손실되면 새 작업/패키지 쌍 및 전체 복제 업데이트를 만들어야 합니다.

API는 복제 작업 및 복제 패키지에 대한 복제 작업을 제공합니다. 서비스 API는 복제 서비스, 복제 소스 및 복제 대상을 관리하는 데 사용됩니다.

표 55 복제 서비스 명령

요청	경로 /api/service/v{1 2}/services에 추가	설명
GET	/replication	복제 서비스 상태 등록 정보를 가져옵니다.
PUT	/replication/enable	복제 서비스를 사용으로 설정합니다.
PUT	/replication/disable	복제 서비스를 사용 안함으로 설정합니다.

복제 서비스 등록 정보 나열

복제 서비스의 상태를 가져옵니다.

요청 예:

```
GET /api/service/v2/services/replication HTTP/1.1
Host: zfs-storage.example.com:215
Accept: application/json
```

결과 예:

```
HTTP/1.1 200 OK

{
  "service": {
    "href": "/api/service/v2/services/replication",
    "<status>": "online",
    "enable_start_finish_alerts": true
  }
}
```

복제 서비스 등록 정보 수정

다른 서비스와 마찬가지로 복제 서비스 상태도 수정할 수 있습니다. 자세한 내용은 서비스 RESTful API를 참조하십시오.

복제 중인 프로젝트 수 및 복제 일정 빈도에 따라 예약된 업데이트에 대한 시작 및 완료 경보 수로 인해 중요한 다른 경보가 숨겨질 수 있습니다. 예약된 업데이트에 대한 시작 및 완료 경보를 사용 안함으로 설정하려면 `enable_start_finish_alerts` 등록 정보를 `false`로 설정합니다.

```
PUT /api/service/v2/services/replication
Host: zfs-storage.example.com:215
Content-Type: application/json

{ "enable_start_finish_alerts":false }
```

복제 대상

다음 표는 사용 가능한 복제 대상 명령을 보여 줍니다.

표 56 복제 대상 명령

요청	경로 <code>/api/storage/v{1 2}</code> 에 추가	설명
POST	<code>/replication/targets</code>	새 복제 대상을 만듭니다.
GET	<code>/replication/targets/target</code>	지정된 복제 대상 등록 정보를 가져옵니다.

요청	경로 <code>/api/storage/v{1 2}</code> 에 추가	설명
GET	<code>/replication/targets</code>	모든 복제 대상 객체를 나열합니다.
PUT	<code>/replication/targets/target</code>	지정된 복제 대상 객체를 수정합니다.
DELETE	<code>/replication/targets/target</code>	지정된 대상 객체를 삭제합니다.

다음 표에서는 복제 대상의 등록 정보를 보여줍니다.

등록 정보	설명
label	표시할 대상 이름입니다.
hostname	대상 어플라이언스의 전체 도메인 이름 또는 IPv4 주소입니다.
host_match	호스트 이름 확인을 수행하거나 무시합니다. "대상 인증서 확인" [191] 을 참조하십시오.
auto_accept_cert	대상의 인증서를 자동으로 수락합니다. "대상 인증서 확인" [191] 을 참조하십시오.

복제 대상 나열

시스템에서 사용할 수 있는 모든 복제 대상을 나열합니다.

요청 예:

```
GET /api/storage/v2/replication/targets HTTP/1.1
Host: zfs-storage.example.com:215
Authorization: Basic Tm8gcGVla2luZyE=
Accept: application/json
```

결과 예:

```
HTTP/1.1 200 OK
X-Zfssa-Replication-API: 1.0
Content-Type: application/json
Content-Length: 529

{
  "targets": [{
    "address": "ipaddr-1",
    "label": "zfs-storage-1",
    "hostname": "ipaddr-2",
    "asn": "9d7a7543-ca83-68f5-a8fc-f818f65e1cfc",
    "actions": 0,
    "target": "target-000",
    "href": "/api/storage/v2/replication/targets/zfs-storage-1"
  }, {
    "address": "ipaddr-3",
    "label": "zfs-storage-2",
    "hostname": "ipaddr-4",
    "asn": "16a4c82c-26c1-4a50-e317-ac53181f2e86",
    "actions": 0,
    "target": "target-001",
```

```

        "href": "/api/storage/v2/replication/targets/zfs-storage-2"
    }
}

```

지정된 복제 대상 나열

이 명령은 단일 복제 대상의 세부정보를 나열합니다. 대상은 해당 호스트 이름으로 액세스합니다.

요청 예:

```

GET /api/storage/v2/replication/targets/zfs-storage-1 HTTP/1.1
Authorization: Basic Tm8gcGVla2luZyE=
Host: zfs-storage.example.com:215
Accept: application/json

```

결과 예:

```

HTTP/1.1 200 OK
X-Zfssa-Replication-API: 1.0
Content-Type: application/json
Content-Length: 337

{
  "target": {
    "href": "/api/storage/v2/replication/targets/zfs-storage-1",
    "address": "ipaddr-1",
    "label": "zfs-storage-1",
    "hostname": "ipaddr-2",
    "asn": "9d7a7543-ca83-68f5-a8fc-f818f65e1cfc",
    "actions": 0
  }
}

```

복제 대상 만들기

`targets` 명령은 원격 복제에 대한 새 복제 대상을 만듭니다.

복제 트래픽이 특정 네트워크 인터페이스를 통과하도록 해야 하는 경우 [“경로 추가” \[103\]](#)에서와 같이 해당 인터페이스를 지정하는 대상에 대해 정적 경로를 설정합니다.

요청 예:

`hostname` 및 `auto_accept_cert` 등록 정보에 대한 자세한 내용은 [“대상 인증서 확인” \[191\]](#)을 참조하십시오.

```

POST /api/replication/v2/targets HTTP/1.1
Authorization: Basic Tm8gcGVla2luZyE=
Host: zfs-storage.example.com:215
Accept: application/json
Content-Type: application/json
Content-Length: 54

```

```
{
  "label": "zfs-storage-3",
  "hostname": "zfs-storage-3.example.com",
  "root_password": "root-password",
  "auto_accept_cert": true
}
```

결과 예:

```
HTTP/1.1 201 Created
Content-Length: 135
Content-Type: application/json
Location: /service/v2/services/replication/targets/target-000
X-Zfssa-Replication-API: 1.0
{
  "target": {
    "actions": 0,
    "address": "123.45.78.9:216",
    "asn": "fa5bf303-0dcb-e20d-ac92-cd129ccd2c81",
    "auto_accept_cert": true,
    "hostname": "zfs-storage-3.example.com",
    "href": "/service/v2/services/replication/targets/target-000",
    "label": "zfs-storage-3"
  }
}
```

대상 인증서 확인

복제 대상을 만들면 인증서 확인이 수행됩니다. 인증서 확인은 다음과 같은 단계로 구성됩니다.

1. 인증서 호스트 이름 검사
2. 인증서 신뢰 검사

호스트 이름 검사 또는 인증서 신뢰 검사가 실패하면 대상이 만들어지지 않습니다.

호스트 이름 검사

`hostname` 등록 정보의 값은 전체 도메인 이름 또는 IPv4 주소일 수 있습니다. 대상의 전체 도메인 이름을 값으로 사용하는 것이 좋습니다.

호스트 이름 검사에서는 대상에 대해 `hostname` 등록 정보에서 지정된 호스트 이름이 인증서에 지정된 호스트와 일치하는지 확인합니다. `hostname`에 대해 IP 주소 또는 비정규화된 도메인 이름을 지정하고 인증서에만 전체 도메인 이름이 있을 경우, 호스트 이름 검사가 실패하고 대상이 만들어지지 않습니다.

대상에서 ASN 기반 인증서가 사용되는 경우 `hostname` 등록 정보의 값에 대해 대상의 전체 도메인 이름을 지정합니다.

호스트 이름 검사는 기본적으로 수행됩니다. `host_match` 등록 정보를 `false`로 설정한 경우 호스트 이름 검사가 수행되지 않습니다.

보안 강화를 위해서는 `hostname` 등록 정보의 값을 대상의 전체 도메인 이름으로 설정하고 `host_match` 등록 정보가 `true`로 설정되도록 해야 합니다.

인증서 신뢰 검사

인증서 신뢰 검사는 다음 인증서 중 하나가 소스의 신뢰할 수 있는 인증서 목록에 추가되었으며 피어용으로 사용으로 설정되어 있는지 확인합니다.

- 대상 어플라이언스의 인증서
- 대상 어플라이언스의 인증서를 발급한 인증 기관의 인증서

인증서를 신뢰할 수 없을 경우 HTTP 상태 409(Conflict)가 반환되고 대상이 만들어지지 않습니다. 다음 예에서는 가독성을 위해 메시지 행이 구분되어 있습니다.

```
{
  "fault": {
    "code": 409,
    "name": "ERR_ILLEGAL_STATE",
    "message": "operation failed due to illegal state (Certificate is not trusted
               (encountered while attempting to run command \"commit\")"
  }
}
```

이 소스에 대해 해당 대상을 처음 만들 때 대상 호스트의 인증서는 신뢰할 수 있는 것으로 확인되지 않습니다. RESTful API는 인증서 확인 프롬프트를 표시할 수 없으므로 대상의 인증서가 자동으로 수락되도록 `auto_accept_cert` 등록 정보를 `true`로 설정합니다.

대상이 만들어지면 인증서가 신뢰할 수 없는 것으로 바뀔 수 있습니다. 예를 들어, 소스의 관리자가 신뢰할 수 있는 인증서 목록에서 인증서를 제거할 수도 있고, 대상의 관리자가 인증서를 바꿀 수도 있습니다. 대상 수정을 통해 `auto_accept_cert` 등록 정보를 `true`로 설정하여 이 인증서가 신뢰할 수 있는 것으로 수락되도록 소스에 알립니다.

인증서 신뢰 검사는 모든 피어 및 복제 연결에 대해 수행됩니다. 인증서를 신뢰할 수 없을 경우 소스가 연결을 거부합니다.

지문이나 발급자의 `commonName` 또는 `SubjectAltName`과 같은 인증서의 등록 정보를 확인하거나, 인증서를 업로드하거나, 인증서를 삭제하려면 [인증서 관리 \[33\]](#)를 참조하십시오.

복제 대상 수정

이 명령은 복제 대상의 등록 정보를 수정합니다.

요청 예:

```
PUT /api/replication/v2/targets/target-001 HTTP/1.1
Authorization: Basic Tm8gcGVla2luZyE=
Host: zfs-storage.example.com:215
Accept: application/json
```

```
Content-Type: application/json
Content-Length: 78

{ "hostname": "zfs-storage-3.example.com" }
```

복제 대상 삭제

이 명령은 기존 복제 대상을 삭제합니다.

요청 예:

```
DELETE /service/v2/services/replication/targets/target-000 HTTP/1.1
Host: zfs-storage.example.com:215
Authorization: Basic Tm8gcGV1a2luZyE=
```

성공한 삭제는 HTTP 상태 204(No Content)를 반환합니다.

복제 작업

복제 작업은 데이터를 복제 대상에 복제하는 규칙을 정의합니다. 다음 명령은 복제 작업을 관리합니다.

일반 작업 인터페이스 사용

복제 작업 관리 요청은 프로젝트 또는 공유를 지정하지 않고 어플라이언스에 대해 직접적으로 생성될 수 있습니다.

다음 표에서는 복제 작업을 관리하는 기본 명령을 보여줍니다.

표 57 기본 작업 인터페이스

요청	경로 <code>/api/storage/v{1 2}</code> 에 추가	설명
GET	<code>/replication/actions</code>	모든 복제 작업 객체를 나열합니다.
GET	<code>/replication/actions/ra_id</code>	지정된 복제 작업 등록 정보를 가져옵니다.
PUT	<code>/replication/actions/ra_id</code>	지정된 복제 작업 객체를 수정합니다.
DELETE	<code>/replication/actions/ra_id</code>	지정된 복제 작업 객체를 삭제합니다.
PUT	<code>/replication/actions/ra_id/sendupdate</code>	선택된 복제 작업을 시작합니다.
PUT	<code>/replication/actions/ra_id/cancelupdate</code>	선택된 복제 작업을 중지합니다.

다음 표에서는 복제 작업 일정을 관리하는 명령을 보여줍니다.

표 58 작업 일정 액세스

요청	경로 <code>/api/storage/v{1 2}</code> 에 추가	설명
GET	<code>/replication/actions/ra_id/schedules</code>	모든 복제 작업 일정 객체를 나열합니다.
GET	<code>/replication/actions/ra_id/schedules/ra_schedule</code>	지정된 복제 작업 일정 등록 정보를 가져옵니다.
POST	<code>/replication/actions/ra_id/schedules</code>	새 복제 작업 일정을 만듭니다.
PUT	<code>/replication/actions/ra_id/schedules/ra_schedule</code>	지정된 복제 작업 일정 객체를 수정합니다.
DELETE	<code>/replication/actions/ra_id/schedules/ra_schedule</code>	지정된 복제 작업 일정 객체를 삭제합니다.

다음 표에서는 복제 자동 스냅샷 관리 명령을 보여줍니다.

주 - 프로젝트 레벨 복제 작업 내에 구성된 공유 레벨 자동 스냅샷 일정은 다음 명령을 사용하여 액세스할 수 없습니다. 프로젝트 레벨 작업의 경우 여러 공유에 여러 개의 자동 스냅샷 일정이 있을 수 있으며, 이 인터페이스는 모든 조합을 식별하는 한정적인 방법을 제공하지 않습니다.

표 59 복제 자동 스냅샷 구성 액세스

요청	경로 <code>/api/storage/v{1 2}</code> 에 추가	설명
GET	<code>/replication/actions/ra_id/autosnaps</code>	선택한 복제 작업에 대한 자동 스냅샷 구성을 검색합니다.
GET	<code>/replication/actions/ra_id/autosnaps/autosnaps_id</code>	지정된 복제 작업 자동 스냅샷 객체를 가져옵니다.
PUT	<code>/replication/actions/ra_id/autosnaps</code>	지정된 복제 작업 자동 스냅샷 등록 정보를 수정합니다.
PUT	<code>/replication/actions/ra_id/autosnaps/autosnaps_id</code>	지정된 복제 작업 자동 스냅샷 객체를 수정합니다.
DELETE	<code>/replication/actions/ra_id/autosnaps/autosnaps_id</code>	지정된 복제 작업 자동 스냅샷 객체를 삭제합니다.

다음 표에는 복제 작업 통계 명령이 나와 있습니다.

표 60 복제 작업 통계 액세스

요청	경로 <code>/api/storage/v{1 2}</code> 에 추가	설명
GET	<code>/replication/actions/ra_id/stats</code>	선택한 복제 작업에 대한 읽기 전용 복제 통계를 검색합니다.

프로젝트, 파일 시스템 또는 LUN 컨텍스트에서 복제 작업

복제 작업 관리 요청은 특정 프로젝트, 파일 시스템 또는 LUN 컨텍스트에서 생성될 수도 있습니다.

다음 표에서는 복제 작업을 관리하는 기본 명령을 보여줍니다.

- 프로젝트 기반 작업 URI는 다음으로 시작합니다.

```
/api/storage/v{1|2}/pools/pool/projects/project
```

- 파일 시스템 기반 작업 URI는 다음으로 시작합니다.

```
/api/storage/v{1|2}/pools/pool/projects/project/filesystems/filesystem
```

- LUN 기반 작업 URI는 다음으로 시작합니다.

```
/api/storage/v{1|2}/pools/pool/projects/project/luns/lun
```

복제 작업을 관리하려면 위에 나열된 원하는 컨텍스트 URI에 다음 기본 명령을 추가하십시오.

표 61 프로젝트, 파일 시스템 또는 LUN 기본 복제 작업 인터페이스

요청	위에 나열된 프로젝트, 파일 시스템 또는 LUN URI에 추가	설명
GET	/replication/actions	모든 복제 작업 객체를 나열합니다.
GET	/replication/actions/ra_id	지정된 복제 작업 등록 정보를 가져옵니다.
POST	/replication/actions	새 복제 작업 만들기
PUT	/replication/actions/ra_id	지정된 복제 작업 객체를 수정합니다.
DELETE	/replication/actions/ra_id	지정된 복제 작업 객체를 삭제합니다.
PUT	/replication/actions/ra_id/sendupdate	선택된 복제 작업을 시작합니다.
PUT	/replication/actions/ra_id/cancelupdate	선택된 복제 작업을 중지합니다.

다음 표에는 복제 일정을 관리하는 기본 명령이 나와 있습니다.

- 프로젝트 기반 작업 URI는 다음으로 시작합니다.

```
/api/storage/v{1|2}/pools/pool/projects/project
```

- 파일 시스템 기반 작업 URI는 다음으로 시작합니다.

```
/api/storage/v{1|2}/pools/pool/projects/project/filesystems/filesystem
```

- LUN 기반 작업 URI는 다음으로 시작합니다.

```
/api/storage/v{1|2}/pools/pool/projects/project/luns/lun
```

복제 일정을 관리하려면 위에 나열된 원하는 컨텍스트 URI에 다음 기본 명령을 추가하십시오.

표 62 프로젝트, 파일 시스템 또는 LUN 복제 작업 일정

요청	위에 나열된 프로젝트, 파일 시스템 또는 LUN URI에 추가	설명
GET	/replication/actions/ra_id/schedules	모든 복제 작업 일정 객체를 나열합니다.
GET	/replication/actions/ra_id/schedules/ra_schedule	지정된 복제 작업 일정 등록 정보를 가져옵니다.
POST	/replication/actions/ra_id/schedules	새 복제 작업 일정을 만듭니다.

요청	위에 나열된 프로젝트, 파일 시스템 또는 LUN URI에 추가	설명
PUT	/replication/actions/ra_id/schedules/ra_schedule	지정된 복제 작업 일정 객체를 수정합니다.
DELETE	/replication/actions/ra_id/schedules/ra_schedule	지정된 복제 작업 일정 객체를 삭제합니다.

다음 표에는 복제 자동 스냅샷 구성을 관리하는 기본 명령이 나와 있습니다.

- 프로젝트 기반 작업 URI는 다음으로 시작합니다.

`/api/storage/v{1|2}/pools/pool/projects/project`

- 파일 시스템 기반 작업 URI는 다음으로 시작합니다.

`/api/storage/v{1|2}/pools/pool/projects/project/filesystems/filesystem`

- LUN 기반 작업 URI는 다음으로 시작합니다.

`/api/storage/v{1|2}/pools/pool/projects/project/luns/lun`

복제 자동 스냅샷 구성을 관리하려면 위에 나열된 원하는 컨텍스트 URI에 다음 기본 명령을 추가하십시오.

주 - 프로젝트 레벨 복제 작업 내에 구성된 공유 레벨 자동 스냅샷 일정은 다음 프로젝트 기반 작업을 사용하여 액세스할 수 없습니다. 프로젝트 레벨 작업의 경우 여러 공유에 여러 개의 자동 스냅샷 일정이 있을 수 있으며, 이 인터페이스는 모든 조합을 식별하는 한정적인 방법을 제공하지 않습니다.

