

**Oracle® ZFS Storage Appliance RESTful
API 설명서, 릴리스 OS8.8.0**



부품 번호: E97786-01
2018년 11월

부품 번호: E97786-01

Copyright © 2014, 2018, Oracle and/or its affiliates. All rights reserved.

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, 재생, 번역, 방송, 수정, 라이선스, 전송, 배포, 진열, 실행, 발행, 또는 전시될 수 없습니다. 본 소프트웨어를 리버스 엔지니어링, 디스어셈블리 또는 디컴파일하는 것은 상호 운용에 대한 법규에 의해 명시된 경우를 제외하고는 금지되어 있습니다.

이 안의 내용은 사전 공지 없이 변경될 수 있으며 오류가 존재하지 않음을 보증하지 않습니다. 만일 오류를 발견하면 서면으로 통지해 주기 바랍니다.

만일 본 소프트웨어나 관련 문서를 미국 정부나 또는 미국 정부를 대신하여 라이선스한 개인이나 법인에게 배송하는 경우, 다음 공지사항이 적용됩니다.

U.S. GOVERNMENT END USERS: Oracle programs, including any operating system, integrated software, any programs installed on the hardware, and/or documentation, delivered to U.S. Government end users are "commercial computer software" pursuant to the applicable Federal Acquisition Regulation and agency-specific supplemental regulations. As such, use, duplication, disclosure, modification, and adaptation of the programs, including any operating system, integrated software, any programs installed on the hardware, and/or documentation, shall be subject to license terms and license restrictions applicable to the programs. 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 포함한 본질적으로 위험한 애플리케이션에서 사용할 목적으로 개발되거나 그 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 애플리케이션에서의 사용으로 인해 발생하는 어떠한 손해에 대해서도 책임지지 않습니다.

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

Intel 및 Intel Xeon은 Intel Corporation의 상표 내지는 등록 상표입니다. SPARC 상표 일체는 라이선스에 의거하여 사용되며 SPARC International, Inc.의 상표 내지는 등록 상표입니다. AMD, Opteron, AMD 로고, 및 AMD Opteron 로고는 Advanced Micro Devices의 상표 내지는 등록 상표입니다. UNIX는 The Open Group의 등록상표입니다.

본 소프트웨어 혹은 하드웨어와 관련문서(설명서)는 제3자로부터 제공되는 콘텐츠, 제품 및 서비스에 접속할 수 있거나 정보를 제공합니다. 사용자와 오라클 간의 합의서에 별도로 규정되어 있지 않는 한 Oracle Corporation과 그 자회사는 제3자의 콘텐츠, 제품 및 서비스와 관련하여 어떠한 책임도 지지 않으며 명시적으로 모든 보증에 대해서도 책임을 지지 않습니다. Oracle Corporation과 그 자회사는 제3자의 콘텐츠, 제품 및 서비스에 접속하거나 사용으로 인해 초래되는 어떠한 손실, 비용 또는 손해에 대해 어떠한 책임도 지지 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 Storage Appliance RESTful API 시작하기	15
RESTful API 인증	15
RESTful API 버전	16
서비스 버전	16
일반 RESTful 작업	17
HTTP 응답 본문	17
HTTP 응답 헤더	18
질의 매개변수	18
질의 매개변수: props	19
질의 매개변수: limit	19
질의 매개변수: start	19
질의 매개변수: depth	19
질의 매개변수: match	21
어플라이언스 오류	22
액세스 설정	23
보안 프로토콜 및 암호화 설정	23
RESTful API 작업	25
서비스 액세스	25
서비스 나열	25
서비스 가져오기 명령	26
인증 세션	27
로그인 세션	27
세션 로그아웃	28
RESTful API 경보 서비스	29
경보 서비스 명령	29
경보 임계값	30
경보 임계값 나열	31

경보 임계값 가져오기	31
경보 임계값 만들기	32
경보 임계값 수정	33
경보 임계값 삭제	33
경보 조치	33
경보 조치 나열	41
경보 조치 가져오기	42
경보 조치 만들기	42
경보 조치 수정	43
경보 조치 삭제	44
경보 조치 항목	44
경보 항목 만들기	44
경보 조치 수정	45
경보 조치 항목 삭제	45
Analytics Services	47
Analytics 명령	47
Analytics 설정	48
설정 가져오기	48
설정 수정	49
Analytics 워크시트	49
워크시트 나열	50
Analytics 워크시트 가져오기	50
워크시트 만들기	51
워크시트 이름 바꾸기	51
워크시트 삭제	52
워크시트 데이터 세트 나열	52
워크시트 데이터 세트 추가	53
워크시트 데이터 세트 수정	53
Analytics 데이터 세트	54
데이터 세트 나열	55
데이터 세트 가져오기	56
데이터 세트 만들기	57
데이터 세트 수정	57
데이터 세트 삭제	58
데이터 세트 저장	58
데이터 세트 데이터 제거	58
데이터 세트 데이터 가져오기	59

하드웨어 서비스	63
클러스터	63
클러스터 등록 정보 가져오기	63
클러스터 리소스 가져오기	64
클러스터 리소스 수정	64
클러스터 명령	64
클러스터 링크	65
클러스터 설정	65
새시	66
새시 나열	66
새시 구성요소 가져오기	68
하드웨어 구성요소 가져오기	69
구성요소 등록 정보 수정	70
 로그 명령	 73
로그 관리 명령	73
로그 나열	73
로그 항목 가져오기	74
로그 다운로드	76
로그 다운로드	76
 네트워크 명령	 77
네트워크 구성	77
네트워크 데이터 링크	78
네트워크 데이터 링크 나열	79
네트워크 데이터 링크 가져오기	80
네트워크 데이터 링크 만들기	80
네트워크 데이터 링크 수정	81
네트워크 데이터 링크 삭제	82
네트워크 장치	82
네트워크 장치 나열	82
네트워크 장치 가져오기	83
네트워크 인터페이스	84
네트워크 인터페이스 나열	84
네트워크 인터페이스 가져오기	85
네트워크 인터페이스 만들기	86
네트워크 인터페이스 수정	86
네트워크 인터페이스 삭제	87

네트워크 경로	87
경로 나열	88
경로 가져오기	88
경로 추가	89
경로 삭제	89
RESTful API 문제 서비스	91
문제 서비스 명령	91
문제 나열	91
문제 나열	92
문제 복구	92
RESTful API 역할 서비스	95
역할 서비스 명령 개요	95
역할 나열	96
역할 가져오기	96
역할 만들기	97
역할 수정	98
역할 취소	98
역할 삭제	99
역할 권한 부여 나열	99
역할 권한 부여 만들기	99
역할 권한 부여 수정	100
역할 권한 부여 삭제	101
RESTful API SAN 서비스	103
SAN 개요	103
SAN 개시자	103
개시자 나열	104
개시자 세부정보 가져오기	105
개시자 만들기	105
개시자 수정	106
개시자 삭제	107
개시자 그룹	107
개시자 그룹 나열	108
개시자 그룹 세부정보 가져오기	108
개시자 그룹 만들기	109
개시자 그룹 삭제	109

대상	110
대상 나열	111
대상 세부정보 가져오기	112
대상 만들기	112
대상 수정	113
대상 삭제	113
대상 그룹	114
대상 그룹 나열	114
대상 그룹 가져오기	115
대상 그룹 만들기	115
대상 그룹 삭제	116
 서비스 명령	117
서비스 명령	117
서비스 나열	117
서비스 가져오기	120
서비스 상태 변경	121
서비스 구성 수정	122
서비스 리소스	122
 RESTful API 스토리지 서비스	125
스토리지 풀 작업	125
풀 나열	125
풀 가져오기	126
풀 구성	127
풀에 스토리지 추가	128
풀에서 스토리지 제거	129
풀 스크러빙	130
풀 구성 해제	130
프로젝트 작업	130
프로젝트 나열	132
프로젝트 등록 정보 가져오기	133
프로젝트 만들기	134
프로젝트 수정	135
프로젝트 삭제	136
프로젝트 사용량	136
파일 시스템 작업	136
파일 시스템 나열	138

파일 시스템 가져오기	139
파일 시스템 만들기	140
파일 시스템 수정	141
파일 시스템 삭제	142
파일 시스템 쿼터 및 사용량	143
LUN 작업	143
LUN 나열	144
LUN 가져오기	145
새 LUN 만들기	146
LUN 수정	147
Lun 삭제	148
스냅샷 및 복제 작업	149
스냅샷 나열	152
스냅샷 가져오기	152
스냅샷 만들기	153
스냅샷 이름 바꾸기	154
스냅샷 복제	154
스냅샷 롤백	155
스냅샷 삭제	156
스냅샷 종속 항목 나열	157
스키마	158
등록 정보 나열	159
등록 정보 가져오기	159
등록 정보 만들기	160
등록 정보 수정	160
등록 정보 삭제	161
복제	161
복제 서비스 가져오기	162
복제 서비스 상태 수정	162
복제 대상	163
복제 대상 나열	163
복제 대상 가져오기	164
복제 대상 만들기	164
복제 대상 삭제	165
복제 작업	165
일반 작업 인터페이스 사용	165
프로젝트, 파일 시스템 또는 LUN 컨텍스트에서 복제 작업	167
복제 작업 나열	169