표 63 프로젝트, 파일 시스템, LUN 복제 자동 스냅샷 구성

요청	위에 나열된 프로젝트, 파일 시스템 또는 LUN URI에 추가	설명
GET	/replication/actions/ra_id/autosnaps	프로젝트/공유의 선택한 복제 작업에 대한 자동 스냅샷 구성을 검색합니다.
GET	/replication/actions/ra_id/autosnaps/autosnaps_id	프로젝트/공유의 지정된 복제 작업 자동 스냅샷 구성을 가져옵니다.
POST	/replication/actions/ra_id/autosnaps	프로젝트/공유의 레벨 복제 작업 자동 스냅샷 객체를 새로 만듭니다.
PUT	/replication/actions/ra_id/autosnaps	프로젝트/공유의 지정된 복제 작업 대상 자동 스냅샷 보존 정책을 수정합니다.
PUT	/replication/actions/ra_id/autosnaps/autosnaps_id	지정된 복제 작업 자동 스냅샷 객체를 수정합니다.
DELETE	/replication/actions/ra_id/autosnaps/autosnaps_id	지정된 복제 작업 자동 스냅샷 객체를 삭제합니다.

다음 표에는 복제 작업 통계에 액세스하는 기본 명령이 나와 있습니다.

- 프로젝트 기반 작업 URI는 다음으로 시작합니다.

`/api/storage/v{1|2}/pools/pool/projects/project`

- 파일 시스템 기반 작업 URI는 다음으로 시작합니다.

```
/api/storage/v{1|2}/pools/pool/projects/project/filesystems/filesystem
```

- LUN 기반 작업 URI는 다음으로 시작합니다.

```
/api/storage/v{1|2}/pools/pool/projects/project/luns/lun
```

복제 작업 통계에 액세스하려면 위에 나열된 원하는 컨텍스트 URI에 다음 기본 명령을 추가하십시오.

표 64 복제 작업 통계 액세스

요청	위에 나열된 프로젝트, 파일 시스템 또는 LUN URI에 추가	설명
GET	/replication/actions/ <i>ra_id</i> /stats	선택한 복제 작업에 대한 읽기 전용 복제 통계를 검색합니다.

복제 작업 나열

사용 가능한 복제 작업 목록을 가져옵니다.

요청 예:

```
GET /api/storage/v2/replication/actions HTTP/1.1
Authorization: Basic Tm8gcGV1a2luZyE=
Accept: application/json
```

결과 예:

```
HTTP/1.1 200 OK
X-Zfssa-Replication-API: 1.0
Content-Type: application/json
Content-Length: 529
```

```
{
  "actions": [{
    "href": ""
    ...
  }, {
    "href": "",
    ...
  }]
}
```

복제 작업 가져오기

복제 작업 상태 가져오기 명령은 해당 ID에 의해 지정된 단일 복제 작업의 상태를 반환합니다.

요청 예:

```
GET /api/storage/v2/replication/actions/1438ed7f-aad3-c631-d869-9e85cd7f15b4 HTTP/1.1
Authorization: Basic Tm8gcGV1a2luZyE=
```

Accept: application/json

결과 예:

HTTP/1.1 200 OK
X-Zfssa-Replication-API: 1.0
Content-Type: application/json
Content-Length: 529

```
{
  "action": {
    "average_throughput": 0.0,
    "bytes_sent": 0.0,
    "collection": "local",
    "compression": true,
    "continuous": false,
    "enabled": true,
    "estimated_size": 0.0,
    "estimated_time_left": 0.0,
    "href": "/api/storage/v2/replication/actions",
    "id": "8373d331-de60-e590-90e8-9ad69fcb4aec",
    "include_clone_origin_as_data": false,
    "include_snaps": true,
    "last_sync": "20130916T21:36:50",
    "last_try": "20130916T21:36:50",
    "max_bandwidth": 0,
    "pool": "p1",
    "project": "proj-01",
    "retain_user_snaps_on_target": false,
    "share": "fs1",
    "state": "sending",
    "target": "38094753-6c90-49ed-aa92-995a296d432a",
    "use_ssl": true
  }
}
```

요청 예:

다음 복제 작업 응답은 RPO(복구 지점 목표) 예와 관련 복제본 지연 경고 및 경보를 보여줍니다.

GET /api/storage/v2/replication/actions HTTP/1.1
Authorization: Basic Tm8gcGVla2luZyE=
Content-Type: application/json

결과 예:

HTTP/1.1 200 OK
X-Zfssa-Replication-API: 1.0
Content-Type: application/json
Content-Length: 529

```
{
  "action": {"id": "12d981c3-b098-c65a-e1e9-a6b8263a0f6a",
    "target_id": "4fd305ac-4af5-c34a-87c3-88203207305b",
    ...
    "replica_lag": "42:25:31",
    "recovery_point_objective": 0,
    "replica_lag_warning_alert": 0,
    "replica_lag_error_alert": 0,
    "replica_lag_over_warning_limit": false,
    "replica_lag_over_error_limit": false,
  }
```

```

    "project": "default"
  }
}

```

복제 작업 만들기

새 복제 작업을 만듭니다.

등록 정보를 만듭니다.

```

Initial values:
                target = cleo
                enabled = true
                continuous = false
                include_snaps = true
retain_user_snaps_on_target = false
                dedup = true
include_clone_origin_as_data = false
                max_bandwidth = unlimited
                bytes_sent = 0
                estimated_size = 0
                estimated_time_left = 0
                average_throughput = 0
                use_ssl = true
                compression = on

```

요청 예:

```

POST /api/storage/v2/replication/actions HTTP/1.1
Host: zfs-storage.example.com:215
Authorization: Basic Tm8gcGV1a2luZyE=
Content-Type: application/json
Content-Length: 121
Accept: application/json

```

```

{
  "pool": "p1",
  "project": "proj-01",
  "share": "fs1",
  "target_pool": "pool1",
  "target": "38094753-6c90-49ed-aa92-995a296d432a"
}

```

결과 예:

```

HTTP/1.1 201 Created
Content-Length: 506
Content-Type: application/json
Location: /api/storage/v2/replication/action/8373d331-de60-e590-90e8-9ad69fcb4aec
X-Zfssa-Replication-API: 1.0

```

```

{
  "action": {
    "project": "blue1",
    "target": "38094753-6c90-49ed-aa92-995a296d432a",
    "bytes_sent": 0.0,
    "compression": true,
    "continuous": false,
    "enabled": true,

```

```
    "dedup": false,
    "max_bandwidth": 0,
    "collection": "local",
    "estimated_size": 0.0,
    "state": "idle",
    "href": "/api/storage/v2/replication/pools/p1/projects/blah1/shares/fs1/
            actions/8373d331-de60-e590-90e8-9ad69fcb4aec",
    "average_throughput": 0.0,
    "use_ssl": true,
    "estimated_time_left": 0.0,
    "retain_user_snaps_on_target": false,
    "share": "fs1",
    "id": "8373d331-de60-e590-90e8-9ad69fcb4aec",
    "pool": "p1",
    "include_clone_origin_as_data": false,
    "include_snaps": true
  }
}
```

복제 작업에 대한 일정을 만듭니다.

요청 예:

```
POST /api/storage/v2/replication/actions/b77bd8cd-17ed-69da-9e4b-aebe3cc63755/schedules
HTTP/1.1
Host: zfs-storage.example.com:215
Authorization: Basic Tm8gcGVla2luZyE=
Accept: */*
Content-Type: application/json
Content-Length: 65

{"frequency": "month", "day": "5th", "hour": "auto", "minute": "auto"}
```

결과 예:

```
HTTP/1.1 201 Created
Date: Thu, 12 Jan 2017 17:35:48 GMT
Server: TwistedWeb/192.0.2
Content-Length: 0
X-Zfssa-Storage-API: 1.1
Content-Type: application/json; charset=utf-8
X-Zfssa-API-Version: 1.0
X-Zfssa-Version: user/generic@2016.12.08,1-0
```

복제 작업 수정

기존 복제 작업을 수정합니다.

요청 예:

```
PUT /api/storage/v2/replication/actions/c141d88d-ffd2-6730-d489-b71905f340cc HTTP/1.1
Host: zfs-storage.example.com:215
Authorization: Basic Tm8gcGVla2luZyE=
Content-Type: application/json

{"use_ssl": false}
```

결과 예:

```

HTTP/1.1 202 Accepted
X-Zfssa-Replication-API: 1.0
Content-Type: application/json
Content-Length: 620

```

```

{
  "action": {
    "target_id": "407642ae-91b5-681c-de5e-afcd5cbf2974",
    "compression": true,
    "continuous": false,
    "enabled": true,
    "max_bandwidth": 0,
    "dedup": false,
    "retain_user_snaps_on_target": false,
    "use_ssl": false,
    "id": "c141d88d-ffd2-6730-d489-b71905f340cc",
    "include_clone_origin_as_data": false,
    "include_snaps": true
  }
}

```

요청 예:

```

PUT /api/storage/v2/replication/actions/action_id -d '{"recovery_point_objective":
60}'HTTP/1.1
Host: zfs-storage.example.com:215
Authorization: Basic Tm8gcGVla2luZyE=
Content-Type: application/json

```

결과 예:

```

X-Zfssa-Replication-API: 1.0
Content-Type: application/json
Content-Length: 620

```

```

{
  "action": {
    "state_description": "Idle (no update in progress)",
    "recovery_point_objective": 60,
    "replica_lag_over_warning_limit": false,
    "bytes_sent": "0",
    "last_try": "Mon Nov 21 2016 23:25:59 GMT+0000 (UTC)",
    "max_bandwidth": 0,
    "estimated_size": "0",
    "href": "/api/storage/v2/replication/actions/12d981c3-b098-c65a-e1e9-a6b8263a0f6a",
    "estimated_time_left": 0,
    "use_ssl": true,
    "id": "12d981c3-b098-c65a-e1e9-a6b8263a0f6a",
    "stats": {"total_logical_bytes": 40656,
    "last_dd_table_build": 9169029,
    "total_after_dedup": 18476,
    "last_try": "Mon Nov 21 2016 23:25:59 GMT+0000 (UTC)",
    "dd_total_updates": 1,
    "href": "/api/storage/v2/replication/actions/12d981c3-b098-c65a-e1e9-a6b8263a0f6a/
stats",
    "dd_total_duration": 47149245470,
    "last_logical_bytes": 40656,
    "dd_total_table_mem": 2097152,
    "last_result": "success",
    "last_after_dedup": 18476,
    "last_duration": 47149245470,
    {"dd_total_logical_bytes": 40656,
    "total_updates": 1,
    "total_duration": 47149245470,

```

```

    "replica_data_timestamp": "Mon Nov 21 2016 23:25:12 GMT+0000 (UTC)",
    "total_to_network": 9623,
    "dd_total_table_build": 9169029,
    "dd_total_phys_bytes": 16800,
    "last_to_network": 9623,
    "total_phys_bytes": 16800,
    "last_phys_bytes": 16800,
    "last_sync": "Mon Nov 21 2016 23:25:59 GMT+0000 (UTC)",
    "last_dd_table_mem": 2097152,
    "dd_total_after_dedup": 18476,
    "dd_total_to_network": 9623},
    "compression": "on",
    "dedup": true,
    "replica_lag_warning_alert": 0,
    "last_result": "success",
    "include_clone_origin_as_data": false,
    "state": "idle",
    "offline": false,
    "export_path": "",
    "export_pending": false,
    "autosnaps": {"autosnaps_retention_policies":
    "synchronized",
    "href": "/api/storage/v2/replication/actions/12d981c3-b098-c65a-e1e9-a6b8263a0f6a/
autosnaps"},
    "replica_data_timestamp": "Mon Nov 21 2016 23:25:12 GMT+0000 (UTC)",
    "continuous": false,
    "target_id": "4fd305ac-4af5-c34a-87c3-88203207305b",
    {"average_throughput": "0B/s",
    "next_update": "Sync now",
    "pool": "p1",
    "replica_lag_over_error_limit": false,
    "target": "pool1",
    "replica_lag": "42:28:24",
    "retain_user_snaps_on_target": false,
    ...
  }
}

```

복제 작업 진행률 모니터

복제 작업 상태 가져오기 명령은 해당 ID에 의해 지정된 단일 복제 작업의 상태를 반환합니다. 복제 진행률을 확인하려면 `state` 및 `state_description`을 검토하십시오.

`state` 등록 정보 값:

- sending
- idle

`state_description` 등록 정보 값:

- Connecting to replication target
- Receiving checkpoint from target
- Estimating size of update
- Building deduplication tables

이 등록 정보 값은 중복 제거된 복제 스트림에만 사용됩니다.

요청 예:

```
GET /api/storage/v2/replication/actions/1438ed7f-aad3-c631-d869-9e85cd7f15b4 HTTP/1.1
Authorization: Basic Tm8gcGVla2luZyE=
Accept: application/json
```

결과 예:

```
HTTP/1.1 200 OK
X-Zfssa-Replication-API: 1.0
Content-Type: application/json
Content-Length: 529
```

```
{
  "action": {
    "id": "1438ed7f-aad3-c631-d869-9e85cd7f15b4",
    "target_id": "4fd3483e-b1f5-4bdc-9be3-b3a4becd0c42",
    "target": "cleo",
    "pool": "p0",
    "replication_of": "testproj",
    "enabled": true,
    "continuous": false,
    "include_snaps": true,
    "retain_user_snaps_on_target": false,
    "dedup": true,
    "include_clone_origin_as_data": false,
    "max_bandwidth": -1,
    "bytes_sent": 0,
    "estimated_size": 0,
    "estimated_time_left": 0,
    "average_throughput": 0,
    "use_ssl": true,
    "compression": "on",
    "export_path": "",
    "state": "sending",
    "state_description": "Receiving checkpoint from target",
    "export_pending": false,
    "offline": false,
    "next_update": "Sync now",
    "replica_data_timestamp": "Thu Apr 28 2016 22:38:03 GMT+0000 (UTC)",
    "last_sync": "<unknown>",
    "last_try": "<unknown>",
    "last_result": "<unknown>",
    "replica_lag": "00:00:18",
    "recovery_point_objective": 0,
    "replica_lag_warning_alert": 0,
    "replica_lag_error_alert": 0,
    "replica_lag_over_warning_limit": false,
    "replica_lag_over_error_limit": false,
    "project": "testproj"
  }
}
```

업데이트 취소

진행 중인 복제 업데이트를 취소합니다.

요청 예:

```
PUT /api/storage/v2/replication/actions/c141d88d-ffd2-6730-d489-b71905f340cc/cancelupdate
HTTP/1.1
Host: zfs-storage.example.com:215
Authorization: Basic Tm8gcGVla2luZyE=
```

결과 예:

```
HTTP/1.1 202 Accepted
X-Zfssa-Replication-API: 1.0
```

업데이트 전송

복제 업데이트를 가능한 한 빨리 시작하도록 예약합니다.

요청 예:

```
PUT /api/storage/v2/replication/actions/c141d88d-ffd2-6730-d489-b71905f340cc/sendupdate
HTTP/1.1
Authorization: Basic Tm8gcGVla2luZyE=
```

결과 예:

```
HTTP/1.1 202 Accepted
X-Zfssa-Replication-API: 1.0
```

복제 작업 삭제

기존 복제 작업을 삭제합니다.

요청 예:

```
DELETE /api/storage/v2/replication/actions/e7e688b1-ff07-474f-d5cd-cac08293506e HTTP/1.1
Host: zfs-storage.example.com:215
Authorization: Basic Tm8gcGVla2luZyE=
```

성공한 삭제는 HTTP 상태 204(No Content)를 반환합니다.

결과 예:

```
HTTP/1.1 204 No-Content
X-Zfssa-Replication-API: 1.0
```

복제 패키지

이 절에서는 복제 패키지 및 소스 명령에 대해 자세히 설명합니다.

표 65 복제 패키지 명령

요청	경로 <code>/api/storage/v{1 2}/replication</code> 에 추가	설명
GET	<code>/packages</code>	모든 복제 패키지를 나열합니다.
GET	<code>/packages/package</code>	지정된 복제 패키지를 가져옵니다.
PUT	<code>/packages/package</code>	지정된 복제 패키지를 수정합니다.
DELETE	<code>/packages/package</code>	지정된 복제 패키지를 삭제합니다.
PUT	<code>/packages/package/cancelupdate</code>	지정된 패키지에 대한 <code>cancelupdate</code> 를 실행합니다.
PUT	<code>/packages/package/sever</code>	지정된 패키지에 대한 <code>sever</code> 를 실행합니다.
PUT	<code>/packages/package/pkgreverse</code>	지정된 패키지에 대한 <code>reverse</code> 를 실행합니다.
PUT	<code>/packages/package/clone</code>	지정된 패키지를 복제합니다.
GET	<code>/packages/package/clone/conflicts</code>	공유 등록 정보 충돌을 나열합니다.
GET	<code>/packages/package/projects</code>	패키지 프로젝트를 나열합니다.
GET	<code>/packages/package/projects/project</code>	패키지 프로젝트를 가져옵니다.
PUT	<code>/packages/package/projects/project</code>	패키지 프로젝트를 수정합니다.
GET	<code>/packages/package/projects/project/usage/groups</code>	패키지 프로젝트 그룹 사용량을 가져옵니다.
GET	<code>/packages/package/projects/project/usage/users</code>	패키지 프로젝트 사용자 사용량을 가져옵니다.
GET	<code>/packages/package/projects/project/snapshots</code>	모든 스냅샷 객체를 나열합니다.
GET	<code>/packages/package/projects/project/snapshots/snapshot</code>	지정된 스냅샷 등록 정보를 가져옵니다.
DELETE	<code>/packages/package/projects/project/snapshots/snapshot</code>	지정된 스냅샷 객체를 삭제합니다.
PUT	<code>/packages/package/projects/project/snapshots/snapshot</code>	패키지 프로젝트 스냅샷의 이름을 바꿉니다.
GET	<code>/packages/package/projects/project/automatic</code>	모든 패키지 프로젝트 자동 스냅샷 객체를 나열합니다.
GET	<code>/packages/package/projects/project/automatic/automatic</code>	지정된 패키지 프로젝트 자동 스냅샷 등록 정보를 가져옵니다.
GET	<code>/packages/package/projects/project/filesystems</code>	패키지 파일 시스템을 나열합니다.
GET	<code>/packages/package/projects/project/filesystems/filesystem</code>	패키지 파일 시스템을 가져옵니다.
PUT	<code>/packages/package/projects/project/filesystems/filesystem</code>	패키지 파일 시스템을 수정합니다.
GET	<code>/packages/package/projects/project/filesystems/filesystem/usage/groups</code>	패키지 파일 시스템 그룹 사용량을 가져옵니다.
GET	<code>/packages/package/projects/project/filesystems/filesystem/usage/users</code>	패키지 파일 시스템 사용자 사용량을 가져옵니다.
GET	<code>/packages/package/projects/project/filesystems/filesystem/snapshots/snapshot</code>	지정된 스냅샷 등록 정보를 가져옵니다.
GET	<code>/packages/package/projects/project/filesystems/filesystem/snapshots</code>	모든 스냅샷 객체를 나열합니다.
DELETE	<code>/packages/package/projects/project/filesystems/filesystem/snapshots/snapshot</code>	지정된 스냅샷 객체를 삭제합니다.
PUT	<code>/packages/package/projects/project/filesystems/filesystem/snapshots/snapshot</code>	패키지 파일 시스템 스냅샷의 이름을 바꿉니다.

복제 패키지

요청	경로 <i>/api/storage/v{1 2}/replication</i> 에 추가	설명
GET	<i>/packages/package/projects/project/filesystems/filesystem/automatic</i>	모든 패키지 파일 시스템 자동 스냅샷 객체를 나열합니다.
GET	<i>/packages/package/projects/project/filesystems/filesystem/automatic/automatic</i>	지정된 패키지 파일 시스템 자동 스냅샷 등록 정보를 가져옵니다.
GET	<i>/packages/package/projects/project/luns</i>	패키지 LUN을 나열합니다.
GET	<i>/packages/package/projects/project/luns/lun</i>	패키지 LUN을 가져옵니다.
PUT	<i>/packages/package/projects/project/luns/lun</i>	패키지 LUN을 수정합니다.
GET	<i>/packages/package/projects/project/luns/lun/usage/groups</i>	패키지 LUN 그룹 사용량을 가져옵니다.
GET	<i>/packages/package/projects/project/luns/lun/usage/users</i>	패키지 LUN 사용자 사용량을 가져옵니다.
GET	<i>/packages/package/projects/project/luns/lun/snapshots/snapshot</i>	지정된 스냅샷 등록 정보를 가져옵니다.
GET	<i>/packages/package/projects/project/luns/lun/snapshots</i>	모든 스냅샷 객체를 나열합니다.
DELETE	<i>/packages/package/projects/project/luns/lun/snapshots/snapshot</i>	지정된 스냅샷 객체를 삭제합니다.
PUT	<i>/packages/package/projects/project/luns/lun/snapshots/snapshot</i>	패키지 LUN 스냅샷의 이름을 바꿉니다.
GET	<i>/packages/package/projects/project/luns/lun/automatic</i>	모든 패키지 LUN 자동 스냅샷 객체를 나열합니다.
GET	<i>/packages/package/projects/project/luns/lun/automatic/automatic</i>	지정된 패키지 LUN 자동 스냅샷 등록 정보를 가져옵니다.

다음 명령을 사용하여 복제 소스 및 해당하는 패키지에도 액세스할 수 있습니다.

표 66 복제 소스 명령

요청	경로 <i>/api/storage/v{1 2}/replication/sources</i> 에 추가	설명
GET	<i>/api/storage/v{1 2}/replication/sources</i> 만 사용	복제 소스를 나열합니다.
GET	<i>/source</i>	복제 소스 세부정보를 나열합니다.
GET	<i>/source/packages/package</i>	지정된 복제 패키지를 가져옵니다.
PUT	<i>/source/packages/package</i>	지정된 복제 패키지를 수정합니다.
DELETE	<i>/source/packages/package</i>	지정된 복제 패키지를 삭제합니다.
PUT	<i>/source/packages/package/cancelupdate</i>	지정된 패키지에 대한 <i>cancelupdate</i> 를 실행합니다.
PUT	<i>/source/packages/package/sever</i>	지정된 패키지에 대한 <i>sever</i> 를 실행합니다.
PUT	<i>/source/packages/package/pkgreverse</i>	지정된 패키지에 대한 <i>reverse</i> 를 실행합니다.
PUT	<i>/source/packages/package/clone</i>	지정된 패키지를 복제합니다.

요청	경로 <i>/api/storage/v{1 2}/replication/sources</i> 에 추가	설명
GET	<i>/source/packages/package/clone/conflicts</i>	공유 등록 정보 충돌을 나열합니다.
GET	<i>/source/packages/package/projects</i>	패키지 프로젝트를 나열합니다.
GET	<i>/source/packages/package/projects/project</i>	패키지 프로젝트를 가져옵니다.
PUT	<i>/source/packages/package/projects/project</i>	패키지 프로젝트를 수정합니다.
GET	<i>/source/packages/package/projects/project/usage/groups</i>	패키지 프로젝트 그룹 사용량을 가져옵니다.
GET	<i>/source/packages/package/projects/project/usage/users</i>	패키지 프로젝트 사용자 사용량을 가져옵니다.
GET	<i>/source/packages/package/projects/project/snapshots/snapshot</i>	지정된 스냅샷 등록 정보를 가져옵니다.
GET	<i>/source/packages/package/projects/project/snapshots</i>	모든 스냅샷 객체를 나열합니다.
DELETE	<i>/source/packages/package/projects/project/snapshots/snapshot</i>	지정된 스냅샷 객체를 삭제합니다.
PUT	<i>/source/packages/package/projects/project/snapshots/snapshot</i>	패키지 프로젝트 스냅샷의 이름을 바꿉니다.
GET	<i>/source/packages/package/projects/project/automatic</i>	모든 패키지 프로젝트 자동 스냅샷 객체를 나열합니다.
GET	<i>/source/packages/package/projects/project/automatic/automatic</i>	지정된 패키지 프로젝트 자동 스냅샷 등록 정보를 가져옵니다.
GET	<i>/source/packages/package/projects/project/filesystems</i>	패키지 파일 시스템을 나열합니다.
GET	<i>/source/packages/package/projects/project/filesystems/filesystem</i>	패키지 파일 시스템을 가져옵니다.
PUT	<i>/source/packages/package/projects/project/filesystems/filesystem</i>	패키지 파일 시스템을 수정합니다.
GET	<i>/source/packages/package/projects/project/filesystems/filesystem/usage/groups</i>	패키지 파일 시스템 그룹 사용량을 가져옵니다.
GET	<i>/source/packages/package/projects/project/filesystems/filesystem/usage/users</i>	패키지 파일 시스템 사용자 사용량을 가져옵니다.
GET	<i>/source/packages/package/projects/project/filesystems/filesystem/snapshots/snapshot</i>	지정된 스냅샷 등록 정보를 가져옵니다.
GET	<i>/source/packages/package/projects/project/filesystems/filesystem/snapshots</i>	모든 스냅샷 객체를 나열합니다.
DELETE	<i>/source/packages/package/projects/project/filesystems/filesystem/snapshots/snapshot</i>	지정된 스냅샷 객체를 삭제합니다.
PUT	<i>/source/packages/package/projects/project/filesystems/filesystem/snapshots/snapshot</i>	패키지 파일 시스템 스냅샷의 이름을 바꿉니다.
GET	<i>/source/packages/package/projects/project/filesystems/filesystem/automatic</i>	모든 패키지 파일 시스템 자동 스냅샷 객체를 나열합니다.
GET	<i>/source/packages/package/projects/project/filesystems/filesystem/automatic/automatic</i>	지정된 패키지 파일 시스템 자동 스냅샷 등록 정보를 가져옵니다.
GET	<i>/source/packages/package/projects/project/luns</i>	패키지 LUN을 나열합니다.
GET	<i>/source/packages/package/projects/project/luns/lun</i>	패키지 LUN을 가져옵니다.

요청	경로 <code>/api/storage/v{1 2}/replication/sources</code> 에 추가	설명
PUT	<code>/source/packages/package/projects/project/luns/lun</code>	패키지 LUN을 수정합니다.
GET	<code>/source/packages/package/projects/project/luns/lun/usage/groups</code>	패키지 LUN 그룹 사용량을 가져옵니다.
GET	<code>/source/packages/package/projects/project/luns/lun/usage/users</code>	패키지 LUN 사용자 사용량을 가져옵니다.
GET	<code>/source/packages/package/projects/project/luns/lun/snapshots/snapshot</code>	지정된 스냅샷 등록 정보를 가져옵니다.
GET	<code>/source/packages/package/projects/project/luns/lun/snapshots</code>	모든 스냅샷 객체를 나열합니다.
DELETE	<code>/source/packages/package/projects/project/luns/lun/snapshots/snapshot</code>	지정된 스냅샷 객체를 삭제합니다.
PUT	<code>/source/packages/package/projects/project/luns/lun/snapshots/snapshot</code>	패키지 LUN 스냅샷의 이름을 바꿉니다.
GET	<code>/source/packages/package/projects/project/luns/lun/automatic</code>	모든 패키지 LUN 자동 스냅샷 객체를 나열합니다.
GET	<code>/source/packages/package/projects/project/luns/lun/automatic/automatic</code>	지정된 패키지 LUN 자동 스냅샷 등록 정보를 가져옵니다.

복제 소스 나열

사용 가능한 모든 복제 소스를 나열합니다.

요청 예:

```
GET /api/storage/v2/replication/sources HTTP/1.1
Host: zfs-storage.example.com:215
Accept: application/json
```

출력 예:

```
HTTP/1.1 200 OK
X-Zfssa-Replication-API: 1.0
Content-Type: application/json
Content-Length: 529

{
  "sources": [{
    "asn": "314d252e-c42b-e844-dab1-a3bca680b563",
    "href": "/api/storage/v2/replication/sources/zfs-repl-host",
    "ip_address": "ipaddr-1",
    "name": "zfs-repl-host",
    "source": "source-000"
  }]
}
```

복제 패키지 나열

모든 복제 패키지를 나열합니다.

요청 예:

```
GET /api/storage/v2/replication/packages HTTP/1.1
Host: zfs-storage.example.com:215
Accept: application/json
```

결과 예:

```
HTTP/1.1 200 OK
X-Zfssa-Replication-API: 1.0
Content-Type: application/json
Content-Length: 529

{
  "packages": [
    {
      "href": "/api/storage/v2/replication/packages/0efaab49-7b22-4d4a-def8-813c27780894",
      "id": "0efaab49-7b22-4d4a-def8-813c27780894",
      "source_name": "sourceA",
      "source_asn": "8a22f6e0-4ee4-4b85-f141-e152f5fac961",
      "source_ip": "jpadr-1",
      "source_pool": "poolA",
      "target_pool": "poolA",
      "replica_of": "projTest",
      "enabled": true,
      "state": "idle",
      "state_description": "Idle (no update in progress)",
      "offline": false,
      "import_path": "",
      "data_timestamp": "2017-03-09T22:36:12Z",
      "last_sync": "2017-03-09T22:36:22Z",
      "last_try": "2017-03-09T22:36:22Z",
      "last_result": "success"
    }
  ]
}
```

패키지 수정

패키지 등록 정보를 수정합니다.

등록 정보	유형	설명
enabled	boolean	복제 업데이트의 현재 상태

요청 예:

```
PUT /api/storage/v2/replication/packages/8373d331-de60-e590-90e8-9ad69fcb4aec HTTP/1.1
Host: zfs-storage.example.com:215
Authorization: Basic Tm8gcGVla2luZyE=
Content-Type: application/json
```

```
{"enabled": false}
```

결과 예:

```
HTTP/1.1 202 Accepted
X-Zfssa-Replication-API: 1.0
```

요청 예:

```
PUT /api/storage/v2/replication/packages/8373d331-de60-e590-90e8-9ad69fcb4aec/pkgreverse
HTTP/1.1
Host: zfs-storage.example.com:215
Authorization: Basic Tm8gcGVla2luZyE=
Content-Type: application/json

{"new_project_name":"restrev", "enable_action_upon_reversal":"true"}
```

결과 예:

```
HTTP/1.1 202 Accepted
X-Zfssa-Replication-API: 1.0
```

패키지 삭제

복제 패키지를 삭제합니다.

요청 예:

```
DELETE /api/storage/v2/replication/packages/8373d331-de60-e590-90e8-9ad69fcb4aec HTTP/1.1
Host: zfs-storage.example.com:215
Authorization: Basic Tm8gcGVla2luZyE=
```

성공한 삭제는 HTTP 상태 204(No Content)를 반환합니다.

결과 예:

```
HTTP/1.1 204 No-Content
X-Zfssa-Replication-API: 1.0
```

업데이트 취소

이 패키지에 대한 진행 중인 업데이트를 취소합니다.

요청 예:

```
PUT /api/storage/v2/replication/packages/8373d331-de60-e590-90e8-9ad69fcb4aec/cancelupdate
HTTP/1.1
Host: zfs-storage.example.com:215
Authorization: Basic Tm8gcGVla2luZyE=
```

진행 중인 업데이트가 없는 경우 HTTP 상태 409(Conflict)가 반환됩니다.

결과 예:

```
HTTP/1.1 409 Conflict
X-Zfssa-Replication-API: 1.0
```

```
Content-Type: application/json
Content-Length: 137

{
  "cancelupdate": {
    "AKSH_ERROR": "EAK_NAS_REPL_BADSTATE",
    "message": "operation illegal for state"
  }
}
```

결과 예:

```
HTTP/1.1 202 Accepted
X-Zfssa-Replication-API: 1.0
```

패키지 복제

패키지 프로젝트를 복제합니다.

요청 예:

```
PUT /api/v2/storage/replication/packages/8373d331-de60-e590-90e8-9ad69fcb4aec/clone
HTTP/1.1
Host: zfs-storage.example.com:215
Authorization: Basic Tm8gcGVla2luZyE=
```

결과 예:

```
HTTP/1.1 202 Accepted
X-Zfssa-Replication-API: 1.0
```

성공한 복제는 HTTP 상태 202(Accepted)를 반환합니다. 도우미 명령은 복제 작업과의 충돌이 있는지 확인하는 데 사용될 수 있습니다.

복제 충돌 요청 예:

```
GET /api/storage/v2/replication/packages/8373d331-de60-e590-90e8-9ad69fcb4aec/clone/
conflicts HTTP/1.1
Host: zfs-storage.example.com:215
Authorization: Basic Tm8gcGVla2luZyE=
```

복제/충돌에서 충돌 반환:

```
HTTP/1.1 200 OK
X-Zfssa-Replication-API: 1.0
Content-Type: application/json
Content-Length: 58

{
  "conflicts": "There are no conflicts."
}
```

등록 정보:

```
Default settings:
  target_project = (unset)
  original_mountpoint = /export
```

```
override_mountpoint = false
mountpoint =
```

패키지 단절

복제 연결을 단절하고 패키지 내용을 새 프로젝트로 이동합니다. 이 작업은 이 패키지 및 복제된 해당 공유를 소스 시스템과 영구적으로 단절하여 이러한 항목을 이 시스템에서 로컬 프로젝트로 만듭니다. 어느 한 방향의 후속 복제 업데이트 시 새 작업을 정의하고 전체 업데이트를 전송해야 합니다.

요청 예:

```
PUT /api/storage/v2/replication/packages/8373d331-de60-e590-90e8-9ad69fcb4aec/sever
  HTTP/1.1
Host: zfs-storage.example.com:215
Authorization: Basic Tm8gcGVla2luZyE=
```

```
{"projname":"restsev"}
```

성공 응답:

```
HTTP/1.1 202 Accepted
X-Zfssa-Replication-API: 1.0
```

패키지 방향 바꾸기

복제의 방향을 바꿉니다. 이 작업은 이 패키지에 대한 복제를 사용 안함으로 설정하고 이 패키지의 내용을 소스로 다시 복제하도록 구성된 새 로컬 프로젝트로 이동합니다. 새 프로젝트가 처음으로 소스로 다시 복제될 때 마지막 성공한 업데이트 이후 소스에서 수행된 모든 메타데이터 또는 데이터 변경사항이 손실됩니다.

요청 예:

```
PUT /api/storage/v2/replication/packages/8373d331-de60-e590-90e8-9ad69fcb4aec/reverse
  HTTP/1.1
Host: zfs-storage.example.com:215
Authorization: Basic Tm8gcGVla2luZyE=
```

```
{"projname":"restrev"}
```

성공 응답:

```
HTTP/1.1 202 Accepted
X-Zfssa-Replication-API: 1.0
```

스토리지 암호화

폴, 프로젝트 및 공유를 암호화할 수 있습니다. 폴이 암호화되면 각 자식 프로젝트 및 공유가 암호화되고 폴에서 암호화 등록 정보 값을 상속합니다. 암호화 등록 정보 값이 프로젝트에 대해 지정되면 상속된 값이 사용되지 않고 자식 공유가 프로젝트에 대해 지정된 로컬 값을 상속합니다. 암호화된 프로젝트는 암호화되지 않은 폴에 만들 수 있습니다. 암호화되지 않은 프로젝트는 암호화된 폴에 만들 수 없습니다.

프로젝트가 암호화되면 각 자식 공유가 암호화되고 프로젝트에서 암호화 등록 정보 값을 상속합니다. 암호화 등록 정보 값이 공유에 대해 지정되면 상속된 값이 사용되지 않습니다. 암호화된 공유는 암호화되지 않은 프로젝트에 만들 수 있습니다. 암호화되지 않은 공유는 암호화된 프로젝트에 만들 수 없습니다.

암호화 키 관리

주 - 암호화는 특정 모델에 대해 사용이 허가된 기능입니다. 자세한 내용은 "Oracle SLA(소프트웨어 라이선스 계약) 및 통합된 소프트웨어 옵션을 사용하는 하드웨어 시스템에 대한 자격" 및 소프트웨어 릴리스의 라이선스 정보 사용자 설명서를 참조하십시오.

Oracle ZFS 스토리지 어플라이언스는 폴, 프로젝트 및 개별 공유(파일 시스템 및 LUN)에 대해 투명한 데이터 암호화를 제공합니다. 어플라이언스는 내장 로컬 키 저장소를 포함하며, 또한 OKM(Oracle Key Manager) 및 KMIP(키 관리 상호 운용성 프로토콜) 암호화를 지원합니다. 암호화된 각 프로젝트 또는 공유에는 로컬, OKM 또는 KMIP 키 저장소의 래핑 키가 필요합니다. 데이터 암호화 키는 스토리지 어플라이언스에 의해 관리되며 래핑 키에 의해 암호화된 영구적으로 저장됩니다.

암호화된 폴, 프로젝트 또는 공유를 만들려면 암호화 키를 만들어야 합니다.

- 폴이 만들어지기 전 키 저장소가 구성되어야 하므로 초기 시스템 구성 시 또는 공장 초기화 재설정 후 암호화된 폴을 만들 수 없습니다.
- 암호화된 폴의 공유 또는 프로젝트에 대해 복제를 설정하기 전에 소스에서 사용된 암호화 키를 대상에서도 사용할 수 있는지 확인합니다.

다음 표에서는 암호화 키를 관리하는 데 사용할 수 있는 RESTful API 요청에 대해 설명합니다. 테이블에서 *keystore*의 값은 *local*, *okm* 또는 *kmip*입니다. *key*의 값은 *keyname* 등록 정보의 값입니다. 키 등록 정보는 "[암호화 키 나열](#)" [217]에서 설명됩니다.