복제 작업 가져오기	170
복제 작업 만들기	171
복제 작업 수정	172
복제 작업 진행률 모니터	174
업데이트 취소	176
업데이트 전송	176
복제 작업 삭제	176
복제 패키지	177
복제 소스 나열	180
복제 패키지 나열	181
패키지 수정	181
패키지 삭제	182
업데이트 취소	182
패키지 복제	183
패키지 단절	184
패키지 방향 바꾸기	184
암호화	185
모든 로컬 키 나열	186
로컬 키 나열	186
모든 OKM 키 나열	186
 시스템 명령	 187
어플라이언스 시스템 명령	187
버전 가져오기	187
시스템 전원 끄기	188
시스템 재부트	188
시스템 관리 다시 시작	189
진단 재부트	189
공장 초기화 재설정	189
시스템 지원 번들	190
지원 번들 만들기	190
지원 번들 나열	191
지원 번들 가져오기	192
지원 번들 취소	192
지원 번들 업로드 재시도	193
지원 번들 업로드	193
지원 번들 삭제	194
시스템 업데이트	194

시스템 업데이트 나열	195
시스템 업데이트 가져오기	196
플랫폼 펌웨어 업데이트 상태 가져오기	196
구성요소 펌웨어 업데이트 상태 가져오기	197
시스템 업데이트 업로드	197
업그레이드	198
롤백	198
업데이트 이미지 삭제	199
RESTful API 사용자 서비스	201
사용자 서비스 명령	201
사용자 나열	202
사용자 가져오기	203
사용자 만들기	204
사용자 수정	207
사용자 삭제	208
워크플로우 및 스크립트 명령	209
워크플로우 및 스크립트 서비스 명령	209
워크플로우 업로드	209
워크플로우 나열	210
워크플로우 가져오기	211
워크플로우 수정	213
워크플로우 실행	213
워크플로우 삭제	214
스크립트 업로드 및 실행	214
실행 중인 모든 스크립트 나열	215
실행 중인 스크립트에 재연결	216
실행 중인 스크립트 중지	217
RESTful 클라이언트	219
Curl Rest 클라이언트	219
리소스 데이터 가져오기	219
새 리소스 만들기	220
기존 리소스 수정	220
기존 리소스 삭제	221
Python RESTful 클라이언트	221
리소스 가져오기	222

리소스 만들기	222
리소스 수정	223
기존 리소스 삭제	224

Oracle ZFS Storage Appliance RESTful API 시작하기

Oracle ZFS Storage Appliance는 네트워크를 통해 효율적인 파일 및 블록 데이터 서비스를 제공합니다. 이 설명서에서는 어플라이언스를 관리하는 데 사용할 수 있는 Oracle ZFS Storage Appliance 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 해 줍니다.

RESTful API 인증

Oracle ZFS Storage Appliance 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 로그인 정보가 포함되어 있어야 합니다.

HTTP 헤더 예:

```
Authorization: Basic abcefgMWE
```

- **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
```

- **세션 인증** – 세션이 인증된 경우 세션이 만료될 때까지 계속 명령을 실행하는 데 세션 헤더를 사용할 수 있습니다. 세션이 만료된 후 명령이 허용되기 전 다시 인증을 수행해야 합니다.

세션 헤더 예:

X-Auth-Session: guigqpQRE4g89ngb

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

서비스 버전

서비스에 액세스하기 위한 URI(Uniform Resource Identifier)의 일부분으로 버전 번호가 각 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

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 4627에 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 반환된 데이터를 수정하거나 개선할 질의 매개변수(선택사항)를 사용합니다. 자세한 내용은 각 리소스에 대한 설명서를 참조하십시오. 모든 리소스에서 모든 질의 매개변수를 지원하는 것은 아닙니다. 이 절에서는 리소스가 지정된 질의 매개변수를 구현할 때 사용될 일반 질의 매개변수에 대해서만 설명합니다.

표 2 일반 질의 매개변수

매개변수	설명
props=true	리소스에 대한 등록 정보 메타데이터를 나열합니다. 기본값은 false입니다.
limit=n	반환된 목록 요소 수를 제한합니다.
start=n	반환된 요소 데이터를 시작하는 데 사용된 인덱스 번호 또는 시간입니다.

매개변수	설명
depth= <i>n</i>	반환된 데이터의 세부정보 레벨을 지정하는 인덱스 번호입니다.
match_property-name=value	지정된 등록 정보 이름 및 값과 일치하는 데이터를 나열합니다.

질의 매개변수: props

props 질의 매개변수는 최종 사용자가 메타데이터에 액세스할 수 있도록 GET, POST 및 PUT 명령에 사용할 수 있습니다. 최종 사용자는 질의 매개변수 props를 true로 설정하여 이 기능을 요청합니다. GET 및 PUT 작업의 경우 반환된 JSON 객체에는 요청된 데이터와 등록 정보의 메타데이터 목록이 포함됩니다. POST의 경우 사용자가 리소스를 제대로 만들 수 있도록 메타데이터만 반환됩니다.

표 3 등록 정보 메타데이터 값

등록 정보	설명
name	등록 정보 이름
label	등록 정보의 설명
immutable	등록 정보를 수정할 수 없음을 나타내는 플래그입니다.
type	String, Integer, Boolean과 같은 등록 정보 유형
choices	열거된 등록 정보의 경우 사용 가능한 값 배열

질의 매개변수: limit

limit 질의는 많은 수의 요소를 반환할 수 있는 여러 GET 명령에 사용하여 반환될 최대 요소 수를 제한할 수 있습니다.

질의 매개변수: start

시간 값을 지원하는 리소스의 경우 index는 20170531T01:13:58과 같은 시간 값일 수 있으며, UTC 시간으로 표시해야 합니다.

질의 매개변수: depth

depth 질의 매개변수를 GET 명령에 사용하여 리소스 목록을 검색할 수 있습니다. 이 매개변수는 반환된 목록의 세부정보 레벨을 지정하는 데 사용됩니다. depth 숫자가 클수록 세부정보가 더 많이 반환됩니다. 예를 들어, 다음과 같습니다.

- /api/...?depth=0 – 노드의 등록 정보와 하위 항목 이름만 반환합니다.
- /api/...?depth=1 – 노드의 등록 정보와 하위 항목의 이름 및 등록 정보, 2차 하위 항목의 이름만 반환합니다.
- /api/...?depth=2 – 노드의 등록 정보와 하위 항목의 이름 및 등록 정보, 2차 하위 항목의 depth=0 출력을 반환합니다.

주 - depth 질의 매개변수는 /api/log/v1을 사용하는 로그 나열, /api/storage/v1을 사용하는 풀, 프로젝트, 파일 시스템 및 LUN 나열에 지원되지 않습니다.

질의 매개변수 depth에 대한 요청의 예:

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

이 예에서 사용자 목록은 depth=2까지에 해당하는 세부정보와 함께 반환됩니다.

응답 예:

```
HTTP/1.1 200 OK
Content-Type: application/json
Content-Length: 1558
X-Zfssa-Access-API: 1.0

{"users":
  [{
    "name": "root",
    "properties": {
      "logname": "root",
      "fullname": "Super-User",
      "initial_password": "password",
      "require_annotation": false
    },
    "children": [{
      "name": "preferences",
      "properties": {
        "locale": "C",
        "login_screen": "status/dashboard",
        "session_timeout": 15,
        "advanced_analytics": false
      },
      "children": [{
        "name": "keys",
        "properties": {},
        "children": [],
        "list": []
      }],
      "list": []
    }],
    "list": [],
    "href": "/api/user/v1/users/root"
  },
  {
    "name": "admin2",
    "properties": {
      "logname": "admin2",
```

```

    "fullname": "Administrator",
    "initial_password": "password",
    "require_annotation": false,
    "roles": ["basic"],
    "kiosk_mode": false,
    "kiosk_screen": "status/dashboard"
  },
  "children": [{
    "name": "exceptions",
    "properties": {},
    "children": [],
    "list": [{
      "name": "auth-000",
      "properties": {
        "scope": "stat",
        "drilldowns": "*",
        "allow_create": false,
        "allow_read": true
      },
      "children": [],
      "list": []
    },
    {
      "name": "auth-001",
      "properties": {
        "scope": "ad",
        "name": "*",
        "allow_domain": true,
        "allow_workgroup": false
      },
      "children": [],
      "list": []
    }
  ]
}, {
  "name": "preferences",
  "properties": {
    "locale": "C",
    "login_screen": "status/dashboard",
    "session_timeout": 15,
    "advanced_analytics": false
  },
  "children": [{
    "name": "keys",
    "properties": {},
    "children": [],
    "list": ["key-000"]
  }],
  "list": []
}],
"list": [],
"href": "/api/user/v1/users/admin2"
}]
}

```

질의 매개변수: match

`match_property-name=value` 질의 매개변수를 GET 명령에 사용하여 리소스 목록을 검색할 수 있습니다. 지정된 등록 정보 이름 및 값과 일치하는 데이터 목록이 반환됩니다. 예를 들어, 다음과 같습니다.

- `/api/...?depth=0&match_kiosk_mode=true - kiosk_mode=true`로 필터링된 목록을 하위 항목의 이름과 함께 반환합니다.
- `/api/...?depth=1&match_kiosk_mode=true - kiosk_mode=true`로 필터링된 목록을 `depth=1`까지에 해당하는 세부정보와 함께 반환합니다.
- `/api/...?depth=2&match_Fullname='Super*&kiosk_mode=true - fullname`에 Super 포함 및 `kiosk_mode=true`로 필터링된 목록을 `depth=2`까지에 해당하는 세부정보와 함께 반환합니다.

주 - `match_property-name=value` 질의 매개변수는 `/api/log/v1`을 사용하는 로그 나열, `/api/storage/v1`을 사용하는 풀, 프로젝트, 파일 시스템 및 LUN 나열에 지원되지 않습니다.

어플라이언스 오류

오류는 다음 결함 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.

액세스 설정

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

보안 프로토콜 및 암호화 설정

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
Content-Type: application/json
```

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

결과 예(가독성을 위해 인위적으로 줄바꿈이 설정됨):

```
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:
DH-DSS-AES256-GCM-SHA384:DHE-DSS-AES256-GCM-SHA384:DH-RSA-AES256-GCM-SHA384:DHE-
RSA-AES256-GCM-SHA384:DHE-RSA-AES256-SHA256:DHE-DSS-AES256-SHA256:DH-RSA-AES256-
SHA256:DH-DSS-AES256-SHA256:DHE-RSA-AES256-SHA:DHE-DSS-AES256-SHA:DH-RSA-AES256-
SHA:DH-DSS-AES256-SHA:DHE-RSA-CAMELLIA256-SHA:DHE-DSS-CAMELLIA256-SHA:DH-RSA-
CAMELLIA256-SHA:DH-DSS-CAMELLIA256-SHA:AES256-GCM-SHA384:AES256-SHA256:AES256-
SHA:CAMELLIA256-SHA:SRP-DSS-AES-128-CBC-SHA:SRP-RSA-AES-128-CBC-SHA:SRP-AES-128-
CBC-SHA:DH-DSS-AES128-GCM-SHA256:DHE-DSS-AES128-GCM-SHA256:DH-RSA-AES128-GCM-
SHA256:DHE-RSA-AES128-GCM-SHA256:DHE-RSA-AES128-SHA256:DHE-DSS-AES128-SHA256:
DH-RSA-AES128-SHA256:DH-DSS-AES128-SHA256:DHE-RSA-AES128-SHA:DHE-DSS-AES128-SHA:
DH-RSA-AES128-SHA:DH-DSS-AES128-SHA:DHE-RSA-CAMELLIA128-SHA:DHE-DSS-CAMELLIA128-
SHA:DH-RSA-CAMELLIA128-SHA:DH-DSS-CAMELLIA128-SHA:AES128-GCM-SHA256:AES128-SHA256:
AES128-SHA:CAMELLIA128-SHA:SRP-DSS-3DES-EDE-CBC-SHA:SRP-RSA-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"
  }
}
```

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

요청 예(가독성을 위해 인위적으로 줄바꿈이 설정됨):

```
PUT /api/service/v1/services/https HTTP/1.1
Host: zfs-storage.example.com
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",
    "DHE-RSA-AES256-SHA", "DHE-DSS-AES256-SHA", "DH-RSA-AES256-SHA", "DH-DSS-AES256-SHA",
    "DHE-RSA-CAMELLIA256-SHA", "DHE-DSS-CAMELLIA256-SHA", "DH-RSA-CAMELLIA256-SHA",
    "DH-DSS-CAMELLIA256-SHA", "AES256-SHA", "CAMELLIA256-SHA", "SRP-DSS-AES-128-CBC-SHA",
    "SRP-RSA-AES-128-CBC-SHA", "SRP-AES-128-CBC-SHA", "DHE-RSA-AES128-SHA", "DHE-DSS-AES128-
    SHA",
    "DH-RSA-AES128-SHA", "DH-DSS-AES128-SHA", "DHE-RSA-CAMELLIA128-SHA", "DHE-DSS-CAMELLIA128-
    SHA",
    "DH-RSA-CAMELLIA128-SHA", "DH-DSS-CAMELLIA128-SHA", "AES128-SHA", "CAMELLIA128-SHA",
    "SRP-DSS-3DES-EDE-CBC-SHA", "SRP-RSA-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"]
}
```

결과 예(가독성을 위해 인위적으로 줄바꿈이 설정됨):

```
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:
    DHE-RSA-AES256-SHA:DHE-DSS-AES256-SHA:DH-RSA-AES256-SHA:DH-DSS-AES256-SHA:DHE-RSA-
    CAMELLIA256-SHA:DHE-DSS-CAMELLIA256-SHA:DH-RSA-CAMELLIA256-SHA:DH-DSS-CAMELLIA256-
    SHA:AES256-SHA:CAMELLIA256-SHA:SRP-DSS-AES-128-CBC-SHA:SRP-RSA-AES-128-CBC-SHA:SRP-
    AES-128-CBC-SHA:DHE-RSA-AES128-SHA:DHE-DSS-AES128-SHA:DH-RSA-AES128-SHA:DH-DSS-
    AES128-SHA:DHE-RSA-CAMELLIA128-SHA:DHE-DSS-CAMELLIA128-SHA:DH-RSA-CAMELLIA128-SHA:
    DH-DSS-CAMELLIA128-SHA:AES128-SHA:CAMELLIA128-SHA:SRP-DSS-3DES-EDE-CBC-SHA:SRP-RSA-
    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.

RESTful API 작업

액세스 서비스는 Oracle ZFS Storage Appliance의 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 URL `http://hostname:215/api/access/v1`을 사용합니다.

다른 서비스에 액세스하려면 액세스 서비스를 사용하여 로그인하여 사용할 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	/api/access/v1	RESTful API 서비스 액세스 지점을 나열합니다.
POST	/api/access/v1	로그인 세션을 만듭니다.
DELETE	/api/access/v1	세션에서 로그아웃합니다.

서비스 나열

`list services` 명령은 사용할 서비스 액세스 URI를 나열합니다. 로그인 세션을 원하지 않는 경우 `list services` 명령을 적합한 자격 증명과 함께 사용하여 사용할 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
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

{
  "access": {
    "services": [{
      "version": "1.0",
      "name": "appliance",
      "uri": "https://zfs-storage.example.com:215/api/appliance/v1"
    }, {
      "version": "1.0",
      "name": "nas",
      "uri": "https://zfs-storage.example.com:215/api/nas/v1"
    }, {
      "version": "1.0",
      "name": "replication",
      "uri": "https://zfs-storage.example.com:215/api/replication/v1"
    }, {
      "version": "1.0",
      "name": "san",
      "uri": "https://zfs-storage.example.com:215/api/san/v1"
    } ... ]
  }
}
```

서비스 가져오기 명령

`get service`는 사용 가능한 모든 명령 목록을 포함하여 해당 서비스에 대한 정보를 반환합니다.

요청 예:

```
GET /api/appliance/v1 HTTP/1.1
Host: zfs-storage.example.com
X-Auth-Session: guigqpQRE4g89ngb
```

응답 예:

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

{
  "service": {
    "name": "appliance",
    "methods": [
      {
        "description": "Get appliance RESTful services",
        "path": "/apis",
        "request": "GET"
      },
      {
        "description": "Get appliance RESTful service properties",
        "path": "/apis/<api:path>",
        "request": "GET"
      }
    ]
  }
}
```

```

    },
    {
      "description": "Create a new alert threshold watch",
      "path": "/alerts/thresholds",
      "request": "POST"
    }, ... ]
  }
}

```

인증 세션

POST 요청을 전송하여 액세스 서비스에서 인증 세션 ID를 얻습니다. 이 인증 세션 ID는 다른 모든 서비스에서 ID 자격 증명으로 사용될 수 있습니다. 인증 ID는 사용자의 세션 시간 초과 등록 정보에 의해 설정된 시간 초과 기간 이후 무효화됩니다. 기본값은 대개 15분입니다. DELETE 요청은 세션 ID를 로그아웃 및 무효화하는 데 사용될 수 있습니다.

클라이언트는 각 요청과 함께 인증 정보를 재전송하므로 인증 세션이 필요하지 않습니다. RESTful API 작업이 stateless이므로 인증 ID만 저장됩니다.

로그인 세션

빈 POST 요청은 새 로그인 세션을 요청합니다. 성공 시 HTTP 상태 201이 사용 가능한 RESTful API 서비스 목록을 포함하는 단일 등록 정보 "access"가 있는 JSON 객체와 함께 반환됩니다.

로그인 요청 예:

```

POST /api/access/v1 HTTP/1.1
Host: zfs-storage.example.com
X-Auth-User: root
X-Auth-Key: password-xxx

```

성공한 로그인은 HTTP 상태 201(Created)은 물론 X-Auth-Session HTTP 헤더를 통해 세션 ID도 반환합니다. 응답 본문은 이 로그인을 통해 액세스할 수 있는 서비스 목록을 포함합니다.

응답 헤더:

```

HTTP/1.1 201 Created
X-Auth-Session: guigqpQRE4g89ngb
Content-Type: application/json
Content-Length: 378
X-Zfssa-Access-API: 1.0

```

```

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

```

세션 로그아웃

빈 `DELETE`는 세션을 로그아웃 및 무효화는 요청을 전송합니다.

로그아웃 요청 예:

```
DELETE /api/access/v1 HTTP/1.1
X-Auth-Session: guigqpQRE4g89ngb
```

응답 예:

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

RESTful API 경고 서비스

경보 RESTful API 서비스를 사용하면 게시된 경고에 대한 경고 임계값 및 응답을 구성할 수 있습니다.

경보 서비스 명령

다음 표는 경고 서비스 명령을 보여 줍니다.

표 6 경고 서비스 명령

요청	경로 <i>/api/alert/v1</i> 에 추가	설명
GET	<i>/api/alert/v1</i> 만 사용	경보 서비스 명령을 나열합니다.
POST	<i>/thresholds</i>	새 경고 임계값 위치를 만듭니다.
GET	<i>/thresholds/threshold</i>	지정된 경고 임계값 위치 등록 정보를 가져옵니다.
GET	<i>/thresholds</i>	모든 경고 임계값 위치 객체를 나열합니다.
PUT	<i>/thresholds/threshold</i>	지정된 경고 임계값 위치 객체를 수정합니다.
DELETE	<i>/thresholds/threshold</i>	지정된 임계값 객체를 삭제합니다.
POST	<i>/actions</i>	새 경고 조치를 만듭니다.
GET	<i>/actions/actions</i>	지정된 경고 조치 등록 정보를 가져옵니다.
GET	<i>/actions</i>	모든 경고 조치 객체를 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 가져옵니다.
PUT	<i>/actions/actions/action</i>	지정된 경고 조치 작업 객체를 수정합니다.
DELETE	<i>/actions/actions/action</i>	지정된 작업 객체를 삭제합니다.
GET	<i>/events</i>	새 경고 이벤트를 수신합니다.

경보 임계값

임계값은 사용자정의 경보 위치를 만들도록 설정될 수 있습니다. 다음 표에는 경보 임계값 관리를 위한 일반적인 등록 정보가 나와 있습니다. 전체 참조는 CLI 도움말을 참조하십시오.

표 7 경보 임계값

등록 정보	유형	설명
uuid	기본값	위치에 대한 고유 식별자("immutable")
statname	AnalyticsStatistics	위치 통계 ["cpu.utilization", "arc.accesses", "arc.size", "arc.l2_bytes", "arc.l2_accesses", "arc.l2_size", "syscap.bytesused", "syscap.percentused", "metacap.bytesused", "metacap.percentused", "repl.bytes", "repl.ops", "shadow.kilobytes", "shadow.ops", "shadow.requests", "io.bytes", "io.ops", "datalink.kilobytes", "nic.kilobytes", "net.kilobytes", "ftp.kilobytes", "fc.bytes", "fc.ops", "http.reqs", "ndmp.bytes", "ndmp.diskkb", "ndmp.ops", "nfs2.bytes", "nfs2.ops", "nfs3.bytes", "nfs3.ops", "nfs4.bytes", "nfs4.ops", "nfs4-1.bytes", "nfs4-1.ops", "sftp.kilobytes", "smb.ops", "srp.bytes", "srp.ops", "iscsi.bytes", "iscsi.ops"]
type	ChooseOne	stat가 제한을 초과하거나(normal) 제한 아래로 떨어질 때(inverted) 경보를 게시할지 여부 ["normal", "inverted"]
limit	PositiveInteger	통계에 대한 제한 값
minpost	Duration	최소 시간 조건은 경보를 게시하기 전에 유지해야 합니다.
days	ChooseOne	특정 일에만 경보 게시 ["all", "weekdays", "weekends"]
window_start	TimeOfDay	window_start와 window_end 사이에만 경보 게시 ["none", "00:00", "00:30", "01:00", "01:30", "02:00", "02:30", "03:00", "03:30", "04:00", "04:30", "05:00", "05:30", "06:00", "06:30", "07:00", "07:30", "08:00", "08:30", "09:00", "09:30", "10:00", "10:30", "11:00", "11:30", "12:00", "12:30", "13:00", "13:30", "14:00", "14:30", "15:00", "15:30", "16:00", "16:30", "17:00", "17:30", "18:00", "18:30", "19:00", "19:30", "20:00", "20:30", "21:00", "21:30", "22:00", "22:30", "23:00", "23:30"]
window_end	TimeOfDay	window_start와 window_end 사이에만 경보 게시 ["none", "00:00", "00:30", "01:00", "01:30", "02:00", "02:30", "03:00", "03:30", "04:00", "04:30", "05:00", "05:30", "06:00", "06:30", "07:00", "07:30", "08:00", "08:30", "09:00", "09:30", "10:00", "10:30", "11:00", "11:30", "12:00", "12:30", "13:00", "13:30", "14:00", "14:30", "15:00", "15:30", "16:00", "16:30", "17:00", "17:30", "18:00", "18:30", "19:00", "19:30", "20:00", "20:30", "21:00", "21:30", "22:00", "22:30", "23:00", "23:30"]("immutable")
frequency	Duration	경보를 재게시하기 전 최소 시간
minclear	Duration	"all clear" 경보를 재게시하기 전 정상 상태의 최소 시간

경보 임계값 나열

구성된 모든 경보 임계값을 나열합니다.

요청 예:

```
GET /api/alert/v1/thresholds HTTP/1.1
Authorization: Basic abcd123MWE=
Host: zfs-storage.example.com:215
Accept: application/json
```

응답 예:

```
HTTP/1.1 200 OK
Date: Tue, 27 Aug 2013 17:38:40 GMT
X-Zfssa-Appliance-API: 1.0
Content-Type: application/json
Content-Length: 689

{
  "thresholds": [
    {
      "days": "all",
      "frequency": 300,
      "href": "/api/alert/v1/thresholds/
        bec758cb-346e-6a7d-c211-b320c09ef6a6",
      "limit": 500,
      "minclear": 300,
      "minpost": 300,
      "statname": "cpu.utilization",
      "threshold": "threshold-000",
      "type": "normal",
      "uuid": "bec758cb-346e-6a7d-c211-b320c09ef6a6",
      "window_end": 0,
      "window_start": -1
    },
    {
      "days": "all",
      "frequency": 300,
      "href": "/api/alert/v1/thresholds/
        475799d8-32c8-6ff6-882c-aa3b66e3a5a2",
      "limit": 100000,
      "minclear": 600,
      "minpost": 300,
      "statname": "datalink.kilobytes",
      "threshold": "threshold-001",
      "type": "normal",
      "uuid": "475799d8-32c8-6ff6-882c-aa3b66e3a5a2",
      "window_end": 300,
      "window_start": 1200
    }
  ]
}
```

경보 임계값 가져오기

단일 경보 임계값에 대한 등록 정보를 나열합니다.

요청 예:

```
GET /api/alert/v1/thresholds/1b15d405-75c4-4c0c-e0f6-8a108165b874
HTTP/1.1
Authorization: Basic abcd123MWE=
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: 363
```

```
{
  "threshold": {
    "days": "weekdays",
    "frequency": 300,
    "href": "/api/alert/v1/thresholds/1b15d405-75c4-4c0c-e0f6-8a108165b874",
    "limit": 100000,
    "minclear": 300,
    "minpost": 300,
    "statname": "datalink.kilobytes",
    "type": "normal",
    "uuid": "1b15d405-75c4-4c0c-e0f6-8a108165b874",
    "window_end": 0,
    "window_start": -1
  }
}
```

경보 임계값 만들기

경보 임계값을 만듭니다.

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}
```

응답 예:

```
HTTP/1.1 201 Created
X-Zfssa-Appliance-API: 1.0
Content-Type: application/json
Content-Length: 321
Location: /api/alert/v1/thresholds/1b15d405-75c4-4c0c-e0f6-8a108165b874

{
  "threshold": {
    "href": "/api/alert/v1/alerts/thresholds/1b15d405-75c4-4c0c-e0f6-8a108165b874",
```

```
    }
  }
}
```

경보 임계값 수정

지정된 경보 임계값에 대한 등록 정보를 수정합니다.

요청 예:

```
PUT /api/alert/v1/thresholds/1b15d405-75c4-4c0c-e0f6-8a108165b874
HTTP/1.1
Authorization: Basic abcd123MWE=
Host: zfs-storage.example.com:215

{"days": "weekdays"}
```

응답 예:

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

{
  "threshold": {
    "days": "weekdays",
    ...
  }
}
```

경보 임계값 삭제

지정된 경보 임계값을 삭제합니다.

요청 예:

```
DELETE /api/alert/v1/thresholds/475799d8-32c8-6ff6-882c-aa3b66e3a5a2
HTTP/1.1
Authorization: Basic abcd123MWE=
Host: zfs-storage.example.com:215
```

응답 예:

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

경보 조치

`category` 등록 정보는 정의 중인 경보 조치 유형을 결정합니다. 각 범주에는 자체 등록 정보 세트가 정의되어 있습니다.

지원되는 범주:

- ad
- all
- appliance_software
- backup
- cluster
- custom
- hardware
- hardware_faults
- ndmp
- network
- replication
- replication_source
- replication_target
- restore
- scrk
- shadow
- smf
- thresholds
- zfs_pool

표 8 경보 조치 "ad"

등록 정보	유형	설명
active_directory_degraded	부울	필터는 active_directory_degraded 이벤트와 일치해야 합니다(true 또는 false).
smb_kerberos_client_authentication_degraded	부울	필터는 mb_kerberos_client_authentication_degraded 이벤트와 일치해야 합니다(true 또는 false).

표 9 경보 조치 "all"

등록 정보	유형	설명
all_defects	부울	필터는 all_defects 이벤트와 일치해야 합니다.
service_alerts	부울	필터는 service_alerts 이벤트와 일치해야 합니다(true 또는 false).
all_hardware_faults	부울	필터는 all_hardware_faults 이벤트와 일치해야 합니다(true 또는 false).

표 10 경보 조치 "appliance_software"

등록 정보	유형	설명
obstacles_to_system_software_update	부울	필터는 obstacles_to_system_software_update 이벤트와 일치해야 합니다(true 또는 false).
operating_system_kernel_panic	부울	필터는 operating_system_kernel_panic 이벤트와 일치해야 합니다(true 또는 false).

표 11 경보 조치 "backup"

등록 정보	유형	설명
backup_finished	부울	필터는 backup_finished 이벤트와 일치해야 합니다(true 또는 false).
backup_started	부울	필터는 backup_started 이벤트와 일치해야 합니다(true 또는 false).

표 12 경보 조치 "cluster"

등록 정보	유형	설명
cluster_i/o_link_down	부울	필터는 cluster_i/o_link_down 이벤트와 일치해야 합니다(true 또는 false).
cluster_i/o_link_failed	부울	필터는 cluster_i/o_link_failed 이벤트와 일치해야 합니다(true 또는 false).
cluster_i/o_link_up	부울	필터는 cluster_i/o_link_up 이벤트와 일치해야 합니다(true 또는 false).
unexpected_peer_error_occurred	부울	필터는 unexpected_peer_error_occurred 이벤트와 일치해야 합니다(true 또는 false).
communication_to_peer_lost	부울	필터는 communication_to_peer_lost 이벤트와 일치해야 합니다(true 또는 false).
cluster_peer_panicked	부울	필터는 cluster_peer_panicked 이벤트와 일치해야 합니다(true 또는 false).
failed_to_set_sp_root_password_on_cluster_peer	부울	필터는 failed_to_set_sp_root_password_on_cluster_peer 이벤트와 일치해야 합니다(true 또는 false).
cluster_rejoin_failed_on_peer	부울	필터는 cluster_rejoin_failed_on_peer 이벤트와 일치해야 합니다(true 또는 false).
cluster_rejoin_mismatch_on_peer	부울	필터는 cluster_rejoin_mismatch_on_peer 이벤트와 일치해야 합니다(true 또는 false).
cluster_rejoin_completed_on_peer	부울	필터는 cluster_rejoin_completed_on_peer 이벤트와 일치해야 합니다(true 또는 false).
cluster_peer_lost_communication_token	부울	필터는 cluster_peer_lost_communication_token 이벤트와 일치해야 합니다(true 또는 false).
cluster_rejoin_failed	부울	필터는 cluster_rejoin_failed 이벤트와 일치해야 합니다(true 또는 false).
cluster_rejoin_mismatch	부울	필터는 cluster_rejoin_mismatch 이벤트와 일치해야 합니다(true 또는 false).

등록 정보	유형	설명
cluster_rejoin_completed	부울	필터는 cluster_rejoin_completed 이벤트와 일치해야 합니다(true 또는 false).
cluster_takeover_complete	부울	필터는 cluster_takeover_complete 이벤트와 일치해야 합니다(true 또는 false).
resource_import_failed_during_cluster_takeover	부울	필터는 resource_import_failed_during_cluster_takeover 이벤트와 일치해야 합니다(true 또는 false).
local_cluster_communication_token_lost	부울	필터는 local_cluster_communication_token_lost 이벤트와 일치해야 합니다(true 또는 false).

표 13 경보 조치 "custom"

등록 정보	유형	설명
patterns	기본값	FMA 이벤트 patterns

표 14 경보 조치 "hardware"

등록 정보	유형	설명
fibre_channel_port_down	부울	필터는 fibre_channel_port_down 이벤트와 일치해야 합니다(true 또는 false).
multiple_transient_fibre_channel_port_status_changes	부울	필터는 multiple_transient_fibre_channel_port_status_changes 이벤트와 일치해야 합니다(true 또는 false).
transient_fibre_channel_port_status_change	부울	필터는 transient_fibre_channel_port_status_change 이벤트와 일치해야 합니다(true 또는 false).
fibre_channel_port_up	부울	필터는 fibre_channel_port_up 이벤트와 일치해야 합니다(true 또는 false).
network_port_down	부울	필터는 network_port_down 이벤트와 일치해야 합니다(true 또는 false).
network_port_up	부울	필터는 network_port_up 이벤트와 일치해야 합니다(true 또는 false).
chassis_connected_to_system	부울	필터는 chassis_connected_to_system 이벤트와 일치해야 합니다(true 또는 false).
chassis_removed	부울	필터는 chassis_removed 이벤트와 일치해야 합니다(true 또는 false).
hardware_component_inserted	부울	필터는 hardware_component_inserted 이벤트와 일치해야 합니다(true 또는 false).
hardware_component_removed	부울	필터는 hardware_component_removed 이벤트와 일치해야 합니다(true 또는 false).
disk_inserted	부울	필터는 disk_inserted 이벤트와 일치해야 합니다(true 또는 false).
disk_removed	부울	필터는 disk_removed 이벤트와 일치해야 합니다(true 또는 false).

등록 정보	유형	설명
i/o_path_added	부울	필터는 i/o_path_added 이벤트와 일치해야 합니다(true 또는 false).
i/o_path_removed	부울	필터는 i/o_path_removed 이벤트와 일치해야 합니다(true 또는 false).
service_processor_offline_or_unavailable	부울	필터는 service_processor_offline_or_unavailable 이벤트와 일치해야 합니다(true 또는 false).
service_processor_online_after_outage	부울	필터는 service_processor_online_after_outage 이벤트와 일치해야 합니다(true 또는 false).
failed_to_set_root_password_on_service_processor	부울	필터는 failed_to_set_root_password_on_service_processor 이벤트와 일치해야 합니다(true 또는 false).

표 15 경보 조치 "hardware_faults"

등록 정보	유형	설명
all_hardware_faults	부울	필터는 all_hardware_faults 이벤트와 일치해야 합니다(true 또는 false).

표 16 경보 조치 "ndmp"

등록 정보	유형	설명
invalid_ndmp_restore	부울	필터는 invalid_ndmp_restore 이벤트와 일치해야 합니다(true 또는 false).
backup_finished	부울	필터는 backup_finished 이벤트와 일치해야 합니다(true 또는 false).
backup_started	부울	필터는 backup_started 이벤트와 일치해야 합니다(true 또는 false).
restore_finished	부울	필터는 restore_finished 이벤트와 일치해야 합니다(true 또는 false).
restore_started	부울	필터는 restore_started 이벤트와 일치해야 합니다(true 또는 false).

표 17 경보 조치 "network"

등록 정보	유형	설명
datalink_failed	부울	필터는 datalink_failed 이벤트와 일치해야 합니다(true 또는 false).
datalink_ok	부울	필터는 datalink_ok 이벤트와 일치해야 합니다(true 또는 false).
network_port_down	부울	필터는 network_port_down 이벤트와 일치해야 합니다(true 또는 false).
network_port_up	부울	필터는 network_port_up 이벤트와 일치해야 합니다(true 또는 false).
ip_address_conflict	부울	필터는 ip_address_conflict 이벤트와 일치해야 합니다(true 또는 false).

등록 정보	유형	설명
ip_address_conflict_resolved	부울	필터는 ip_address_conflict_resolved 이벤트와 일치해야 합니다(true 또는 false).
ip_interface_degraded	부울	필터는 ip_interface_degraded 이벤트와 일치해야 합니다(true 또는 false).
ip_interface_failed	부울	필터는 ip_interface_failed 이벤트와 일치해야 합니다(true 또는 false).
ip_interface_ok	부울	필터는 ip_interface_ok 이벤트와 일치해야 합니다(true 또는 false).

표 18 경보 조치 "replication"

등록 정보	유형	설명
receive_failed_(unsupported_version)	부울	필터는 receive_failed_(unsupported_version) 이벤트와 일치해야 합니다(true 또는 false).
receive_failed_(cancelled)	부울	필터는 receive_failed_(cancelled) 이벤트와 일치해야 합니다(true 또는 false).
receive_failed_(all_others)	부울	필터는 receive_failed_(all_others) 이벤트와 일치해야 합니다(true 또는 false).
receive_failed_(out_of_space)	부울	필터는 receive_failed_(out_of_space) 이벤트와 일치해야 합니다(true 또는 false).
receive_failed_(package_not_upgraded)	부울	필터는 receive_failed_(package_not_upgraded) 이벤트와 일치해야 합니다(true 또는 false).
receive_finished	부울	필터는 receive_finished 이벤트와 일치해야 합니다(true 또는 false).
receive_started	부울	필터는 receive_started 이벤트와 일치해야 합니다(true 또는 false).
send_failed_(unsupported_version)	부울	필터는 send_failed_(unsupported_version) 이벤트와 일치해야 합니다(true 또는 false).
send_failed_(cancelled)	부울	필터는 send_failed_(cancelled) 이벤트와 일치해야 합니다(true 또는 false).
send_failed_(all_others)	부울	필터는 send_failed_(all_others) 이벤트와 일치해야 합니다(true 또는 false).
send_failed_(connectivity)	부울	필터는 send_failed_(connectivity) 이벤트와 일치해야 합니다(true 또는 false).
send_failed_(out_of_space)	부울	필터는 send_failed_(out_of_space) 이벤트와 일치해야 합니다(true 또는 false).
send_failed_(remote_verification)	부울	필터는 send_failed_(remote_verification) 이벤트와 일치해야 합니다(true 또는 false).
send_finished	부울	필터는 send_finished 이벤트와 일치해야 합니다(true 또는 false).
send_skipped_(already_running)	부울	필터는 send_skipped_(already_running) 이벤트와 일치해야 합니다(true 또는 false).
send_started	부울	필터는 send_started 이벤트와 일치해야 합니다(true 또는 false).

표 19 경보 조치 "replication_source"

등록 정보	유형	설명
send_failed_(unsupported_version)	부울	필터는 send_failed_(unsupported_version) 이벤트와 일치해야 합니다(true 또는 false).
send_failed_(cancelled)	부울	필터는 send_failed_(cancelled) 이벤트와 일치해야 합니다(true 또는 false).
send_failed_(all_others)	부울	필터는 send_failed_(all_others) 이벤트와 일치해야 합니다(true 또는 false).
send_failed_(connectivity)	부울	필터는 send_failed_(connectivity) 이벤트와 일치해야 합니다(true 또는 false).
send_failed_(out_of_space)	부울	필터는 send_failed_(out_of_space) 이벤트와 일치해야 합니다(true 또는 false).
send_failed_(remote_verification)	부울	필터는 send_failed_(remote_verification) 이벤트와 일치해야 합니다(true 또는 false).
send_finished	부울	필터는 send_finished 이벤트와 일치해야 합니다(true 또는 false).
send_skipped_(already_running)	부울	필터는 send_skipped_(already_running) 이벤트와 일치해야 합니다(true 또는 false).
send_started	부울	필터는 send_started 이벤트와 일치해야 합니다(true 또는 false).

표 20 경보 조치 "replication_target"

등록 정보	유형	설명
receive_failed_(unsupported_verssion)	부울	필터는 receive_failed_(unsupported_version) 이벤트와 일치해야 합니다(true 또는 false).
receive_failed_(cancelled)	부울	필터는 receive_failed_(cancelled) 이벤트와 일치해야 합니다(true 또는 false).
receive_failed_(all_others)	부울	필터는 receive_failed_(all_others) 이벤트와 일치해야 합니다(true 또는 false).
receive_failed_(out_of_space)	부울	필터는 receive_failed_(out_of_space) 이벤트와 일치해야 합니다(true 또는 false).
receive_failed_(package_not_upgraded)	부울	필터는 receive_failed_(package_not_upgraded) 이벤트와 일치해야 합니다(true 또는 false).
receive_finished	부울	필터는 receive_finished 이벤트와 일치해야 합니다(true 또는 false).
receive_started	부울	필터는 receive_started 이벤트와 일치해야 합니다(true 또는 false).

표 21 경보 조치 "restore"

등록 정보	유형	설명
restore_finished	부울	필터는 restore_finished 이벤트와 일치해야 합니다(true 또는 false).

등록 정보	유형	설명
restore_started	부울	필터는 restore_started 이벤트와 일치해야 합니다(true 또는 false).

표 22 경보 조치 "scrk"

등록 정보	유형	설명
support_bundle_build_failed	부울	필터는 support_bundle_build_failed 이벤트와 일치해야 합니다(true 또는 false).
support_bundle_sent	부울	필터는 support_bundle_sent 이벤트와 일치해야 합니다(true 또는 false).
support_bundle_upload_failed	부울	필터는 support_bundle_upload_failed 이벤트와 일치해야 합니다(true 또는 false).
an_update_is_available_on_my_oracle_support.	부울	필터는 an_update_is_available_on_my_oracle_support 이벤트와 일치해야 합니다(true 또는 false).
no_updates_available.	부울	필터는 no_updates_available 이벤트와 일치해야 합니다(true 또는 false).
the_appliance_failed_to_verify_if_an_update_is_available.	부울	필터는 the_appliance_failed_to_verify_if_an_update_is_available 이벤트와 일치해야 합니다(true 또는 false).

표 23 경보 조치 "shadow"

등록 정보	유형	설명
shadow_migration_complete	부울	필터는 shadow_migration_complete 이벤트와 일치해야 합니다(true 또는 false).

표 24 경보 조치 "smf"

등록 정보	유형	설명
service_failures	부울	필터는 service_failures 이벤트와 일치해야 합니다(true 또는 false).

표 25 경보 조치 "thresholds"

등록 정보	유형	설명
thresholdid	기본값	경보가 일치해야 하는 위치의 UUID

표 26 경보 조치 "zfs_pool"

등록 정보	유형	설명
resilver_finished	부울	필터는 resilver_finished 이벤트와 일치해야 합니다(true 또는 false).
resilver_started	부울	필터는 resilver_started 이벤트와 일치해야 합니다(true 또는 false).

등록 정보	유형	설명
scrub_finished	부울	필터는 scrub_finished 이벤트와 일치해야 합니다 (true 또는 false).
scrub_started	부울	필터는 scrub_started 이벤트와 일치해야 합니다 (true 또는 false).
hot_spare_activated	부울	필터는 hot_spare_activated 이벤트와 일치해야 합니다 (true 또는 false).

경보 조치 나열

경보 조치 나열 명령은 모든 경보 조치를 나열합니다. 단일 리소스에 대한 데이터를 가져오려면 HTTP GET 요청을 지정된 경보 조치 리소스의 href 등록 정보에 전송합니다.

경보 조치를 가져오는 요청의 예:

```
GET /api/alert/v1/actions HTTP/1.1
Authorization: Basic abcd123MWE=
Host: zfs-storage.example.com:215
Accept: application/json
```

응답 예:

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

```
{
  "actions": [
    {
      "action": "actions-000",
      "category": "smf",
      "href": "/api/alert/v1/actions/actions-000",
      "service_failures": true
    },
    {
      "action": "actions-001",
      "category": "scrk",
      "href": "/api/alert/v1/actions/actions-001",
      "action-000": {
        "handler": "snmp_trap",
        "href": "/api/alert/v1/alerts/actions/actions-001/action-000"
      },
      "action-001": {
        "address": "admin@example.com",
        "handler": "email",
        "href": "/api/alert/v1/actions/actions-001/action-001",
        "subject": "Phone Home Alert"
      },
      "support_bundle_build_failed": true,
      "support_bundle_sent": true,
      "support_bundle_upload_failed": true
    }
  ]
}
```

```

        "action": "actions-002",
        "category": "thresholds",
        "href": "/api/alert/v1/actions/actions-002",
        "action-000": {
            "address": "admin@example.com",
            "handler": "email",
            "href": "/api/alert/v1/actions/actions-002
                        /action-000",
            "subject": "CPU Busy Alert"
        },
        "thresholdid": "b182ca05-53d3-6604-b874-ec353335704d"
    }
}

```

경보 조치 가져오기

이 명령은 경보 조치 나열과 유사하지만 지정된 경보 조치만 반환합니다.

요청 예:

```
GET /api/alert/v1/actions/actions-002 HTTP/1.1
```

응답 예:

```

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

{
  "action": {
    "category": "thresholds",
    "href": "/api/alert/v1/actions/actions-002",
    "action-000": {
      "address": "admin@example.com",
      "handler": "email",
      "href": "/api/alert/v1/alerts/actions/actions-002
                /action-000",
      "subject": "CPU Busy"
    },
    "thresholdid": "b182ca05-53d3-6604-b874-ec353335704d"
  }
}

```

경보 조치 만들기

JSON 객체를 포함하는 경보 조치 `POST` 요청을 만들 때 조치 등록 정보를 `/api/alert/v1/alerts/actions`로 전송해야 합니다. 만들 조치 유형을 선택하도록 `category` 등록 정보를 설정해야 합니다. 지정된 시스템에서 사용 가능한 모든 `category` 값은 CLI 설명서를 참조하십시오.

일반적인 `category` 값은 다음과 같습니다.

```

"ad"
"all"

```

```
"appliance_software"
"backup"
"cluster"
"custom"
"hardware"
"hardware_faults"
"ndmp"
"network"
"replication"
"replication_source"
"replication_target"
"restore"
"scrk"
"shadow",
"smf"
"thresholds"
"zfs_pool"
```

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": "hardware_faults"}
```

응답 예:

```
HTTP/1.1 201 Created
X-Zfssa-Appliance-API: 1.0
Content-Type: application/json
Content-Length: 118
Location: /api/alert/v1/actions/actions-006
```

```
{
  "action": {
    "href": "/api/alert/v1/actions/actions-006",
    "category": "hardware_faults",
    "all_hardware_faults": true
  }
}
```

경보 조치 수정

HTTP PUT 요청을 전송하면 `list` 명령에서 반환된 일부 등록 정보를 수정할 수 있습니다.

요청 예:

```
PUT /api/alert/v1/actions/actions-001 HTTP/1.1
Host: zfs-storage.example.com:215
X-Auth-Session: uerqghq84vbdv
Content-Type: application/json
Content-Length: 30
```

```
{"support_bundle_sent": false}
```

응답 예:

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

{
  "action": {
    "href": "/api/alert/v1/actions/actions-001",
    "category": "scrk",
    "support_bundle_build_failed": true,
    "support_bundle_sent": false,
    "support_bundle_upload_failed": true
  }
}
```

경보 조치 삭제

HTTP DELETE 요청을 경보 조치 href 또는 조치 href에 전송하면 지정된 리소스가 삭제됩니다. 성공한 삭제 응답은 HTTP 상태 204(No Content)입니다.

요청 예:

```
DELETE /api/alert/v1/actions/actions-003 HTTP/1.1
Authorization: Basic abcd123MWE=
Host: zfs-storage.example.com:215
```

응답 예:

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

경보 조치 항목

각 경보 조치 목록에 개별 조치 항목이 추가됩니다.

경보 항목 만들기

이렇게 하면 기존 경보 조치 그룹에 경보 조치가 추가됩니다.

요청 예:

```
POST /api/alert/v1/actions/actions-001 HTTP/1.1
Host: zfs-storage.example.com:215
X-Auth-Session: uerqghq84vbdv
Content-Type: application/json
Content-Length: 68

{"address": "admin@example.com", "handler": "email", "subject": "CPU Busy"}
```

응답 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-001/action-001

```

```

{
  "action": {
    "href": "/api/alert/v1/actions/actions-001/
              /action-001",
    "handler": "email",
    "address": "admin@example.com",
    "subject": "CPU Busy"
  }
}

```

경보 조치 수정

이렇게 하면 기존 경보 조치가 수정됩니다.