표 67 암호화 키 작업

요청	경로 <code>/api/storage/v{1 2}</code> 에 추가	설명
GET	<code>/encryption/keystore</code>	모든 <code>keystore</code> 등록 정보를 나열합니다.
PUT	<code>/encryption/keystore</code>	<code>keystore</code> 등록 정보를 수정합니다.
GET	<code>/encryption/keystore/keys</code>	모든 <code>keystore</code> 키를 나열합니다.
GET	<code>/encryption/keystore/keys/key</code>	지정된 키에 대한 세부정보를 가져옵니다.
POST	<code>/encryption/keystore/keys</code>	키를 만듭니다.
DELETE	<code>/encryption/keystore/keys/key</code>	키를 삭제합니다.
GET	<code>/encryption/keystore/keys/key/dependents</code>	이 키에 종속된 공유를 나열합니다.

로컬 키 저장소 구성

로컬 키 저장소를 구성하려면 마스터 문장암호를 설정합니다. 로컬 키 저장소의 경우 `master_passphrase`는 나열하거나 수정할 수 있는 유일한 등록 정보입니다.

`master_passphrase`가 설정되었는지 여부를 확인합니다.

```
GET /api/storage/v2/encryption/local HTTP/1.1
```

출력:

```
HTTP/1.1 200 OK
Content-Type: application/json

{
  "keystore": {
    "href": "/api/storage/v2/encryption/local",
    "master_passphrase": false,
    "keys": []
  }
}
```

`master_passphrase`에 대한 값을 지정합니다.

```
PUT /api/storage/v2/encryption/local HTTP/1.1
Host: zfs-storage.example.com:215
Content-Type: application/json
Accept: application/json
```

```
{
  "master_passphrase": "passphrase"
}
```

`master_passphrase`가 설정되었는지 확인합니다.

```
GET /api/storage/v2/encryption/local HTTP/1.1
```

출력:

```
HTTP/1.1 200 OK
```

```
Content-Type: application/json

{
  "keystore": {
    "href": "/api/storage/v2/encryption/local",
    "master_passphrase": true,
    "keys": []
  }
}
```

OKM 키 저장소 구성

다음 표에서는 OKM 키 저장소를 구성하기 위해 설정해야 하는 등록 정보에 대해 설명합니다.

표 68 OKM 키 저장소 등록 정보

등록 정보	유형	설명
agent_id	string	에이전트 ID입니다.
registration_pin	string	이 값은 OKM 보안 책임자가 제공합니다.
server_addr	string	OKM 서버의 IP 주소입니다.

위 등록 정보가 설정되었는지 여부를 확인합니다.

```
GET /api/storage/v2/encryption/okm HTTP/1.1
```

출력:

```
HTTP/1.1 200 OK
Content-Type: application/json

{
  "keystore": {
    "href": "/api/storage/v2/encryption/okm",
    "agent_id": "",
    "registration_pin": false,
    "server_addr": "",
    "keys": []
  }
}
```

agent_id, registration_pin 및 server_addr에 대한 값을 지정합니다.

```
PUT /api/storage/v2/encryption/okm HTTP/1.1
Host: zfs-storage.example.com:215
Content-Type: application/json
Accept: application/json
```

```
{
  "agent_id": "agent-id",
  "registration_pin": "reg-pin",
  "server_addr": "ipaddr"
}
```

KMIP 키 저장소 구성

KMIP 키 저장소는 Oracle Key Vault를 포함하여 KMIP 호환 서버와 함께 사용됩니다. Oracle Key Vault는 전용 서버에 설치되고 OASIS KMIP 표준을 지원하는 소프트웨어 어플라이언스입니다.

KMIP를 사용하여 암호화를 구성하려면 “키 또는 인증서 업로드” [41]에 설명된 대로 KMIP 관리자로부터 받은 키 및 인증서를 업로드합니다.

키 및 인증서를 업로드한 후 KMIP 서버, 클라이언트 인증서 및 키 이름을 지정합니다.

다음 표에서는 KMIP 키 저장소를 구성하기 위해 설정할 등록 정보에 대해 설명합니다.

표 69 KMIP 키 저장소 등록 정보

등록 정보	유형	설명
server_list	list	KMIP 서버의 IP 주소 또는 호스트 이름입니다. 이 등록 정보는 여러 값을 포함할 수 있습니다.
client_cert	string	KMIP 서버 관리자가 제공한 파일로부터 만든 인증서입니다.
host_match	boolean	서버 인증서의 서버 ID에 대해 서버 호스트 이름을 검증합니다.
destroy_key_on_remove	boolean	어플라이언스에서 키가 삭제될 때 KMIP 서버에서 키를 삭제하거나 보존합니다.

host_match 및 destroy_key_on_remove에 대한 자세한 내용은 [Oracle ZFS Storage Appliance Administration Guide, Release OS8.8.x](#)의 "Key Management Interoperability Protocol (KMIP) Keystore"를 참조하십시오.

위 등록 정보가 설정되었는지 여부를 확인합니다.

```
GET /api/storage/v2/encryption/kmip HTTP/1.1
```

출력:

```
HTTP/1.1 200 OK
Content-Type: application/json

{
  "keystore": {
    "href": "/api/storage/v2/encryption/kmip",
    "server_list": [
      "ipaddr-or-hostname"
    ],
    "client_cert": "134a9138-29a0-4720-80bb-ec2b13457c39",
    "host_match": false,
    "destroy_key_on_remove": true,
    "keys": [],
    ... detailed information about the private key, certificate, and certificate authority ...
  }
}
```

암호화 키 만들기

키 저장소가 구성된 후 키를 만들려는 경우 키 이름을 설정하면 됩니다. 다음 예에서는 새 KMIP 키를 만듭니다. 예제 결과는 “[암호화 키 나열](#)” [217]를 참조하십시오.

요청 예:

```
POST /api/storage/v2/encryption/kmip/keys HTTP/1.1
Host: zfs-storage.example.com:215
Authorization: Basic Tm8gcGV1a2luZyE=
Content-Type: application/json
Accept: application/json
{
  "keyname": "atz-1-27-2021"
}
```

암호화 키 나열

이 명령은 모든 암호화 키의 등록 정보를 나열합니다. 성공한 명령에 대해 HTTP 상태 200 (OK)이 반환됩니다. HTTP 본문에는 JSON 형식의 키 배열이 포함되어 있습니다. 다음 표에서는 암호화 키 등록 정보에 대해 설명합니다.

표 70 암호화 키 등록 정보

등록 정보	유형	설명
cipher	string	AES 암호화 유형입니다.
key	string	(로컬만) 암호화된 형식으로 저장되는 16진수로 인코딩된 원시 256 비트 키입니다. 값을 지정하지 않을 경우 자동으로 이 값이 생성됩니다.
keyname	string	특정 키입니다.
href	string	키에 대한 경로입니다.

다음 예에서는 모든 로컬, OKM 및 KMIP 키를 나열합니다.

요청 예:

```
GET /api/storage/v2/encryption/local/keys /api/storage/v2/encryption/okm/keys /api/storage/v2/encryption/kmip/keys HTTP/1.1
```

결과 예:

이 결과에 따라 이 어플라이언스에 로컬 또는 OKM 키가 없고 KMIP 키가 하나만 있음을 알 수 있습니다.

```
{
  "keys": {
  }
}
{
  "keys": {
```

```
    }  
  }  
  {  
    "keys": [{  
      "cipher": "AES",  
      "keyname": "atz-1-27-2021",  
      "href": "/api/storage/v2/encryption/kmip/keys/key-000"  
    }]  
  }  
}
```

지정된 키로 암호화된 스토리지 나열

dependents 질의는 지정된 키로 암호화된 모든 공유, 프로젝트 또는 풀을 나열합니다.

다음 예에서는 기본 풀의 fs-enc 파일 시스템만 atz-1-27-2021 KMIP 키로 암호화되는 것을 보여줍니다.

요청 예:

```
GET /api/storage/v2/encryption/kmip/keys/atz-1-27-2021/dependents HTTP/1.1
```

결과 예:

```
{  
  "dependents": [  
    "pool-0/local/default/fs-enc"  
  ]  
}
```

키 삭제

키를 삭제하려면 키의 href 등록 정보에 대한 값(키 이름 아님)을 사용합니다. 성공한 삭제는 HTTP 상태 204(No Content)를 반환합니다.

키가 삭제되면 해당 키를 사용하는 모든 풀과 공유의 모든 데이터에 액세스할 수 없게 됩니다. 이 동작은 보안 데이터 삭제와 동일하며 키를 백업하여 키 복원을 준비하지 않는 한, 영구적으로 액세스할 수 없게 되며 이는 되돌릴 수 없습니다.

요청 예:

```
DELETE /api/storage/v2/encryption/kmip/keys/key-000 HTTP/1.1  
Host: zfs-storage.example.com:215  
Accept: application/json
```

암호화된 풀, 프로젝트 또는 공유 만들기

암호화된 풀, 프로젝트 또는 공유를 만들려면 “풀 구성” [151], “프로젝트 만들기” [159], “파일 시스템 만들기” [166] 및 “새 LUN 만들기” [171]에 설명된 필수 등록 정보와 함께 encryption, keystore 및 keyname 등록 정보에 대한 값을 지정합니다.

다음 표에서는 특정 풀, 프로젝트 또는 공유에 대한 암호화 등록 정보를 설명합니다.

표 71 풀, 프로젝트 및 공유에 대한 암호화 등록 정보

등록 정보	유형	설명
encryption	string	AES 암호화 유형 및 키 길이입니다.
keystore	string	키 저장소의 유형: local, okm 또는 kmip.
keyname	string	특정 키 이름입니다.
keylastchanged	string	(읽기 전용) 키가 마지막으로 변경된 날짜입니다. 이 값은 ISO-8601 날짜 시간 형식(v2의 경우) 및 JavaScript 날짜 시간 형식(v1의 경우)입니다. 값이 비어 있을 경우 이 키는 만들어진 이후 변경되지 않은 것입니다.
keystatus	string	(읽기 전용) available, unavailable 또는 none입니다. 이 등록 정보의 값이 unavailable인 경우 키가 삭제된 것입니다.

요청 예:

```
POST /api/storage/v2/pools/p1/projects HTTP/1.1
Host: zfs-storage.example.com:215
Content-Type: application/json
Accept: application/json
```

```
{
  "name": "proj-enc",
  "encryption": "aes-128-ccm",
  "keystore": "local",
  "keyname": "Key-0"
}
```

결과 예:

```
HTTP/1.1 201 Created
Content-Type: application/json
Location: http://zfs-storage.example.com:215/pools/p1/projects/proj-enc
```

```
{
  "project": {
    "name": "proj-enc",
    "href": "/api/storage/v2/pools/p1/projects/proj-enc",
    ...
    "encryption": "aes-128-ccm",
    "keystore": "local",
    "keychangedate": "",
    "keystatus": "available",
    "keyname": "Key-0",
    ...
  }
}
```


시스템 명령

시스템 명령은 시스템 ID 정보를 얻고 최상위 레벨 시스템 관리 명령을 수행하는 데 사용됩니다. 다음 표에는 사용 가능한 시스템 명령이 나와 있습니다.

어플라이언스 시스템 명령

다음 시스템 명령을 사용할 수 있습니다.

표 72 어플라이언스 시스템 명령

요청	경로 <code>/api/system/v{1 2}</code> 에 추가	설명
GET	<code>/version</code>	어플라이언스 하드웨어 및 소프트웨어 버전 정보를 나열합니다.
PUT	<code>/reboot</code>	어플라이언스를 재부트합니다. 이 재부트 중 대기열에 있는 플랫폼 업데이트가 모두 적용됩니다.
PUT	<code>/reboot?skip_update=true</code>	대기열에 있는 플랫폼 업데이트를 적용하지 않고 어플라이언스를 재부트합니다.
PUT	<code>/reboot?diag=true</code>	진단 재부트: 프로세스에서 추가 진단 정보를 수집하여 어플라이언스를 재부트합니다.
PUT	<code>/poweroff</code>	어플라이언스를 끕니다.
PUT	<code>/restart</code>	관리 인터페이스를 다시 시작하고 진단 정보를 수집합니다.
PUT	<code>/factoryreset</code>	어플라이언스 구성을 출하 시 설정으로 다시 재설정합니다.
GET	<code>/disks</code>	모든 시스템 디스크를 나열합니다.
GET	<code>/disks/disk</code>	지정된 시스템 디스크 등록 정보를 나열합니다.
GET	<code>/memory</code>	시스템 메모리 상태 보고서입니다.

버전 가져오기

이 명령은 시스템 ID 정보를 포함하는 시스템 구조를 반환합니다. 성공한 명령에 대해 HTTP 상태 200(OK)이 반환됩니다.

요청 예:

```
GET /api/system/v1/version HTTP/1.1
Host: zfs-storage.example.com:215
Accept: application/json
```

결과 예:

```
HTTP/1.1 200 OK
Content-Type: application/json

{
  "version": {
    "hw_csn": "1211FM2009",
    "updated": "20130528T16:21:17",
    "fw_vendor": "American Megatrends Inc.",
    "os_isa": "i386",
    "os_boot": "20130528T16:25:44",
    "hw_product": "Sun Netra X4270 M3",
    "http_version": "Apache/2.2.24 (Unix)",
    "hw_asn": "2f4aeeb3-b670-ee53-e0a7-d8e0ae410749",
    "ssl_version": "OpenSSL 1.0.0k 5 Feb 2013",
    "os_machine": "i86pc",
    "os_nodename": "admin1",
    "os_version": "nas/generic@2013.05.16,1-0",
    "ak_product": "SUNW,iwashiG2",
    "fw_version": "21000208",
    "os_release": "5.11",
    "installed": "20130411T19:50:16",
    "sp_version": "3.1.2.0",
    "os_platform": "i86pc",
    "fw_release": "10/22/2012"
  }
}
```

시스템 전원 끄기

이 명령은 어플라이언스의 클린 종료를 수행합니다. 어플라이언스가 클러스터의 일부가 아닌 모든 데이터 서비스를 영구적으로 사용할 수 없습니다. 시스템의 전원을 다시 켜려면 전원 스위치에 대한 서비스 프로세서 액세스 또는 물리적 액세스가 필요합니다. 이 명령은 비동기적으로 실행되고 HTTP 상태 202(Accepted)를 반환합니다. 실제 명령의 상태를 따라가도록 어플라이언스를 모니터링해야 합니다.

요청 예:

```
PUT /api/system/v1/poweroff HTTP/1.1
Host: zfs-storage.example.com:215
```

시스템 재부트

이 명령은 어플라이언스의 클린 전원 꺾다가 켜기를 수행합니다. 모든 서비스를 일시적으로 사용할 수 없습니다. 이 명령은 비동기적으로 실행되고 HTTP 상태 202(Accepted)를 반환합니다. 실제 명령의 상태를 따라가도록 어플라이언스를 모니터링해야 합니다.

주 - 보류 중인 플랫폼 업데이트를 어플라이언스에 사용할 수 있을 경우 이 재부트 중 적용됩니다. 보류 중인 플랫폼 업데이트를 적용하지 않고 재부트를 수행하려면 `/reboot?skip_update=true` 명령을 사용합니다.

요청 예:

```
PUT /api/system/v1/reboot HTTP/1.1
Host: zfs-storage.example.com:215
```

요청 예:

```
PUT /api/system/v1/reboot?skip_update=true HTTP/1.1
Host: zfs-storage.example.com:215
```

시스템 관리 다시 시작

이 명령은 관리 인터페이스를 다시 시작하고 진단 정보를 수집합니다. 이 명령은 비동기적으로 실행되고 HTTP 상태 202(Accepted)를 반환합니다. 실제 명령의 상태를 따라가도록 어플라이언스를 모니터링해야 합니다.

요청 예:

```
PUT /api/system/v1/restart HTTP/1.1
Host: zfs-storage.example.com:215
```

진단 재부트

이 명령은 프로세스에서 추가 진단 정보를 수집하여 어플라이언스를 재부트합니다. 이 명령은 비동기적으로 실행되고 HTTP 상태 202(Accepted)를 반환합니다. 실제 명령의 상태를 따라가도록 어플라이언스를 모니터링해야 합니다.

주 - 어플라이언스가 사용할 수 있는 보류 중인 플랫폼 업데이트가 있을 경우 이 진단 재부트 중 적용되지 않습니다.

요청 예:

```
PUT /api/system/v1/reboot?diag=true HTTP/1.1
Host: zfs-storage.example.com :215
```

공장 초기화 재설정

이 명령은 어플라이언스 구성을 원래 출하 시 설정으로 복원합니다. 모든 구성 변경사항이 손실되고 어플라이언스는 처음 설치 당시 초기 설정을 통해 수행되어야 합니다. 이 명령은 비동

기적으로 실행되고 HTTP 상태 202(Accepted)를 반환합니다. 실제 명령의 상태를 따라가도록 어플라이언스를 모니터링해야 합니다. 이 명령을 실행하면 모든 구성 데이터가 손실될 수 있으므로, 명령이 성공하려면 질의 매개변수 `?confirm=true`를 추가해야 합니다.

요청 예:

```
PUT /api/system/v1/factoryreset?confirm=true HTTP/1.1
Host: zfs-storage.example.com:215
```

시스템 지원 번들

다음 지원 번들 명령을 사용할 수 있습니다.

표 73 지원 번들 명령

요청	경로 <code>/api/system/v{1 2}</code> 에 추가	설명
GET	<code>/bundles</code>	모든 지원 번들을 나열합니다.
GET	<code>/bundles/bundle</code>	지정된 번들 데이터 또는 등록 정보를 가져옵니다.
POST	<code>/bundles</code>	지원 번들을 빌드하여 오라클 고객 지원 센터로 업로드합니다.
PUT	<code>/bundles/bundle/retry</code>	지정된 번들의 업로드를 재시도합니다.
PUT	<code>/bundles/bundle/cancel</code>	지정된 번들의 업로드를 취소합니다.
PUT	<code>/bundles/bundle/send</code>	지정된 번들을 선택사항인 SR 번호와 함께 오라클 고객 지원센터로 업로드합니다.
DELETE	<code>/bundles/bundle</code>	지정된 번들을 삭제합니다.

지원 번들 만들기

서비스 요청 해결을 위해 새 지원 번들을 만듭니다. 지원 번들을 미결 서비스 요청과 연관시켜 오라클 고객지원센터로 보내려면 SR(서비스 요청) 번호를 제공해야 합니다. SR 번호는 `3-nnnnnnnnnn` 형식이어야 합니다. 지원 번들을 자동으로 오라클 고객 지원 센터로 업로드하려면 업로드 권한이 있는 유효한 MOS 자격 증명을 사용하여 Phone Home 설정을 등록해야 합니다.

요청 예:

```
POST /api/system/v1/bundles HTTP/1.1
Authorization: Basic Tm8gcGVla2luZyE=
Host: zfs-storage.example.com:215
Accept: application/json
Content-Type: application/json
Content-Length: 23

{"srn": "3-0123456789"}
```

결과 예:

```
HTTP/1.1 201 Created
X-Zfssa-Appliance-API: 1.0
```

SRN(서비스 요청 번호)을 제공하지 않으면 시스템이 대신 로컬 번들을 생성합니다.

요청 예:

```
POST /api/system/v1/bundles HTTP/1.1
Authorization: Basic Tm8gcGVla2luZyE=
Host: zfs-storage.example.com:215
Accept: application/json
Content-Type: application/json
Content-Length: 23
```

결과 예:

```
{
  "bundle": {
    "status": "",
    "uuid": "d4431d57-ba4f-4f37-fa1e-a09fcbf3e56b",
    "associated_bundle": [
      {
        "href": "/api/system/v1/bundles/4050963a-4082-663f-99c0-fee915f2839c"
      }
    ],
    "srn": null,
    "filename": "ak.d4431d57-ba4f-4f37-fa1e-a09fcbf3e56b.tar.gz",
    "href": "/api/system/v1/bundles/d4431d57-ba4f-4f37-fa1e-a09fcbf3e56b",
    "date": "Thu Mar 10 2016 19:38:58 GMT+0000 (UTC)",
    "type": "User initiated"
  }
}
```

지원 번들 나열

이 명령은 시스템에서 처리 중이거나 수집 중인 모든 지원 번들을 나열합니다. 지원 번들이 오라클 고객 지원 센터로 업로드된 후 시스템에서 지원 번들이 제거됩니다.

요청 예:

```
GET /api/system/v1/bundles HTTP/1.1
Authorization: Basic Tm8gcGVla2luZyE=
Host: zfs-storage.example.com:215
Accept: */*
```

결과 예:

```
{
  "bundles": [{
    "status": "building",
    "step_progress": 6.25,
    "srn": "3-0123456789",
    "filename": "/upload/issue/3-0123456789/3-0123456789_ak.ba8ebd55-2349-c31c-cde3-acf3fb0c3389.tar.gz",
  }
]
```

```
        "href": "/api/system/v1/bundles/ba8ebd55-2349-c31c-cde3-acf3fb0c3389",
        "date": "Wed Apr 30 2014 19:31:06 GMT+0000 (UTC)",
        "type": "User initiated",
        "uuid": "ba8ebd55-2349-c31c-cde3-acf3fb0c3389"
    }],
}
```

지원 번들 가져오기

단일 번들에서 등록 정보를 가져옵니다.

요청 예:

```
GET /api/system/v1/bundles/9604155c-928b-cf97-c826-cda9fc17ac57 HTTP/1.1
Authorization: Basic Tm8gcGVla2luZyE=
Host: zfs-storage.example.com:215
Accept: */*
```

결과 예:

```
HTTP/1.1 200 OK
X-Zfssa-Appliance-API: 1.0
Content-Type: application/json
Content-Length: 165

{
  "bundle": {
    "status": "building",
    "step_progress": 62.5,
    "srn": "3-0123456789",
    "filename": "/upload/issue/3-0123456789/3-0123456789_ak.ba8ebd55-2349-c31c-cde3-acf3fb0c3389.tar.gz",
    "href": "/api/system/v1/bundles/ba8ebd55-2349-c31c-cde3-acf3fb0c3389",
    "date": "Wed Apr 30 2014 19:31:06 GMT+0000 (UTC)",
    "type": "User initiated",
    "uuid": "ba8ebd55-2349-c31c-cde3-acf3fb0c3389"
  }
}
```

지원 번들 취소

이 명령은 지원 번들의 자동 업로드를 취소합니다.

요청 예:

```
PUT /api/system/v1/bundles/9aef7c38-073c-603f-f35c-be64e26e90e3/cancel HTTP/1.1
Authorization: Basic Tm8gcGVla2luZyE=
Host: zfs-storage.example.com:215
```

결과 예:

```
HTTP/1.1 202 Accepted
```

X-Zfssa-Appliance-API: 1.0

지원 번들 업로드 재시도

이 명령은 번들을 오라클 고객 지원 센터로 업로드하려고 하는 새 번들 업로드 작업을 만듭니다. 번들 가져오기 명령은 지원 번들 업로드의 상태를 모니터링하는 데 사용될 수 있습니다.

요청 예:

```
PUT /api/system/v1/bundles/9aef7c38-073c-603f-f35c-be64e26e90e3/retry HTTP/1.1
Authorization: Basic Tm8gcGV1a2luZyE=
Host: zfs-storage.example.com:215
```

결과 예:

```
HTTP/1.1 202 Accepted
X-Zfssa-Appliance-API: 1.0
```

다른 SR(서비스 요청) 번호를 사용하여 번들 업로드를 재시도하려면 `send` 명령을 사용합니다. SR(서비스 요청) 번호를 제공하지 않으면 시스템이 원래 SR 번호를 사용하여 업로드를 재시도합니다.

주 - 로컬로 생성된 번들에 대해 `send`를 실행하려면 SR 번호가 필요합니다. SR 번호가 없으면 오류가 발생합니다.

요청 예:

```
PUT /api/system/v1/bundles/9aef7c38-073c-603f-f35c-be64e26e90e3/send HTTP/1.1
Authorization: Basic Tm8gcGV1a2luZyE=
Host: zfs-storage.example.com:215
```

```
{"srn": "3-0123456789"}
```

결과 예:

```
HTTP/1.1 202 Accepted
X-Zfssa-Appliance-API: 1.0
```

지원 번들 업로드

오라클 고객 지원 센터로 자동으로 업로드되지 않는 지원 번들은 수동으로 업로드될 수 있습니다.

주 - 로컬로 생성된 번들에 대해 `send`를 실행하려면 SR 번호가 필요합니다. SR 번호가 없으면 오류가 발생합니다.

요청 예:

```
PUT /api/system/v1/bundles/9aef7c38-073c-603f-f35c-be64e26e90e3/send HTTP/1.1
Authorization: Basic Tm8gcGVla2luZyE=
Host: zfs-storage.example.com:215

{"srn": "3-0123456789"}
```

결과 예:

```
HTTP/1.1 202 Accepted
X-Zfssa-Appliance-API: 1.0
```

지원 번들 삭제

이 명령은 어플라이언스에서 지원 번들을 제거합니다.

요청 예:

```
DELETE /api/system/v1/bundles/9aef7c38-073c-603f-f35c-be64e26e90e3 HTTP/1.1
Authorization: Basic Tm8gcGVla2luZyE=
Host: zfs-storage.example.com:215
```

결과 예:

```
HTTP/1.1 204 No Content
X-Zfssa-Appliance-API: 1.0
```

시스템 업데이트

이러한 명령은 시스템 업데이트 이미지를 관리합니다.

표 74 업데이트 명령

요청	경로 <code>/api/system/v{1 2}</code> 에 추가	설명
GET	<code>/updates</code>	모든 시스템 업데이트를 나열합니다.
GET	<code>/updates/update</code>	지정된 시스템 업데이트 등록 정보를 가져옵니다.
GET	<code>/update/platform</code>	플랫폼 펌웨어(컨트롤러의 서비스 프로세서 및 시스템 보드 펌웨어를 칭함)의 업데이트 상태를 표시합니다.
GET	<code>/update/firmware</code>	구성요소 펌웨어(디스크 및 SSD 펌웨어와 디스크 인클로저 IOM 펌웨어를 칭함)의 업데이트 상태를 표시합니다.
PUT	<code>/updates/update</code>	업데이트 설정 수정
PUT	<code>/updates/update/upgrade</code>	지정된 업데이트 이미지로 업그레이드합니다.
PUT	<code>/updates/update/check</code>	지정된 업데이트 이미지에 대한 업그레이드 상태 검사를 실행합니다.

요청	경로 /api/system/v{1 2}에 추가	설명
PUT	/updates/update/rollback	지정된 업데이트 이미지로 롤백합니다.
PUT	/updates-apply	호환되지 않는 지연 업데이트를 적용합니다.
DELETE	/updates/update	지정된 시스템 업데이트를 삭제합니다.
POST	/updates	업데이트 이미지를 어플라이언스로 로드합니다.

표 75 시스템 업데이트 등록 정보

등록 정보	유형	설명
version	String	매체 버전 업데이트
release_date	DateTime	릴리스 날짜 업데이트
install_date	DateTime	최신 설치 날짜를 업데이트합니다. 설치되지 않은 경우 어플라이언스로 다운로드한 날짜를 업데이트합니다.
status	String	매체 상태 업데이트(변경할 수 없음)
update_deferred	ChooseOne	지연 설정(onreboot 또는 onrequest)

지연 업데이트 통지:

The following updates enable features that are incompatible with earlier software versions. As these updates cannot be reverted once committed, and peer system resources are updated across a cluster, verifying first that the system upgrade is functioning properly before applying deferred updates is advised.

시스템 업데이트 나열

시스템 업데이트를 가져오는 요청 예:

```
GET /api/system/v1/updates HTTP/1.1
Authorization: Basic Tm8gcGV1a2luZyE=
Host: zfs-storage.example.com:215
Accept: application/json
```

결과 예:

```
HTTP/1.1 200 OK
X-Zfssa-Appliance-API: 1.0
Content-Length: 541
Content-Type: application/json
```

```
{
  "updates": [
    {
      "release_date": "Tue Aug 13 2013 17:47:32 GMT+0000 (UTC)",
      "install_date": "Wed Aug 14 2013 12:33:08 GMT+0000 (UTC)",
      "href": "/api/system/v1/updates/nas@2013.08.13,1-0",
      "status": "previous",
      "version": "2013.08.13,1-0"
    }
  ],
}
```

```
{
  "release_date": "Sat Aug 24 2013 17:54:23 GMT+0000 (UTC)",
  "install_date": "Sun Aug 25 2013 11:30:14 GMT+0000 (UTC)"
  "href": "/api/system/v1/updates/nas@2013.08.24,1-0",
  "status": "current",
  "version": "2013.08.24,1-0"
},
{
  "release_date": "Sun Aug 25 2013 12:56:57 GMT+0000 (UTC)",
  "install_date": "Mon Aug 26 2013 18:50:33 GMT+0000 (UTC)"
  "href": "/api/system/v1/updates/nas@2013.08.25,1-0",
  "status": "waiting",
  "version": "2013.08.25,1-0"
}
}]
}
```

시스템 업데이트 가져오기

단일 업데이트 이미지에 대한 등록 정보를 가져옵니다.

요청 예:

```
GET /api/system/v1/updates/nas@2013.08.25,1-0 HTTP/1.1
Authorization: Basic Tm8gcGVla2luZyE=
Host: zfs-storage.example.com:215
Accept: application/json
```

결과 예:

```
HTTP/1.1 200 OK
X-Zfssa-Appliance-API: 1.0
Content-Length: 541
Content-Type: application/json
```

```
{
  "update": {
    "release_date": "Sat Aug 24 2013 17:54:23 GMT+0000 (UTC)",
    "install_date": "Sun Aug 25 2013 11:30:14 GMT+0000 (UTC)"
    "href": "/api/system/v1/updates/nas@2013.08.24,1-0",
    "status": "current",
    "version": "2013.08.24,1-0",
    "update_deferred": "on_request"
  }
}
```

플랫폼 펌웨어 업데이트 상태 가져오기

보류 중인 플랫폼 펌웨어 업데이트에 대한 업데이트 상태를 가져옵니다. 플랫폼 펌웨어는 컨트롤러의 서비스 프로세서 및 시스템 보드 펌웨어를 총칭하는 용어입니다.

요청 예:

```
GET /api/system/v1/update/platform HTTP/1.1
```

```
Authorization: Basic Tm8gcGV1a2luZyE=
Host: zfs-storage.example.com:215
Accept: application/json
```

결과 예:

```
HTTP/1.1 200 OK
X-Zfssa-Appliance-API: 1.0
Content-Length: 541
Content-Type: application/json

{
  "platform": {
    "href": "/api/system/v1/update/platform",
    "power_down_needed": true,
    "update_needed": "true"
  }
}
```

구성요소 펌웨어 업데이트 상태 가져오기

보류 중, 실패 및 진행 중 상태의 구성요소 펌웨어 업데이트 수를 가져옵니다. 구성요소 펌웨어는 디스크 및 SSD 펌웨어와 디스크 인클로저 IOM 펌웨어를 총칭하는 용어입니다.

요청 예:

```
GET /api/system/v1/update/firmware HTTP/1.1
Authorization: Basic Tm8gcGV1a2luZyE=
Host: zfs-storage.example.com:215
Accept: application/json
```

결과 예:

```
HTTP/1.1 200 OK
X-Zfssa-Appliance-API: 1.0
Content-Length: 541
Content-Type: application/json

{
  "firmware": {
    "href": "/api/system/v1/update/firmware",
    "upgrades_pending": 0,
    "upgrades_failed": 0,
    "upgrades_in_progress": 0
  }
}
```

시스템 업데이트 업로드

이 명령은 새 시스템 업데이트 이미지를 업로드합니다.

curl을 사용한 명령 업로드 예:

```
$ curl --user root:root-password -k --data-binary @nas@2013.08.24,1-0.pkg.gz \
--header "Content-Type: application/octet-stream" \
https://zfs-storage.example.com/api/system/v1/updates
```

이미지가 업로드되고 압축 해제된 이후 이미지 업데이트의 등록 정보가 반환됩니다. 성공 시 HTTP 상태가 201(Created)로 설정되고 위치 헤더에서 새 이미지의 상대적 위치가 반환됩니다.

결과 예:

```
HTTP/1.1 201 Created
X-Zfssa-Appliance-API: 1.0
Content-Length: 541
Content-Type: application/json
Location: /api/system/v1/updates/nas@2013.08.24,1-0

{
  "update": {
    "release_date": "Sat Aug 24 2013 17:54:23 GMT+0000 (UTC)",
    "install_date": "Sun Aug 25 2013 11:30:14 GMT+0000 (UTC)",
    "href": "/api/system/v1/updates/nas@2013.08.24,1-0",
    "status": "current",
    "version": "2013.08.24,1-0",
    "update_deferred": "on_request"
  }
}
```

업그레이드

이 명령은 업데이트 이미지를 로드하고 어플라이언스를 지정된 업데이트 이미지로 재부트합니다. 지정된 이미지 상태는 "previous"와 동일해야 합니다. 그렇지 않으면 명령이 실패합니다.

요청 예:

```
PUT /api/system/v1/updates/nas@2013.08.25,1-0/upgrade
Host: zfs-storage.example.com:215
Authorization: Basic Tm8gcGVla2luZyE=
Content-Length: 0
```

결과 예:

```
HTTP/1.1 202 Accepted
X-Zfssa-Appliance-API: 1.0
```

롤백

롤백은 어플라이언스를 이전 업데이트 이미지로 재부트합니다.

요청 예:

```
PUT /api/system/v1/updates/nas@2013.08.24,1-0/rollback
```

```
Host: zfs-storage.example.com:215
Authorization: Basic Tm8gcGV1a2luZyE=
Content-Length: 0
```

결과 예:

```
HTTP/1.1 202 Accepted
X-Zfssa-Appliance-API: 1.0
```

업데이트 이미지 삭제

어플라이언스에서 사용되지 않는 업데이트 이미지를 제거합니다.

요청 예:

```
DELETE /api/system/v1/updates/nas@2013.08.13,1-0 HTTP/1.1
Host: zfs-storage.example.com:215
Authorization: Basic Tm8gcGV1a2luZyE=
```

결과 예:

```
HTTP/1.1 204 No Content
X-Zfssa-Appliance-API: 1.0
```


RESTful API 사용자 서비스

RESTful API 사용자 서비스는 어플라이언스의 사용자를 구성합니다.

사용자 서비스 명령

다음 사용자 서비스 명령을 사용할 수 있습니다.

표 76 사용자 서비스 명령

요청	경로 <i>/api/user/v{1 2}</i> 에 추가	설명
GET	<i>/api/user/v{1 2}</i> 만 사용	사용자 서비스 명령을 나열합니다.
GET	<i>/users</i>	모든 어플라이언스 사용자 나열
GET	<i>/users/user</i>	지정된 사용자의 등록 정보를 나열합니다.
POST	<i>/users</i>	새 어플라이언스 사용자를 만듭니다.
PUT	<i>/users/user</i>	지정된 사용자의 등록 정보를 수정합니다.
DELETE	<i>/users/user</i>	지정된 사용자를 어플라이언스에서 제거합니다.
GET	<i>/users/user/preferences</i>	지정된 사용자에 대한 환경 설정 등록 정보를 나열합니다.
PUT	<i>/users/user/preferences</i>	지정된 사용자에 대한 환경 설정 등록 정보를 수정합니다.
GET	<i>/users/user/exceptions</i>	지정된 사용자에 대한 모든 권한 부여 예외사항을 나열합니다.
GET	<i>/users/user/exceptions/auth</i>	지정된 사용자에 대한 <i>auth</i> 권한 부여를 나열합니다.
POST	<i>/users/user/exceptions</i>	지정된 사용자에 대한 새 권한 부여 예외사항을 만듭니다.
PUT	<i>/users/user/exceptions/auth</i>	지정된 사용자에 대한 지정된 권한 부여를 수정합니다.
DELETE	<i>/users/user/exceptions/auth</i>	지정된 사용자에 대한 지정된 권한 부여를 삭제합니다.
GET	<i>/users/user/preferences/keys</i>	지정된 사용자에 대한 모든 SSH 키를 나열합니다.
GET	<i>/users/user/preferences/keys/key</i>	지정된 사용자에 대한 지정된 SSH 키의 등록 정보를 나열합니다.
POST	<i>/users/user/preferences/keys</i>	지정된 사용자에 대한 새 SSH 키를 만듭니다.

요청	경로 <i>/api/user/v{1 2}</i> 에 추가	설명
PUT	<i>/users/user/preferences/keys/key</i>	지정된 사용자에게 대한 지정된 SSH 키를 수정합니다.
DELETE	<i>/users/user/preferences/keys/key</i>	지정된 사용자에게 대한 지정된 SSH 키를 삭제합니다.
GET	<i>/users/user/preferences/tokens</i>	지정된 사용자에게 대한 모든 REST 로그인 토큰을 나열합니다.
GET	<i>/users/user/preferences/tokens?token=token</i>	토큰 값으로 REST 로그인 토큰을 나열합니다.
GET	<i>/users/user/preferences/tokens/token-id</i>	토큰 ID로 REST 로그인 토큰을 나열합니다.
POST	<i>/users/user/preferences/tokens</i>	지정된 사용자에게 대한 REST 로그인 토큰을 만듭니다.
DELETE	<i>/users/user/preferences/tokens?token=token</i>	토큰 값으로 REST 로그인 토큰을 제거합니다.
DELETE	<i>/users/user/preferences/tokens/token-id</i>	토큰 ID로 REST 로그인 토큰을 제거합니다.