요청 예:

```

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

```

```

{"address": "admin@example.com"}

```

응답 예:

```

HTTP/1.1 202 Accepted
X-Zfssa-Appliance-API: 1.0
Content-Type: application/json
Content-Length: 176
X-Zfssa-Version: user/generic@2013.06.08,1-0

```

```

{
  "action": {
    "href": "/api/alert/v1/actions/actions-001/
              /action-000",
    "handler": "email",
    "address": "admin@example.com",
    "subject": "CPU Busy"
  }
}

```

경보 조치 항목 삭제

지정된 경보 조치의 경우 단일 조치가 삭제될 수 있습니다. 조치를 삭제하려면 DELETE 요청을 조치 href 등록 정보에 전송합니다.

조치를 삭제하는 요청의 예:

```
DELETE /api/alert/v1/actions/actions-001/action-000 HTTP/1.1
Host: zfs-storage.example.com:215
X-Auth-Session: uerqghq84vbdv

HTTP/1.1 204 No Content
```

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/v1.0/`에서 다음 Analytics Services를 사용할 수 있습니다.

표 27 Analytics 명령		
요청	경로 <i>/analytics/v1</i> 에 추가	설명
GET	<i>/analytics/v1</i> 만 사용	Analytics Services 정보를 나열합니다.
POST	<i>/worksheets</i>	새 Analytics 데이터 세트를 만듭니다.
GET	<i>/worksheets/worksheet</i>	지정된 Analytics 데이터 세트 등록 정보를 가져옵니다.
GET	<i>/worksheets</i>	모든 Analytics 데이터 세트 객체를 나열합니다.
PUT	<i>/worksheets/worksheet</i>	지정된 Analytics 데이터 세트 객체를 수정합니다.
DELETE	<i>/worksheets/worksheet</i>	지정된 워크시트 객체를 삭제합니다.
PUT	<i>/worksheets/worksheet/suspend</i>	모든 워크시트 데이터 세트를 일시중단합니다.
PUT	<i>/worksheets/worksheet/resume</i>	모든 워크시트 데이터 세트를 재개합니다.
POST	<i>/worksheets/worksheet/datasets</i>	새 워크시트 데이터 세트를 만듭니다.
GET	<i>/worksheets/worksheet/datasets/dataset</i>	지정된 워크시트 데이터 세트 등록 정보를 가져옵니다.
GET	<i>/worksheets/worksheet/datasets</i>	모든 워크시트 데이터 세트 객체를 나열합니다.
PUT	<i>/worksheets/worksheet/datasets/dataset</i>	지정된 워크시트 데이터 세트 객체를 수정합니다.
DELETE	<i>/worksheets/worksheet/datasets/dataset</i>	지정된 데이터 세트 객체를 삭제합니다.
POST	<i>/datasets</i>	새 Analytics 데이터 세트를 만듭니다.
GET	<i>/datasets/dataset</i>	지정된 Analytics 데이터 세트 등록 정보를 가져옵니다.
GET	<i>/datasets</i>	모든 Analytics 데이터 세트 객체를 나열합니다.
PUT	<i>/datasets/dataset</i>	지정된 Analytics 데이터 세트 객체를 수정합니다.

요청	경로 <i>/analytics/v1</i> 에 추가	설명
DELETE	<i>/datasets/dataset</i>	지정된 데이터 세트 객체를 삭제합니다.
PUT	<i>/datasets</i>	모든 데이터 세트를 일시중단하거나 재개합니다.
PUT	<i>/datasets/dataset/data</i>	이 데이터 세트를 저장합니다(저장되지 않은 경우).
DELETE	<i>/datasets/dataset/data</i>	이 데이터 세트에서 지정된 [단위]의 데이터를 제거합니다.
GET	<i>/settings</i>	Analytics 설정을 나열합니다.
PUT	<i>/settings</i>	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
<code>retain_second_data</code>	초 단위 데이터에 대한 시간의 보존 간격
<code>retain_minute_data</code>	분 단위 데이터에 대한 시간의 보존 간격
<code>retain_hour_data</code>	시간 단위 데이터에 대한 시간의 보존 간격
<code>hostname_lookup</code>	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 ab6rt4psMWE=
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 ab6rt4psMWE=
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 ab6rt4psMWE=
Host: zfs-storage.example.com:215
Accept: application/json
```

응답 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 ab6rt4psMWE=
Host: zfs-storage.example.com:215
Accept: application/json
```

응답 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 ab6rt4psMWE=
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 ab6rt4psMWE=
Content-Type: application/json
Content-Length: 26
```

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

응답 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-Version: user/generic@2013.06.05.7.0,1-1.12
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 ab6rt4psMWE=
Content-Type: application/json
Content-Length: 26
```

응답 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 ab6rt4psMWE=
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 ab6rt4psMWE=
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 ab6rt4psMWE=
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:

- `arc.accesses[hit/miss]`
- `arc.l2_accesses[hit/miss]`
- `arc.l2_size`
- `arc.size`
- `arc.size[component]`
- `cpu.utilization`

- cpu.utilization[mode]
- dnld.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 ab6rt4psMWE=
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 ab6rt4psMWE=
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 중단 요청 예:

```

POST /api/analytics/v1/datasets/nfs4.ops

```

```
{"suspended":true}
```

재개 요청 예:

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

응답 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
```

응답 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
```

응답 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. endtime은 ISO 8601 시간/날짜 문자열(예: 20130531T01:13:58)입니다.

요청 예:

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

응답 예:

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

데이터 세트 데이터 가져오기

Analytic 데이터 세트에서 데이터를 가져옵니다. 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. 기본값은 현재 시간입니다.
seconds	샘플 데이터를 수집할 시간(초)입니다. 기본값은 1입니다. span 및 interval 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.

start 매개변수가 제공되지 않은 경우 시작 시간은 현재 시간에서 지정된 샘플 데이터의 seconds 수를 뺀 값으로 설정됩니다. 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.

다음 목록에서는 span 및 granularity 매개변수를 결합할 수 있는 방법에 대해 설명합니다.

- 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
startTime	반환된 첫번째 샘플의 시간
sample	반환된 첫번째 샘플의 샘플 인덱스
data	샘플 데이터의 배열
min	지정된 단위 내 초당 최소값
max	지정된 단위 내 초당 최대값

startTime 등록 정보는 다음 형식 중 하나일 수 있습니다.

- ISO 8601 시간/날짜 문자열(예: 20130531T01:13:58)
- 샘플 인덱스 번호
- 문자열 리터럴 now

3초 동안의 라이브 데이터를 수집하는 요청의 예:

```
GET /api/analytics/v1/datasets/nfs4.ops%5Bfile%5D/data?start=now&seconds=3 HTTP/1.1
Authorization: Basic ab6rt4psMWE=
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: 239024557
  data:
    value: 5
    startTime: 20130912T21:42:38
    samples: 239024558
- sample: 239024558
  data:
    value: 15
    startTime: 20130912T21:42:39
    samples: 239024559
- sample: 239024559
  data:
    value: 25
```

```

      startTime:      20130912T21:42:40
      samples:        239024560

size:      3
---
```

지속 시간 1주 내 7일 동안의 라이브 데이터를 수집하는 요청의 예:

```

GET /api/analytics/v1/datasets/nfs4.ops%5Bfile%5D/data?
start=239024557&span=week&granularity=day
HTTP/1.1
Authorization: Basic ab6rt4psMWE=
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: 239024557
  data:
    value:      5
    max:        79
    min:        0
    startTime:   20130912T21:42:38
    samples:     240074328

- sample: 239110957
  data:
    value:      15
    max:        150
    min:        1
    startTime:   20130913T21:42:38
    samples:     240074328

...

- sample: 239629357
  data:
    value:      25
    max:        120
    min:        2
    startTime:   20130914T21:42:38
    samples:     240074328

size:      7
```


하드웨어 서비스

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

클러스터

클러스터 명령은 클러스터링을 설정하고 클러스터화된 리소스를 관리하는 데 사용됩니다.

표 28 클러스터 명령

요청	경로 /hardware/v1에 추가	설명
GET	/cluster	클러스터 등록 정보 및 클러스터 리소스 목록을 가져옵니다.
GET	/cluster/resources/resource:path	지정된 클러스터 리소스에 대한 등록 정보를 가져옵니다.
PUT	/cluster/resources/resource:path	지정된 클러스터 리소스를 수정합니다.
PUT	/cluster/failback	클러스터 피어에 지정된 모든 리소스를 페일백합니다.
PUT	/cluster/takeover	클러스터 피어에 지정된 모든 리소스를 인계합니다.
PUT	/cluster/unconfigure	클러스터화된 어플라이언스를 독립형 모드로 구성 해제합니다.
GET	/cluster/links	클러스터 카드 링크 상태를 가져옵니다.
PUT	/cluster/setup	초기 클러스터 설정을 실행합니다.

클러스터 등록 정보 가져오기

현재 클러스터 구성 상태 및 리소스 등록 정보를 가져옵니다.