사용자 서비스 등록 정보

사용자 이름 및 암호와 함께 사용자 서비스 등록 정보는 특성(예: 사용자에게 부여된 권한, 사용자에게 적용되는 제한사항 및 사용자의 로케일)을 정의합니다.

사용자 등록 정보

사용자 등록 정보는 다음과 같을 수 있습니다. 일부 등록 정보는 특정 유형의 사용자에게 대해서만 사용할 수 있습니다.

표 77 사용자 등록 정보

등록 정보	유형	설명
logname	string	고유 사용자 이름입니다. logname은 사용자가 만들어진 후 변경할 수 없습니다.
type	string	사용자 유형(local, directory, data, nologin)입니다. type은 사용자가 만들어진 후 변경할 수 없습니다. 기본 값: local
uid	number	사용자 ID입니다. 사용자 ID를 지정하거나 시스템의 사용자 ID 지정을 허용할 수 있습니다. 사용자 ID를 지정하는 경우 사용자 ID는 100보다 작을 수 없으며, 2147483646보다 클 수 없고, 60001, 60002 또는 65534와 같을 수 없습니다. uid는 사용자가 만들어진 후 변경할 수 없습니다.
fullname	string	사용자의 전체 이름 또는 실제 이름입니다. BUI에서 전체 이름이 대시보드 상단의 Logout(로그아웃) 버튼 왼쪽에 표시되며 브라우저 탭에도 표시될 수 있습니다. 기본 값: logname과 동일
initial_password	string	로컬 및 데이터 사용자에게 대한 암호입니다.

등록 정보	유형	설명
require_annotation	boolean	<p>true인 경우</p> <ul style="list-style-type: none"> ■ BUI – 초기 BUI 페이지를 표시하기 전에 사용자에게 설명을 입력하도록 합니다. ■ CLI – CLI 프롬프트를 표시하기 전에 사용자에게 설명을 입력하도록 합니다. ■ REST – 요청이 권한 부여되지 않은 상태로 실패합니다. <p>세션 주석은 감사 로그에 나타납니다.</p>
roles	list	디렉토리 또는 로컬 사용자에게 지정된 역할입니다.
kiosk_mode	boolean	<p>true인 경우 이 사용자는 키오스크 사용자입니다.</p> <ul style="list-style-type: none"> ■ BUI – 사용자가 <code>kiosk_screen</code> 등록 정보 값인 화면만 볼 수 있도록 제한됩니다. ■ CLI – 로그인을 실패합니다. ■ REST – 요청이 권한 부여되지 않은 상태로 실패합니다.
kiosk_screen	string	<code>kiosk_mode</code> 가 true인 경우 이 사용자가 제한되는 BUI 화면입니다. 기본값: <code>status/dashboard</code>
exceptions	list	디렉토리 사용자나 로컬 사용자에게 지정된 추가 권한 부여 또는 역할에서 지정된 권한 부여에 대한 제한사항입니다.
preferences	list	사용자 환경 설정(예: 로케일, BUI 시작 페이지, 시간 초과, SSH 공개 키 및 REST 로그인 토큰)입니다.

해당 등록 정보에 대한 추가 설명은 다음 설명서를 참조하십시오.

- [“Understanding Users and Roles” in Oracle ZFS Storage Appliance Administration Guide, Release OS8.8.x](#)
- [“User Properties” in Oracle ZFS Storage Appliance Administration Guide, Release OS8.8.x](#)

사용자 역할 및 예외사항

`roles` 등록 정보를 사용하여 사용자에게 기존 역할을 지정할 수 있습니다. 역할을 만들려면 [RESTful API 역할 서비스 \[249\]](#)를 참조하십시오.

`exceptions` 등록 정보를 통해 사용자에게 대한 권한 부여를 추가할 수 있습니다. 이 방식으로 권한 부여를 제거할 수도 있습니다. 예를 들어, 특정 권한을 부여하는 역할이 사용자에게 지정된 경우 예외사항 목록에서 개별 권한 부여를 `false`로 설정할 수 있습니다. 이때 해당 역할에 포함된 다른 권한 부여는 유지됩니다.

다음 요청은 사용자 `user1`에 대한 Kerberos 서비스를 다시 시작할 수 있는 권한 부여를 추가합니다.

POST /api/user/v1/users/user1/exceptions HTTP/1.1

```
{
  "scope": "svc",
  "service": "kerberos",
  "allow_restart": true
}
```

다음 결과는 user1에 대한 Kerberos 서비스에 사용 가능한 모든 권한 부여를 보여줍니다. 이 예에서 기타 권한 부여의 기본값은 false입니다.

```
{
  "auth": {
    "href": "/api/user/v1/users/user1/exceptions/auth-001",
    "scope": "svc",
    "service": "kerberos",
    "allow_administer": false,
    "allow_configure": false,
    "allow_restart": true
  }
}
```

사용자 환경 설정 등록 정보

로그인한 사용자 또는 allow_changePreferences 권한이 부여된 사용자에 대한 환경 설정을 지정할 수 있습니다. allow_changePreferences 권한 부여를 얻으려면 [“사용자 역할 및 예외사항” \[237\]](#)을 참조하십시오.

표 78 사용자 환경 설정 등록 정보

등록 정보	유형	설명
locale	string	지역성입니다. 기본값: C
login_screen	string	페이지가 URL에서 지정되지 않은 경우 로그인을 성공할 때 표시되는 BUI 페이지입니다. 기본값: status/dashboard
session_timeout	integer	사용자가 BUI에서 벗어난 후 브라우저에서 자동으로 세션이 로그아웃될 때까지의 시간(분)입니다. 기본값: 15
cli_idle_timeout	integer	세션이 종료되기 전 CLI가 유휴 상태일 수 있는 기간(초)입니다. 기본값인 0은 유휴 상태에서 CLI가 자동으로 닫히지 않음을 의미합니다.
advanced_analytics	string	고급 Analytics 통계 사용
keys	list	RSA/DSA 공개 키입니다
tokens	list	REST 로그인 토큰

해당 등록 정보에 대한 자세한 내용은 다음 절을 참조하십시오.

- [“CLI 시간 초과” \[239\]](#)
- [“SSH 키” \[239\]](#)
- [“REST 로그인 토큰” \[239\]](#)

CLI 시간 초과

기본적으로 CLI가 유휴 상태일 수 있는 기간에는 제한이 없습니다(`cli_idle_timeout` 값이 -1임). CLI가 유휴 상태일 수 있는 기간에 대한 제한을 설정하려면 `cli_idle_timeout` 초에 대해 양의 정수 값을 설정합니다. 시간 초과 제한에 도달하면 CLI가 닫힙니다.

다음 예에서는 CLI 시간 초과를 1시간으로 설정합니다.

```
PUT /api/user/v1/users/user1/preferences HTTP/1.1
{ "cli_idle_timeout": 3600 }
{
  "preferences": {
    "href": "/api/user/v1/users/user1/preferences",
    "locale": "C",
    "login_screen": "configuration/preferences",
    "session_timeout": 15,
    "cli_idle_timeout": 3600,
    "advanced_analytics": false,
    "keys": [],
    "tokens": []
  }
}
```

시간 초과를 사용 안함으로 설정하려면 `cli_idle_timeout`의 값을 -1로 설정하거나 다음 예에서와 같이 `unset`를 사용합니다.

```
PUT /api/user/v1/users/user1/preferences HTTP/1.1
{ "<unset>": ["cli_idle_timeout"] }
```

`unset`를 꺾쇠 괄호로 묶으면 `unset`라는 등록 정보가 있어도 문제가 발생하지 않습니다.

SSH 키

SSH 공개 키를 사용하면 암호를 사용하지 않고 SSH 연결을 허용할 수 있습니다.

표 79 SSH 키 등록 정보

등록 정보	유형	설명
<code>type</code>	string	SSH 키 유형: RSA 또는 DSA
<code>key</code>	string	SSH 키 내용
<code>comment</code>	string	이 SSH 키와 연관된 설명

REST 로그인 토큰

지속적 또는 비지속적 REST 로그인 토큰을 만들고 반환된 토큰 값 또는 토큰 ID를 통해 토큰에 액세스할 수 있습니다.

표 80 REST 로그인 토큰 등록 정보

등록 정보	유형	설명
name	string	토큰 이름입니다.
token_username	string	이 로그인 토큰을 사용할 수 있는 사용자의 이름입니다. 이 값은 요청 경로에서 설정됩니다.
preserve	boolean	false인 경우 기본 expiration 값을 사용합니다. true인 경우 사용자정의 expiration 값을 설정합니다. 기본값: false
expiration	number	토큰을 만드는 경우 이 값은 토큰이 만료될 때까지의 시간(초)입니다. preserve가 true로 설정된 경우 이 등록 정보를 설정해야 합니다. preserve가 false로 설정된 경우 expiration의 값은 900입니다. 토큰 등록 정보를 나열하는 경우 이 값은 해당 토큰이 만료될 날짜 및 시간입니다.

REST 로그인 토큰 질의 매개변수

GET 명령에서 `token=token` 질의 매개변수를 사용하여 해당 토큰에 대한 등록 정보 값을 표시할 수 있습니다. DELETE 명령에서 `token=token` 질의 매개변수를 사용하여 해당 토큰을 제거할 수 있습니다. `token`은 토큰 만들기 응답에서 `x-Auth-Session`의 값입니다.

`token`은 토큰이 만들어질 때 표시된 후 다시 표시되지 않습니다. `token`을 복사하여 보관해야 합니다. 예 3. “REST 로그인 토큰 만들기”를 참조하십시오.

사용자 나열

다음 예에서 `root` 사용자는 이 어플라이언스에 대해 로컬이며 `user1` 사용자는 LDAP, NIS 또는 AD 사용자입니다. `directory` 유형 사용자의 경우 사용자 ID, 전체 이름 및 암호는 디렉토리 서비스에서 가져온 것이며 암호 등록 정보가 나열되지 않습니다. “[Understanding Users and Roles](#)” in *Oracle ZFS Storage Appliance Administration Guide, Release OS8.8.x*에서 사용자 유형에 대한 자세한 내용을 참조하십시오.

요청 예:

```
GET /api/user/v1/users HTTP/1.1
```

결과 예:

```
{
  "users": [
    {
      "logname": "root",
      "type": "local",
      "uid": 0,
      "fullname": "Super-User",
```

```

        "initial_password": true,
        "require_annotation": false,
        "href": "/api/user/v2/users/root"
    }, {
        "logname": "user1",
        "type": "directory",
        "uid": user1uid,
        "fullname": "Real Name",
        "require_annotation": false,
        "roles": ["basic"],
        "kiosk_mode": false,
        "kiosk_screen": "status/dashboard",
        "href": "/api/user/v2/users/user1"
    }
}

```

특정 사용자 나열

특정 사용자에 대한 정보에는 사용자 환경 설정 및 권한 부여 예외사항이 포함됩니다. 이 예에서 user1에게는 user1에게 경보 구성 및 게시 권한을 부여하는 auth-000이라는 권한 부여 예외사항이 있습니다. user1에 대한 환경 설정은 모두 기본값입니다.

요청 예:

```
GET /api/user/v1/users/user1 HTTP/1.1
```

결과 예:

```

{
  "user": {
    "href": "/api/user/v2/users/user1",
    "logname": "user1",
    "type": "directory",
    "uid": user1uid,
    "fullname": "Real Name",
    "require_annotation": false,
    "roles": ["basic"],
    "kiosk_mode": false,
    "kiosk_screen": "status/dashboard",
    "exceptions": [
      {
        "scope": "alert",
        "allow_configure": true,
        "allow_post": true,
        "href": "/api/user/v2/users/user1/exceptions/auth-000"
      }
    ],
    "preferences": {
      "href": "/api/user/v2/users/user1/preferences",
      "locale": "C",
      "login_screen": "status/dashboard",
      "session_timeout": 15,
      "cli_idle_timeout": -1,
      "advanced_analytics": false,
      "keys": [],
      "tokens": []
    }
  }
}

```

사용자 만들기

새 사용자를 만들려면 최소한 사용자 이름(logname)을 제공해야 합니다. 유형을 지정하지 않을 경우 새 사용자의 유형은 local이 됩니다. 다음과 같은 사용자 유형 목록에 설명된 대로 사용자의 유형에 따라 다른 등록 정보가 필요합니다.

- `directory` – 사용자 이름은 기존 NIS, LDAP 또는 AD 사용자여야 합니다. UID, 암호 및 전체 이름을 NIS, LDAP 또는 AD에서 가져옵니다.
- `local` 및 `data` – 사용자 이름 및 암호를 지정합니다. UID를 지정할 수 있습니다. 그렇지 않으면 UID가 자동으로 지정됩니다.
- `nologin` – 사용자 이름을 지정합니다. UID를 지정할 수 있습니다. 그렇지 않으면 UID가 자동으로 지정됩니다.

사용자 및 사용자 유형에 대한 자세한 내용은 [“Understanding Users and Roles” in Oracle ZFS Storage Appliance Administration Guide, Release OS8.8.x](#)를 참조하십시오.

예 1 로컬 사용자 만들기

요청 예:

```
POST /api/user/v1/users HTTP/1.1
```

```
{
  "logname": "test_user",
  "initial_password": "password"
}
```

결과 예:

```
{
  "user": {
    "href": "/api/user/v1/users/test_user",
    "logname": "test_user",
    "type": "local",
    "uid": 2000000002,
    "fullname": "test_user",
    "initial_password": true,
    "require_annotation": false,
    "roles": ["basic"],
    "kiosk_mode": false,
    "kiosk_screen": "status/dashboard",
    "exceptions": [],
    "preferences": {
      "href": "/api/user/v1/users/test_user/preferences",
      "locale": "C",
      "login_screen": "status/dashboard",
      "session_timeout": 15,
      "cli_idle_timeout": -1,
      "advanced_analytics": false,
      "keys": [],
      "tokens": []
    }
  }
}
```

예 2 사용자 복제

기존 사용자와 유형이 동일하고 지정된 역할 및 권한 부여가 동일한 새 사용자를 만들려면 다음 등록 정보를 지정합니다.

- user – 복제하려는 사용자의 사용자 이름입니다.
- clonename – 복제된 새 사용자의 사용자 이름입니다.
- password – 복제된 새 사용자의 초기 암호입니다.

요청 예:

POST /api/user/v1/users HTTP/1.1

```
{
  "user": "test_user",
  "clonename": "clone_user",
  "password": "password"
}
```

결과 예:

```
{
  "user":
  {
    "href": "/api/user/v1/users/clone_user",
    "logname": "clone_user",
    "type": "local",
    "uid": 2000000003,
    "fullname": "clone_user",
    "initial_password": true,
    "require_annotation": false,
    "roles": ["basic"],
    "kiosk_mode": false,
    "kiosk_screen": "status/dashboard",
    "exceptions": [],
    "preferences": {
      "href": "/api/user/v1/users/clone_user/preferences",
      "locale": "C",
      "login_screen": "status/dashboard",
      "session_timeout": 15,
      "cli_idle_timeout": -1,
      "advanced_analytics": false,
      "keys": [],
      "tokens": []
    }
  }
}
```

사용자 등록 정보 수정

직접 사용자 등록 정보를 수정합니다. logname, type 및 uid는 변경할 수 없습니다.

역할, 예외사항, 환경 설정, SSH 키 또는 로그인 토큰을 추가하거나 수정, 삭제하려면 다음 설명서를 참조하십시오.

- [RESTful API 역할 서비스 \[249\]](#)

■ “사용자 서비스 등록 정보” [236]

요청 예:

```
PUT /api/user/v1/users/test_user HTTP/1.1
{"fullname": "Test A. User", "require_annotation": true}
```

결과 예:

```
{
  "user": {
    "href": "/api/user/v1/users/test_user",
    "logname": "test_user",
    "type": "local",
    "uid": 2000000002,
    "fullname": "Test A. User",
    "initial_password": true,
    "require_annotation": true,
    "roles": ["basic"],
    "kiosk_mode": false,
    "kiosk_screen": "status/dashboard",
    "exceptions": [],
    "preferences": {
      "href": "/api/user/v2/users/test_user/preferences",
      "locale": "C",
      "login_screen": "status/dashboard",
      "session_timeout": 15,
      "cli_idle_timeout": -1,
      "advanced_analytics": false,
      "keys": [],
      "tokens": []
    }
  }
}
```

사용자 삭제

시스템에서 사용자를 삭제합니다.

요청 예:

```
DELETE /api/user/v1/users/clone_user HTTP/1.1
```

결과 예:

```
HTTP/1.1 204 No Content
```

토큰 관리

지속적 및 비지속적 REST 로그인 토큰을 만들고, 토큰의 등록 정보를 확인하고, 토큰을 삭제할 수 있습니다. REST 로그인 토큰의 등록 정보는 토큰이 만들어진 후 읽기 전용입니다. 반환된 토큰 값 또는 토큰 ID를 통해 토큰에 액세스할 수 있습니다.

예 3 REST 로그인 토큰 만들기

REST 로그인 토큰을 만들려면 사용자 이름 및 암호가 필요합니다.

토큰 `name`을 설정합니다. 기본적으로 `preserve`의 값은 `false`이며 `expiration`의 값은 900입니다. `preserve`를 `true`로 설정한 경우 `expiration`을 시간(초)으로 설정해야 합니다.

결과에서 `X-Auth-Session`의 값을 저장해야 합니다. 자세한 내용은 ["REST 로그인 토큰" \[239\]](#)을 참조하십시오.

요청 예:

```
POST /api/user/v2/users/test_user/preferences/tokens HTTP/1.1
Host: zfs-storage.example.com:215
X-Auth-User: test_user
X-Auth-Key: password-xxx
Content-Type: application/json
```

```
{"name": "Test Token"}
```

결과 예:

```
HTTP/1.1 201 Created
...
X-Auth-Session: JjZJsZrVQfbZULyAuvSJJTftnBHCcQT

{
  "token": {
    "href":
"/api/user/v2/users/test_user/preferences/tokens/fb65a127-a04c-4f58-bc52-efa884447efb",
    "name": "Test Token",
    "token_username": "test_user",
    "preserve": false,
    "expiration": "2020-04-30T02:33:44Z",
    "id": "fb65a127-a04c-4f58-bc52-efa884447efb"
  }
}
```

요청 예:

```
POST /api/user/v2/users/test_user/preferences/tokens HTTP/1.1
Host: zfs-storage.example.com:215
X-Auth-User: test_user
X-Auth-Key: password-xxx
Content-Type: application/json
```

```
{"name": "Another Token", "preserve": true, "expiration": 3600}
```

결과 예:

```
HTTP/1.1 201 Created
...
X-Auth-Session: pviHrthBGQhGZHoPuqxFQRDccPZgwEK

{
  "token": {
    "href":
"/api/user/v2/users/test_user/preferences/tokens/21f981ad-6221-4fb4-a7d1-dd5560256dfb",
```

```

        "name": "Another Token",
        "token_username": "test_user",
        "preserve": true,
        "expiration": "2020-04-30T03:20:31Z",
        "id": "21f981ad-6221-4fb4-a7d1-dd5560256dfb"
    }
}

```

예 4 지정된 사용자의 모든 토큰 나열

요청 예:

```

GET /api/user/v2/users/test_user/preferences/tokens HTTP/1.1
X-Auth-Session: pviHrthBGQhGZHoPuqxFQrDcCPZgwEK

```

결과 예:

```

{
  "tokens": [{
    "name": "Another Token",
    "token_username": "test_user",
    "preserve": true,
    "expiration": "2020-04-30T03:20:31Z",
    "id": "21f981ad-6221-4fb4-a7d1-dd5560256dfb",
    "href": "/api/user/v2/users/testuser1/preferences/tokens/21f981ad-6221-4fb4-a7d1-dd5560256dfb"
  }, {
    "name": "Test Token",
    "token_username": "test_user",
    "preserve": false,
    "expiration": "2020-04-30T02:33:44Z",
    "id": "fb65a127-a04c-4f58-bc52-efa884447efb",
    "href": "/api/user/v2/users/testuser1/preferences/tokens/fb65a127-a04c-4f58-bc52-efa884447efb"
  }]
}

```

예 5 토큰 값으로 특정 토큰 나열

요청 예:

```

GET /api/user/v2/users/test_user/preferences/tokens?token=pviHrthBGQhGZHoPuqxFQrDcCPZgwEK HTTP/1.1
X-Auth-Session: pviHrthBGQhGZHoPuqxFQrDcCPZgwEK

```

결과 예:

```

{
  "token": {
    "href":
"/api/user/v2/users/test_user/preferences/tokens/21f981ad-6221-4fb4-a7d1-dd5560256dfb",
    "name": "Another Token",
    "token_username": "test_user",
    "preserve": true,
    "expiration": "2020-04-30T03:20:31Z",
    "id": "21f981ad-6221-4fb4-a7d1-dd5560256dfb"
  }
}

```

예 6 토큰 ID로 특정 토큰 나열

```
GET /api/user/v2/users/test_user/preferences/tokens/21f981ad-6221-4fb4-a7d1-dd5560256dfb
HTTP/1.1
X-Auth-Session: pviHrthBGQhGZHoPuqxFQrDcCPZgwEK
```

결과는 위와 동일합니다.

예 7 토큰 값으로 토큰 삭제

요청 예:

```
DELETE /api/user/v2/users/test_user/preferences/tokens?
token=pviHrthBGQhGZHoPuqxFQrDcCPZgwEK HTTP/1.1
X-Auth-Session: pviHrthBGQhGZHoPuqxFQrDcCPZgwEK
```

결과 예:

```
HTTP/1.1 204 No Content
```

예 8 토큰 ID로 토큰 삭제

```
DELETE /api/user/v2/users/test_user/preferences/tokens/21f981ad-6221-4fb4-a7d1-dd5560256dfb
HTTP/1.1
X-Auth-Session: pviHrthBGQhGZHoPuqxFQrDcCPZgwEK
```


RESTful API 역할 서비스

역할은 사용자에게 지정할 수 있는 권한 부여의 명명된 모음입니다. 용도에 따라 권한 부여가 다른 역할을 만들 수 있습니다. 어플라이언스 사용자에게 필요한 모든 역할이 지정될 수 있습니다. 역할 사용이 관리자 암호 공유보다 안전합니다. 역할은 사용자를 필요한 권한 부여로만 제한하며 속성 사용자 작업을 감사 로그의 해당 사용자 이름으로 제한합니다.

"기본 관리" 역할은 기존 역할이며 기본적으로 모든 사용자에게 지정됩니다.

역할 서비스 명령

다음 목록에서는 역할 서비스 명령을 보여줍니다.

표 81 역할 서비스 명령

요청	경로 <i>/role/v{1 2}</i> 에 추가	설명
GET	<i>/role/v{1 2}</i> 만 사용	역할 서비스 명령을 나열합니다.
GET	<i>/roles</i>	모든 역할을 나열합니다.
GET	<i>/roles/role</i>	지정된 역할의 등록 정보를 나열합니다.
POST	<i>/roles</i>	새 역할을 만듭니다.
PUT	<i>/roles/role</i>	지정된 역할의 등록 정보를 수정합니다.
PUT	<i>/roles/role/revoke</i>	모든 사용자에서 지정된 역할을 제거합니다.
DELETE	<i>/roles/role</i>	지정된 역할을 삭제합니다.
GET	<i>/roles/role/authorizations</i>	지정된 역할에 대한 모든 권한 부여를 나열합니다.
GET	<i>/roles/role/authorizations/auth</i>	지정된 역할 권한 부여의 등록 정보를 나열합니다.
POST	<i>/roles/role/authorizations</i>	지정된 역할에 대한 새 권한 부여를 만듭니다.
PUT	<i>/roles/role/authorizations/auth</i>	지정된 역할 권한 부여의 등록 정보를 수정합니다.
DELETE	<i>/roles/role/authorizations/auth</i>	지정된 역할 권한 부여를 삭제합니다.

역할 서비스 등록 정보

`name` 및 `description` 등록 정보는 역할을 만들 때 필요하며 모든 역할을 나열할 때 표시됩니다. 권한 부여는 역할이 만들어진 후 추가되며 특정 역할을 나열할 때 표시됩니다.

표 82 역할 등록 정보

등록 정보	유형	설명
<code>name</code>	string	목록에 표시되는 역할의 이름입니다.
<code>description</code>	string	역할에 대한 상세 설명입니다.
<code>authorizations</code>	list	이 역할에 대한 권한 부여입니다.

역할 나열

정의된 모든 역할을 나열합니다.

요청 예:

```
GET /api/role/v1/roles HTTP/1.1
Authorization: Basic Tm8gcGVla2luZyE=
Host: zfs-storage.example.com:215
Accept: application/json
```

결과 예:

```
{
  "roles": [
    {
      "description": "Basic administration",
      "href": "/api/role/v1/roles/basic",
      "name": "basic",
      "role": "basic"
    },
    {
      "description": "a",
      "href": "/api/role/v1/roles/rola",
      "name": "rola",
      "role": "rola"
    }
  ]
}
```

역할 가져오기

단일 역할에 대한 등록 정보를 검색합니다. 등록 정보 메타데이터를 반환하려면 `props` 질의 매개변수를 `true`로 설정합니다.

요청 예:

```
GET /api/role/v1/roles/basic?props=true HTTP/1.1
```

```
Authorization: Basic Tm8gcGVla2luZyE=
Host: zfs-storage.example.com:215
Accept: application/json
```

결과 예:

```
HTTP/1.1 200 OK
X-Zfssa-Appliance-API: 1.0
Content-Type: application/json
Content-Length: 390

{
  "props": [{
    "immutable": true,
    "label": "Role name",
    "name": "name",
    "type": "String"
  }, {
    "label": "A description of this role",
    "name": "description",
    "type": "String"
  }],
  "role": {
    "authorizations": [],
    "description": "Basic administration",
    "href": "/api/role/v1/roles/basic",
    "name": "basic"
  }
}
```

역할 만들기

이 명령은 새 역할을 만듭니다.

표 83 새 역할 만들기 등록 정보

등록 정보	유형	설명
name	string	새 역할의 이름(필수)
clone	string	원래 등록 정보를 복제할 역할의 이름(선택사항)
description	string	역할 설명(필수)

요청 예:

```
POST /api/role/v1/roles HTTP/1.1
Authorization: Basic Tm8gcGVla2luZyE=
Host: zfs-storage.example.com:215
Accept: application/json
Content-Type: application/json
Content-Length: 71

{"name": "role_workflow", "description": "Role to run workflows"}
```

결과 예:

```
HTTP/1.1 201 Created
X-Zfssa-Appliance-API: 1.0
Content-Type: application/json
Content-Length: 143
Location: /api/role/v1/roles/role_workflow

{
  "role": {
    "authorizations": [],
    "description": "Role to run workflows",
    "href": "/api/role/v1/roles/role_workflow",
    "name": "role_workflow"
  }
}
```

역할 수정

역할을 만든 후 역할 등록 정보를 수정할 수 있습니다.

요청 예:

```
PUT /api/role/v1/roles/role_workflow HTTP/1.1
Authorization: Basic Tm8gcGVla2luZyE=
Host: zfs-storage.example.com:215
Accept: application/json
Content-Type: application/json
Content-Length: 54

{"description":"Role allowing user to run workflows!"}
```

결과 예:

```
HTTP/1.1 202 Accepted
X-Zfssa-Appliance-API: 1.0
Content-Type: application/json
Content-Length: 158

{
  "role": {
    "authorizations": [],
    "description": "Role allowing user to run workflows!",
    "href": "/api/role/v1/roles/role_workflow",
    "name": "role_workflow"
  }
}
```

역할 취소

모든 사용자의 역할을 취소합니다.

요청 예:

```
PUT /api/role/v1/role_worksheets/revoke HTTP/1.1
```

```
Authorization: Basic Tm8gcGVla2luZyE=
Host: zfs-storage.example.com:215
Accept: application/json
```

결과 예:

```
HTTP/1.1 202 Accepted
X-Zfssa-Appliance-API: 1.0
Content-Type: application/json
Content-Length: 0
```

역할 삭제

시스템에서 역할을 삭제합니다. 역할이 아직 한 명 이상의 사용자에게 지정된 경우, `confirm=true`를 DELETE 명령에 추가합니다.

요청 예:

```
DELETE /api/role/v1/roles/rola?confirm=true HTTP/1.1
Authorization: Basic Tm8gcGVla2luZyE=
Host: zfs-storage.example.com:215
Accept: */*
```

결과 예:

```
HTTP/1.1 204 No Content
X-Zfssa-Appliance-API: 1.0
```

역할 권한 부여 나열

선택된 역할에 대한 권한 부여를 나열합니다.

요청 예:

```
GET /api/role/v1/roles/role_workflow/authorizations HTTP/1.1
Authorization: Basic Tm8gcGVla2luZyE=
Host: zfs-storage.example.com:215
Accept: application/json
```

결과 예:

```
{
  "authorizations": [{
    "allow_modify": false,
    "allow_read": true,
    "auth": "auth-000",
    "href": "/api/role/v1/roles/role_workflow/authorizations/auth-000",
    "owner": "*",
    "scope": "workflow",
    "uuid": "*"
  }]
}
```

역할 권한 부여 만들기

새 역할 권한 부여를 만듭니다. 입력 등록 정보는 CLI에 정의되어 있는 등록 정보와 동일합니다. 각 권한 부여에는 정의된 `scope` 등록 정보가 있습니다. 기타 등록 정보는 입력 범위를 기반으로 설정될 수 있습니다. 범위 값에는 다음이 포함됩니다.

ad	cluster	keystore	role	stmf	user
alert	dataset	nas	schema	svc	workflow
appliance	hardware	net	stat	update	worksheet

요청 예:

```
POST /api/role/v1/roles/role_workflow/authorizations HTTP/1.1
Authorization: Basic Tm8gcGVla2luZyE=
Host: zfs-storage.example.com:215
Accept: application/json
Content-Type: application/json
Content-Length: 41
{"scope": "workflow", "allow_read": true}
```

결과 예:

```
HTTP/1.1 201 Created
X-Zfssa-Appliance-API: 1.0
Content-Type: application/json
Content-Length: 171
Location: /api/role/v1/roles/role_workflow/authorizations/auth-000

{
  "auth": {
    "allow_modify": false,
    "allow_read": true,
    "href": "/api/role/v1/roles/role_workflow/authorizations/auth-000",
    "owner": "*",
    "scope": "workflow",
    "uuid": "*"
  }
}
```

역할 권한 부여 수정

역할 권한 부여 등록 정보를 수정할 수 있습니다.

요청 예:

```
PUT /api/role/v1/roles/role_workflow/authorizations/auth-000 HTTP/1.1
Authorization: Basic Tm8gcGVla2luZyE=
Host: zfs-storage.example.com:215
Accept: application/json
Content-Type: application/json
Content-Length: 29

{"allow_modify": true}
```

결과 예:

```
HTTP/1.1 202 Accepted
X-Zfssa-Appliance-API: 1.0
Content-Type: application/json
Content-Length: 171
```

```
{
  "auth": {
    "allow_modify": true,
    "allow_read": true,
    "href": "/api/role/v1/roles/role_workflow/authorizations/auth-000",
    "owner": "**",
    "scope": "workflow",
    "uuid": "**"
  }
}
```

역할 권한 부여 삭제

역할 권한 부여를 삭제합니다.

요청 예:

```
DELETE /api/role/v1/roles/role_workflow/authorizations/auth-000 HTTP/1.1
Authorization: Basic Tm8gcGV1a2luZyE=
Host: zfs-storage.example.com:215
Accept: */*
```

결과 예:

```
HTTP/1.1 204 No Content
X-Zfssa-Appliance-API: 1.0
```


워크플로우 및 스크립트 명령

이 서비스는 워크플로우를 관리하는 데 사용됩니다. 워크플로우는 어플라이언스에 업로드되고 어플라이언스에서 관리되는 스크립트입니다. 워크플로우는 브라우저 인터페이스 또는 명령 줄 인터페이스에서 첫번째 클래스 방식으로 매개변수화 및 실행할 수 있습니다. 워크플로우는 경보 조치로 실행되거나 지정된 시간에 실행될 수도 있습니다. 따라서 워크플로우를 통해 특정 정책 및 절차를 포착하는 방식으로 어플라이언스를 확장할 수 있으며, 특정 조직 또는 응용 프로그램의 최적의 사용법을 공식적으로 인코딩하는 등의 작업에 사용될 수 있습니다.

워크플로우 및 스크립트 서비스 명령

다음 표는 워크플로우 서비스 명령을 보여 줍니다.

표 84 워크플로우 서비스 명령

요청	경로 /api/workflow/v{1 2}에 추가	설명
GET	/api/workflow/v{1 2}만 사용	워크플로우 서비스 명령을 나열합니다
POST	/workflows	새 워크플로우를 어플라이언스로 업로드합니다.
GET	/workflows	모든 워크플로우를 나열합니다.
GET	/workflows/workflow	지정된 워크플로우 등록 정보를 나열합니다.
PUT	/workflows/workflow	지정된 워크플로우 등록 정보를 수정합니다.
PUT	/workflows/workflow/execute	지정된 워크플로우를 실행합니다.
DELETE	/workflows/workflow	지정된 워크플로우를 삭제합니다.
POST	/scripts	스크립트를 업로드 및 실행합니다.
GET	/scripts	실행 중인 모든 스크립트를 나열합니다.
GET	/scripts/script	실행 중인 스크립트에 재연결합니다.
DELETE	/scripts/script	실행 중인 스크립트를 중지합니다.

워크플로우 업로드

워크플로우를 어플라이언스에 업로드합니다.

요청 예:

```
POST /api/workflow/v1/workflows HTTP/1.1
Authorization: Basic Tm8gcGVla2luZyE=
Host: zfs-storage.example.com:215
Accept: application/json
Content-Type: application/javascript
Content-Length: 290

var workflow = {
  name: 'Echo',
  description: 'Echo bird repeats a song.',
  parameters: {
    song: {
      label: 'Words of a song to sing',
      type: 'String',
    }
  },
  execute: function (params) { return (params.song) }
};
```

결과 예:

```
HTTP/1.1 201 Created
X-Zfssa-Appliance-API: 1.0
Content-Type: application/json
Content-Length: 268
X-Zfssa-Version: user/generic@2013.09.14,1-0
Location: /api/workflow/v1/workflows/f4fe892f-cf46-4d6a-9026-cd0c0cce9971

{
  "workflow": {
    "href": "/api/workflow/v1/workflows/f4fe892f-cf46-4d6a-9026-cd0c0cce9971",
    "name": "Echo",
    "description": "Echo bird repeats a song.",
    "uuid": "f4fe892f-cf46-4d6a-9026-cd0c0cce9971",
    "owner": "root",
    "origin": "<local>",
    "setid": false,
    "alert": false,
    "version": "",
    "scheduled": false
  }
}
```

워크플로우 나열

어플라이언스에 설치된 모든 워크플로우를 나열합니다. 질의 매개변수 `showhidden=true`가 설정된 경우 목록에는 일반적으로 숨겨진 워크플로우가 포함됩니다.

요청 예:

```
GET /api/workflow/v1/workflows HTTP/1.1
Authorization: Basic Tm8gcGVla2luZyE=
Host: zfs-storage.example.com:215
Accept: application/json
```

결과 예:

```

HTTP/1.1 200 OK
X-Zfssa-Appliance-API: 1.0
Content-Type: application/json; charset=utf-8
Content-Length: 1908

{
  "workflows": [
    {
      "description": "Clear locks held on behalf of an NFS client",
      "href": "/api/workflow/v1/workflows/10f25f2c-3a56-e733-d9c7-d4c6fd84e073",
      ...
    },
    {
      "description": "Sets up environment for Oracle Solaris Cluster NFS",
      "href": "/api/workflow/v1/workflows/2793f2dc-72de-eac4-c58b-cfbe527df92d",
      ...
    },
    {
      "description": "Removes the artifacts from the appliance used by Oracle Solaris Cluster NFS",
      "href": "/api/workflow/v1/workflows/9e2d5eed-cc72-67b0-e913-bf5ffad1d9e1",
      ...
    },
    {
      "description": "Sets up environment to be monitored by Oracle Enterprise Manager",
      "href": "/api/workflow/v1/workflows/bb5de1b8-b950-6da6-a650-f6fb19f1172c",
      ...
    },
    {
      "description": "Removes the artifacts from the appliance used by Oracle Enterprise Manager",
      "href": "/api/workflow/v1/workflows/bd7214fc-6bba-c7ad-ed1f-942c0189e757",
      ...
    }
  ]
}

```

워크플로우 가져오기

단일 워크플로우에 대한 등록 정보를 가져옵니다. 헤더에서 Accept가 application/javascript로 지정된 경우 워크플로우의 콘텐츠가 반환되고, 그렇지 않으면 워크플로우 등록 정보가 반환됩니다.

Accept가 application/javascript로 지정된 요청 예:

```

GET /api/workflow/v1/workflows/cc574599-4763-4523-9e72-b74e1246d448 HTTP/1.1
Authorization: Basic Tm8gcGVla2luZyE=
Host: zfs-storage.example.com:215
Accept: application/javascript

```

결과 예:

```

HTTP/1.1 200 OK
X-Zfssa-Appliance-API: 1.0
Content-Type: application/javascript; charset=utf-8
Content-Length: 916

var workflow = {
  name: 'Clear locks',

```

```

description: 'Clear locks held on behalf of an NFS client',
origin: 'Oracle Corporation',
version: '1.0.0',
parameters: {
  hostname: {
    label: 'Client hostname',
    type: 'String'
  },
  ipaddrs: {
    label: 'Client IP address',
    type: 'String'
  }
},
validate: function (params) {
  if (params.hostname == '') {
    return ({ hostname: 'Hostname cannot be empty.' });
  }

  if (params.ipaddrs == '') {
    return ({ ipaddrs: 'IP address cannot be empty.' });
  }
},
execute: function (params) {
  try {
    nas.clearLocks(params.hostname, params.ipaddrs);
  } catch (err) {
    return ('Failed to clear NFS locks: ' + err.message);
  }

  return ('Clear of locks held for ' + params.hostname +
    ' returned success. ');
}
};

```

Accept가 application/json으로 지정된 요청 예:

```

GET /api/workflow/v1/workflows/cc574599-4763-4523-9e72-b74e1246d448 HTTP/1.1
Authorization: Basic Tm8gcGVla2luZyE=
Host: zfs-storage.example.com:215
Accept: application/json

```

결과 예:

```

HTTP/1.1 200 OK
X-Zfssa-Appliance-API: 1.0
Content-Type: application/json; charset=utf-8
Content-Length: 649

```

```

{
  "workflow": {
    "href": "/api/workflow/v1/workflows/cc574599-4763-4523-9e72-b74e1246d448",
    "name": "Clear locks",
    "description": "Clear locks held on behalf of an NFS client",
    "uuid": "cc574599-4763-4523-9e72-b74e1246d448",
    "checksum": "695d029224f614258e626fe0b3c449c1233dee119571f23b678f245f7748d13c",
    "installdate": "Wed Apr 01 2015 17:59:44 GMT+0000 (UTC)",
    "owner": "root",
    "origin": "Oracle Corporation",
    "setid": false,
    "alert": false,
    "version": "1.0.0",
    "scheduled": false
  }
}

```

워크플로우 수정

워크플로우 리소스에 PUT 요청을 전송하여 단일 워크플로우에 대한 등록 정보를 수정할 수 있습니다.

요청 예:

```
PUT /api/workflow/v1/workflows/6c2b6545-fa78-cc7b-8cc1-ff88bd628e7d HTTP/1.1
Authorization: Basic Tm8gcGV1a2luZyE=
Host: zfs-storage.example.com:215
Accept: application/json
Content-Type: application/json
Content-Length: 28
```

```
{"setid": false}
```

결과 예:

```
HTTP/1.1 202 Accepted
X-Zfssa-Appliance-API: 1.0
Content-Type: application/json
Content-Length: 234
```

```
{
  "workflow": {
    "alert": false,
    "description": "Echo bird repeats a song.",
    "href": "/api/workflow/v1/workflows/448b78e1-f219-e8f4-abb5-e01e09e1fac8",
    "name": "Echo",
    "origin": "<local>",
    "owner": "root",
    "scheduled": false,
    "setid": true,
    "uuid": "448b78e1-f219-e8f4-abb5-e01e09e1fac8",
    "version": ""
  }
}
```

워크플로우 실행

워크플로우 스크립트를 실행하고 결과를 반환합니다. 워크플로우 매개변수는 본문 내 JSON 객체에 전달됩니다. 성공 시 HTTP 상태 202(Accepted)가 워크플로우 출력을 포함하는 단일 결과 등록 정보가 있는 JSON 객체와 함께 반환됩니다.

요청 예:

```
PUT /api/workflow/v1/workflows/6c2b6545-fa78-cc7b-8cc1-ff88bd628e7d/run HTTP/1.1
Authorization: Basic Tm8gcGV1a2luZyE=
Host: zfs-storage.example.com:215
Accept: application/json
Content-Type: application/json
Content-Length: 28
```

```
{"song": "tweet tweet tweet"}
```

결과 예:

```
HTTP/1.1 202 Accepted
X-Zfssa-Appliance-API: 1.0
Content-Type: application/json
Content-Length: 34

{
  "result": "tweet tweet tweet\n"
}
```

워크플로우 삭제

어플라이언스에서 워크플로우 스크립트를 삭제합니다.

요청 예:

```
DELETE /api/workflow/v1/workflows/f4fe892f-cf46-4d6a-9026-cd0c0cce9971 HTTP/1.1
Authorization: Basic Tm8gcGVla2luZyE=
Host: zfs-storage.example.com:215
Accept: */*
```

결과 예:

```
HTTP/1.1 204 No Content
X-Zfssa-Appliance-API: 1.0
```

스크립트 업로드 및 실행

어플라이언스에서 스크립트를 업로드 및 실행합니다.

루트 사용자는 어플라이언스로 업로드된 모든 스크립트를 확인하고 이에 액세스할 수 있습니다. 비루트 사용자는 자신의 스크립트만 확인하고 이에 액세스할 수 있습니다.

스크립트 작성에 대한 자세한 내용은 [“Working with CLI Scripting” in Oracle ZFS Storage Appliance Administration Guide, Release OS8.8.x](#)을 참조하십시오.

이 스크립트는 어플라이언스의 모든 공유를 나열합니다.

요청 예:

```
$ curl -kv --user root:pw --data-binary @listShares.aksh \
https://hostname:215/api/workflow/v1/scripts

POST /api/workflow/v1/scripts HTTP/1.1
Host: zfs-storage.example.com:215
Authorization: Basic Tm8gcGVla2luZyE=
User-Agent: curl/7.45.0
Accept: */*
Content-Length: 12
```

```
Content-Type: application/x-www-form-urlencoded
```

결과 예:

```
HTTP/1.1 201 Created
Date: Mon, 27 Mar 2017 22:16:38 GMT
X-Zfssa-Workflow-API: 1.1
X-Zfssa-Version: user/generic@2017.02.27,1-0
X-Zfssa-API-Version: 1.0
Content-Type: plain/text; charset=utf-8
Transfer-Encoding: chunked
```

```
default
share1
share2
fs1
lun1
```

실행 중인 모든 스크립트 나열

다음 명령을 사용하여 실행 중인 모든 스크립트를 나열할 수 있습니다.

루트 사용자는 어플라이언스로 업로드된 모든 스크립트를 확인하고 이에 액세스할 수 있습니다. 비루트 사용자는 자신의 스크립트만 확인하고 이에 액세스할 수 있습니다.

스크립트 작성에 대한 자세한 내용은 [“Working with CLI Scripting” in Oracle ZFS Storage Appliance Administration Guide, Release OS8.8.x](#)을 참조하십시오.

요청 예:

```
$ curl -kv --user root:pw https://hostname:215/api/workflow/v1/scripts
```

```
GET /api/workflow/v1/scripts HTTP/1.1
Host: zfs-storage.example.com:215
Authorization: Basic Tm8gcGVla2luZyE=
User-Agent: curl/7.45.0
Accept: */*
```

결과 예:

```
HTTP/1.1 200 OK
Date: Mon, 27 Mar 2017 22:41:06 GMT
Content-Length: 96
X-Zfssa-Workflow-API: 1.1
X-Zfssa-API-Version: 1.0
Content-Type: application/json; charset=utf-8
```

```
{
  "scripts": [
    {
      "time": 4,
      "href": "/api/workflow/v1/scripts/1",
      "user": "root",
      "script": "1"
    },
    {
```

```
        "time": 39,
        "href": "/api/workflow/v1/scripts/9",
        "user": "root",
        "script": "9"
      }
    ]
  }
}
```

실행 중인 스크립트에 재연결

루트 사용자는 어플라이언스로 업로드된 실행 중인 모든 스크립트에 재연결할 수 있습니다. 비 루트 사용자는 자신의 실행 중인 스크립트에만 재연결할 수 있습니다.

스크립트 작성에 대한 자세한 내용은 [“Working with CLI Scripting” in Oracle ZFS Storage Appliance Administration Guide, Release OS8.8.x](#)을 참조하십시오.

요청 예:

```
$ curl -kv -H "Accept: text/plain" --user root:pw \
https://hostname:215/api/workflow/v1/scripts/9
```

```
GET /api/workflow/v1/scripts/9 HTTP/1.1
Host: zfs-storage.example.com:215
Authorization: Basic Tm8gcGVla2luZyE=
User-Agent: curl/7.45.0
Accept: text/plain
```

결과 예:

```
{
  "test2": {,
    "str": "this is a string",
    "bool": "True",
    "posint": 994,
    "int": 1123,
    "address": "",
    "host": "192.0.2.0",
    "hostname": "example.com",
    "color": "red",
    "languages": "latin",
    "size": "red",
    "onoff": "off",
    "number": 0,
    "stringlist": "this is another string",
    "emptystringlist": "this is another string",
    "yetanotherstr": "You can't change me",
    "emptystr": "Any string",
    "password": "password",
    "longpassword": "longpassword",
    "permissions": "022",
    "nonnegativeint": 42,
    "port": 21,
    "time": "Thu Jan 01 1970 15:22:30 GMT+0000 (UTC)",
    "date": "Sun Jun 17 2007 00:00:00 GMT+0000 (UTC)",
    "datetime": "Sun Jun 17 2007 15:22:00 GMT+0000 (UTC)",
    "hostport": "ipaddr-1",
    "dn": "uid=root,ou=people,dc=fishpong,dc=com",
```

```

    "commalist": "foo,bar"
  }
},
"utask": []
}

```

실행 중인 스크립트 중지

루트 사용자는 어플라이언스로 업로드된 실행 중인 모든 스크립트를 삭제할 수 있습니다. 비루트 사용자는 자신의 실행 중인 스크립트에만 액세스하고 이를 삭제할 수 있습니다.

스크립트 작성에 대한 자세한 내용은 [“Working with CLI Scripting” in Oracle ZFS Storage Appliance Administration Guide, Release OS8.8.x](#)을 참조하십시오.

요청 예:

```

$ curl -kv -X DELETE --user root:11a \
  https://hostname:215/api/workflow/v1/scripts/9

DELETE /api/workflow/v1/scripts/9 HTTP/1.1
Host: zfs-storage.example.com:215
Authorization: Basic Tm8gcGV1a2luZyE=
User-Agent: curl/7.45.0
Accept: */*

```

결과 예:

```

HTTP/1.1 204 No Content
Date: Mon, 27 Mar 2017 22:59:12 GMT
Content-Length: 0
X-Zfssa-Workflow-API: 1.1
X-Zfssa-Version: build/generic@2017.02.27,1-0
X-Zfssa-API-Version: 1.0
Content-Type: application/json; charset=utf-8

```


RESTful 클라이언트

모든 HTTP 클라이언트는 RESTful 클라이언트로 사용될 수 있습니다. BUI도 리소스 URL을 입력하여 RESTful API GET 결과를 반환할 수 있습니다. Mozilla Firefox에는 RESTful 요청 (<https://addons.mozilla.org/en-us/firefox/addon/restclient/>)을 수행하기 위해 설치될 수 있는 RESTful 클라이언트 모듈이 있습니다. 이 모듈은 PUT, POST 및 DELETE 요청뿐만 아니라 일반 HTTP GET 요청도 허용합니다.

SSLv2/3 프로토콜은 더 이상 지원되지 않으므로 RESTful 클라이언트는 TLS 프로토콜을 사용해야 합니다. Curl 클라이언트는 curl 버전 7.34.0 이상을 사용해야 합니다.

이 절에서는 여러 RESTful 클라이언트에 대해 자세히 다룹니다..

Curl REST 클라이언트

Curl 클라이언트는 curl 버전 7.34.0 이상을 사용해야 합니다. 두 가지 일반적인 CLI 기반 HTTP 클라이언트는 wget 및 curl입니다. 이 절에는 curl을 사용하여 RESTful API 호출을 수행하는 방법에 대한 여러 가지 예가 나와 있습니다. wget을 사용하면 유사한 기능을 수행할 수 있습니다.

리소스 데이터 가져오기

이 예에서는 간단한 HTTP GET 요청을 사용하여 일부 JSON 데이터를 얻는 방법을 보여 줍니다.

```
$ curl --user ${USER}:${PASSWORD} -k \
-i https://hostname:215/api/storage/v1/pools/p1
```

```
HTTP/1.1 200 OK
Date: Tue, 23 Jul 2018 12:57:02 GMT
Server: WSGIServer/0.1 Python/2.6.4
Content-Length: 284
Content-Type: application/json
X-Zfs-Sa-Nas-API: 1.0
```

```
{
  "pool": {
    "profile": "mirror",
    "name": "p1",
```

```

    "usage": {
      "available": 895468984832.0,
      "total": 895500681216.0,
      "dedupratio": 100,
      "used": 31696384.0
    },
    "peer": "00000000-0000-0000-0000-000000000000",
    "state": "online",
    "owner": "admin1",
    "asn": "314d252e-c42b-e844-dab1-a3bca680b563"
  }
}

```

새 리소스 만들기

이 예에서는 요청의 JSON 데이터를 전송하여 새 리소스를 만드는 방법을 보여 줍니다.

```

$ curl --user ${USER}:${PASSWORD} -s -k -i -X POST -d @- \
  -H "Content-Type: application/json" \
  https://zfs-storage.example.com:215/api/user/v1/users <<JSON
> {"logname": "rest_user",
> "fullname": "REST User",
> "initial_password": "password"}
> JSON

```

```

HTTP/1.1 201 Created
Date: Tue, 23 Jul 2018 13:07:37 GMT
Server: WSGIServer/0.1 Python/2.6.4
X-Zfs-Sa-Appliance-API: 1.0
Content-Type: application/json
Content-Length: 357

```

```

{
  "user": {
    "logname": "rest_user",
    "fullname": "REST User",
    "initial_password": "password",
    "require_annotation": false,
    "kiosk_mode": false,
    "kiosk_screen": "status/dashboard",
    "roles": ["basic"],
    "exceptions": {},
    "preferences": {
      "href": "/api/user/v1/users/admin1/preferences",
      "locale": "C",
      "login_screen": "status/dashboard",
      "session_timeout": 15,
      "cli_idle_timeout": -1,
      "advanced_analytics": false,
      "keys": {}
    }
  }
}

```

기존 리소스 수정

이 예에서는 사용자의 세션 시간 초과를 수정합니다.

```
$ curl --user admin1:password -3 -s -k -i -X PUT \
  -H "Content-Type: application/json" -d @- \
  https://zfs-storage.example.com:215/api/appliance/v1/users/admin1/preferences <<JSON
> {"session_timeout":60}
> JSON

HTTP/1.1 202 Accepted
Date: Wed, 24 Jul 2018 05:43:17 GMT
X-Zfs-Sa-Appliance-API: 1.0
Content-Type: application/json
Content-Length: 0

{
  "preferences": {
    "href": "appliance/v1/users/admin1/preferences",
    "locale": "C",
    "login_screen": "status/dashboard",
    "session_timeout": 60,
    "cli_idle_timeout": -1,
    "advanced_analytics": false,
    "keys": {}
  }
}
```

기존 리소스 삭제

이 명령은 시스템에서 사용자를 제거합니다.

```
$ curl --user ${USER}:${PASSWORD} -s -k -i -X DELETE \
  https://zfs-storage.example.com:215/api/appliance/v1/users/admin1
HTTP/1.1 204 No Content
Date: Tue, 23 Jul 2018 13:21:11 GMT
Server: WSGIServer/0.1 Python/2.6.4
X-Zfs-Sa-Appliance-API: 1.0
Content-Length: 0
```

Python RESTful 클라이언트

Python RESTful API 클라이언트는 RESTful 서비스의 테스트 개발을 지원하는 REST 테스트 라이브러리와 함께 제공됩니다.

RESTful 클라이언트 프로그램 예:

```
>>> import urllib2
>>> import json

>>> request = urllib2.Request("https://zfsssa.example:215/api/access/v1", "")
>>> request.add_header("X-Auth-User", "rest_user")
>>> request.add_header("X-Auth-Key", "password")
>>> response = urllib2.urlopen(request)
>>> response.getcode()
201

>>> info = response.info()
>>>
```

```
opener = urllib2.build_opener(urllib2.HTTPHandler)
>>> opener.addheaders = [("X-Auth-Session", info.getheader("X-Auth-Session")),
... ('Content-Type', 'application/json'), ('Accept', 'application/json')]
```

그런 다음 opener를 사용하여 이미 사전 인증되었으며 JSON 데이터를 보내고 받을 준비가 완료된 요청을 열 수 있습니다.

리소스 가져오기

다음 Python 코드를 사용하여 RESTful API 리소스에서 데이터를 가져올 수 있습니다.

GET 예:

```
>>> request = urllib2.Request("https://zfs-storage.example.com:215/api/network/v1/routes")
>>> response = opener.open(request)
>>> response.getcode()
200
>>> body = json.loads(response.read())
>>> print json.dumps(body, sort_keys=True, indent=4)
{
  "routes": [
    {
      "destination": "ipaddr-0",
      "family": "IPv4",
      "gateway": "ipaddr-1",
      "href":
"/api/network/v1/routes/ixgbe0,ipaddr-0,ipaddr-1",
      "interface": "ixgbe0",
      "mask": 0,
      "type": "static"
    }
  ]
}
```

리소스 만들기

새 리소스를 만드는 Python 코드 예:

```
>>> action = {'category': 'network'}
>>> post_data = json.dumps(action)
>>> request = urllib2.Request("https://zfs-storage.example.com:215/api/alert/v1/actions",
post_data)
>>> request.add_header('Content-Type', 'application/json')

>>> response = opener.open(request)
>>> response.getcode()
201
>>> response.info().getheader('Location')
'/api/alert/v1/actions/actions-001'
>>> body = json.loads(response.read())
>>> print json.dumps(body, sort_keys=True, indent=4)
```

```

{
    "actions": {
        "category": "network",
        "datalink_failed": true,

        "datalink_ok": true,
        "href":
        "/api/alert/v1/actions/actions-001",

        "ip_address_conflict": true,

        "ip_address_conflict_resolved": true,

        "ip_interface_degraded": true,
        "ip_interface_failed":
        true,
        "ip_interface_ok": true,

        "network_port_down": true,
        "network_port_up":
        true
    }
}

```

리소스 수정

기존 리소스를 수정하는 Python 코드 예:

```

>>> put_data = '{"ip_address_conflict_resolved": false}'
>>>
>>> request = urllib2.Request("https://zfs-storage.example.com:215/api/alert/v1/
actions/actions-001", put_data)
>>> request.add_header('Content-Type', 'application/json')
>>> request.get_method = lambda: 'PUT'

>>> response = opener.open(request)
>>> response.getcode()
202
>>> body = json.loads(response.read())
>>> print json.dumps(body, sort_keys=True, indent=4)
{
    "actions": {
        "category": "network",
        "datalink_failed": true,

        "datalink_ok": true,
        "href":
        "/api/alert/v1/actions/actions-001",

        "ip_address_conflict": true,

        "ip_address_conflict_resolved": false,

        "ip_interface_degraded": true,
        "ip_interface_failed":
        true,
        "ip_interface_ok": true,

        "network_port_down": true,

```

```
        "network_port_up":  
        true  
    }  
}
```

기존 리소스 삭제

기존 리소스를 삭제하는 Python 코드 예:

```
>>> request = urllib2.Request("https://zfs-storage.example.com:215/api/alert/v1/actions/  
actions-001")  
>>> request.get_method = lambda: 'DELETE'  
>>> response = opener.open(request)  
>>> response.getcode()  
204
```