요청 예:

```
GET /api/hardware/v1/cluster HTTP/1.1
Authorization: Basic abcd45sMWE=
Host: zfs-storage.example.com:215
Accept: application/json
```

응답 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": {
    "description": "Clustering is not configured",
    "peer_asn": "",
    "peer_description": "",
    "peer_hostname": "",
    "peer_state": "",
    "resources": {
      "net/ixgbe0": {
        "details": ["ipaddr-1"],
        "href": "/hardware/v1/cluster/resources/resources/net/ixgbe0",
        "owner": "admin1",
        "type": "singleton",
        "user_label": "Untitled Interface"
      },
      "zfs/zfs-storage-1": {
        "details": ["821G"],
        "href": "/hardware/v1/cluster/resources/resources/zfs/zfs-storage-1",
        "owner": "admin1",
        "type": "singleton",
        "user_label": ""
      }
    },
    "state": "AKCS_UNCONFIGURED"
  }
}

```

클러스터 리소스 가져오기

클러스터 리소스에서 href 등록 정보를 따르면 해당 단일 클러스터 리소스에 대한 데이터를 가져올 수 있습니다. 이전 예에서 두 가지 리소스인 `/hardware/v1/cluster/resources/resources/zfs/zfs-storage-1` 및 `/hardware/v1/cluster/resources/resources/net/ixgbe0`을 사용할 수 있습니다.

클러스터 리소스 수정

시스템이 클러스터화되면 이 명령을 사용하여 각 클러스터 리소스에 대한 등록 정보를 수정할 수 있습니다. 자세한 내용은 CLI "구성 클러스터 리소스"를 참조하십시오.

클러스터 명령

클러스터에서 지원하는 명령은 `failback`, `takeover` 및 `unconfigure`입니다. 모든 명령은 추가된 명령 이름과 함께 클러스터 리소스에 대한 PUT 요청을 수행합니다. 성공 시 두 명령은 HTTP 상태 202(Accepted)를 반환합니다.

파일백은 비동기 작업입니다. REST 클라이언트가 PUT 명령을 사용하여 파일백 요청을 전송하면 요청이 성공적으로 수신된 후 HTTP 상태 202(Accepted)가 반환됩니다. 클라이언트는 경보를 수신하거나 클러스터 상태를 폴링하여 파일백 진행 상황을 모니터링해야 합니다.

요청 예:

```
PUT /api/hardware/v1/cluster/failback HTTP/1.1
Authorization: Basic abcd123MWE=
Host: zfs-storage.example.com:215
```

결과 예:

```
HTTP/1.1 202 Accepted
X-Zfssa-Appliance-API: 1.0
```

클러스터가 명령을 수락하는 올바른 상태가 아닌 경우 HTTP 상태 409(Conflict)가 반환됩니다.

클러스터 링크

이 명령은 클러스터 카드의 현재 링크 상태를 반환합니다. 출력은 `aksh` 명령 "구성 클러스터 링크"와 동일합니다. 클러스터 케이블 연결에 문제가 없는지 확인하기 위해 클러스터 설정을 실행하기 전에 이 명령을 실행하는 것이 좋습니다. 설정을 실행하기 전에 모든 링크는 `AKCIOS_ACTIVE` 상태여야 합니다.

요청 예:

```
GET /api/hardware/v1/cluster/links HTTP/1.1
Authorization: Basic abcd123MWE=
Host: zfs-storage.example.com:215
Accept: application/json
```

응답 예:

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

{
  "links": {
    "clustron2_embedded:0/clustron_uart:0 = AKCIOS_TIMEOUT\n
    clustron2_embedded:0/clustron_uart:1 = AKCIOS_TIMEOUT\n
    clustron2_embedded:0/dlpi:0 = AKCIOS_TIMEOUT"
  }
}
```

클러스터 설정

`setup cluster` 명령은 시스템에 대한 초기 클러스터링을 설정합니다. 모든 클러스터 링크는 `AKCIOS_ACTIVE` 상태여야 하며 피어 시스템은 전원이 켜져야 하지만 구성되지는 않아야 합니다. 그렇지 않으면 이 명령이 실패합니다.

요청 예:

```
PUT /api/hardware/v1/cluster/setup HTTP/1.1
Authorization: Basic abcd123MWE=
Host: zfs-storage.example.com:215
Accept: application/json

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

결과 예:

```
HTTP/1.1 202 Accepted
X-Zfssa-Appliance-API: 1.0
```

새시

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

표 29 하드웨어 명령

요청	경로 /hardware/v1.0에 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

등록 정보	유형	설명
type	string	새시 스토리지 유형
faulted	boolean	결함 표시기
fru	string	새시의 FMRI 표현
uuid	string	새시 uuid 식별자

요청 예:

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

응답 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 ZS3-1",
    "name": "cairo",
    "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/nas/v1/chassis/chassis-001 HTTP/1.1
Host: zfs-storage.example.com
Accept: application/json
```

응답 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,
      "manufacturer": "HITACHI",
      "model": "H7230AS60SUN3.0T",
      "pathcount": 4,
      "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 포함된 구성요소 등록 정보를 포함합니다.

등록 정보	유형	설명
device	string	FRU 장치 ID
faulted	boolean	FRU에 결함이 있는지 여부를 나타내는 플래그
fru	string	FRU의 FMRI 표현
interface	string	FRU 인터페이스 유형
label	string	FRU 위치 레이블
locate	boolean	플래그의 찾기 표시기
manufacturer	string	FRU 제조업체
model	string	FRU 모델
part	string	FRU 부품 번호
present	boolean	FRU 존재 표시기
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
Accept: application/json
```

응답 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,
    "manufacturer": "HITACHI",
    "model": "H7230AS60SUN3.0T",
    "pathcount": 4,
    "present": true,
    "revision": "A310",
    "rpm": 7200,
    "serial": "001210R322ED      YHJ322ED",
    "size": 3000592982016,
    "type": "data",
    "use": "peer"
  }
}
```

구성요소 등록 정보 수정

PUT 요청은 선택된 하드웨어 구성요소에 대한 등록 정보를 설정하는 데 사용할 수 있습니다. 성공한 요청은 HTTP 상태 201(Accepted)은 물론 JSON 형식의 구성요소 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 30 로그 관리 명령

요청	경로 <i>/api/log/v1</i> 에 추가	설명
GET	<i>/api/log/v1</i> 만 사용	로그 서비스 명령을 나열합니다.
GET	<i>/logs</i>	모든 로그 유형을 나열합니다.
GET	<i>/logs/?start=index/time&limit=entry limit</i>	선택된 범위에 대한 로그 항목을 가져옵니다.
GET	<i>/logs/alert</i>	모든 경고 로그를 나열합니다.
GET	<i>/logs/alert?start=index/time&limit=entry limit</i>	선택된 범위에 대한 로그 항목을 가져옵니다.
GET	<i>/collect</i>	모든 로그 항목의 모음을 다운로드합니다.
GET	<i>/collect?start=index/time&limit=entry limit</i>	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 ab6rt4psMWE=
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” \[73\]](#)에 설명된 `collect` 함수를 사용하십시오.

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

매개변수	설명
<code>limit=number</code>	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 abcd45sMWE=
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 다중 경로 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 지정을 위한 기본 구성체입니다. 각 IP 인터페이스는 단일 데이터 링크와 연관되거나 다른 인터페이스로 구성된 IPMP(IP Multipathing) 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 31 네트워크 데이터 링크 명령

요청	경로 <i>/network/v1</i> 에 추가	설명
POST	<i>/datalinks</i>	새 네트워크 데이터 링크를 만듭니다.
GET	<i>/datalinks/datalink</i>	지정된 네트워크 데이터 링크 등록 정보를 가져옵니다.
GET	<i>/datalinks</i>	모든 네트워크 데이터 링크 객체를 나열합니다.
PUT	<i>/datalinks/datalink</i>	지정된 네트워크 데이터 링크 객체를 수정합니다.
DELETE	<i>/datalinks/datalink</i>	지정된 데이터 링크 객체를 삭제합니다.

표 32 물리적 장치 데이터 링크 등록 정보

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

표 33 VNIC 장치 데이터 링크 등록 정보

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

표 34 VLAN 장치 데이터 링크 등록 정보

등록 정보	유형	설명
<i>class</i>	String	"vlan"("immutable")
<i>label</i>	NetworkLabel	레이블

등록 정보	유형	설명
links	ChooseOne	링크 ["ixgbe0"]
mtu	PositiveInteger	최대 전송 단위(MTU)
id	VLAN	VLAN ID

표 35 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	점보 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 36 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
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
```

```
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"
}
```

응답 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}
```

응답 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
```

응답 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 37 네트워크 장치 명령

요청	경로 /network/v1에 추가	설명
GET	/devices/device	지정된 네트워크 장치 등록 정보를 가져옵니다.
GET	/devices	모든 네트워크 장치 객체를 나열합니다.

표 38 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 ab6rt4psMWE=
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 ab6rt4psMWE=
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 39 네트워크 인터페이스 명령

요청	경로 <i>/api/network/v1</i> 에 추가	설명
POST	<i>/interfaces</i>	새 네트워크 인터페이스를 만듭니다.
GET	<i>/interfaces/interface</i>	지정된 네트워크 인터페이스 등록 정보를 가져옵니다.
GET	<i>/interfaces</i>	모든 네트워크 인터페이스 객체를 나열합니다.
PUT	<i>/interfaces/interface</i>	지정된 네트워크 인터페이스 객체를 수정합니다.
DELETE	<i>/interfaces/interface</i>	지정된 인터페이스 객체를 삭제합니다.

표 40 네트워크 인터페이스 등록 정보

등록 정보	설명
admin	이 인터페이스에 대한 관리가 허용되는지 여부를 나타내는 플래그
class	클래스 유형("ip", "icmp")(만든 후 변경할 수 없음)
curaddrs	현재 IP 주소(변경할 수 없음)
enable	이 인터페이스를 사용으로 설정할지 여부를 나타내는 플래그
label	인터페이스에 대한 사용자 레이블
links	이 인터페이스에 대한 네트워크 링크 선택
state	인터페이스의 상태(변경할 수 없음)
v4addrs	IPv4 주소
v6dhcp	IPv4 DHCP 플래그
v6addrs	IPv6 주소
v6dhcp	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 abcd1234MWE=
Host: zfs-storage.example.com:215
Accept: application/json
```

응답 예:

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

```
{
  "interfaces": [{
    "href": "/api/network/v1/interfaces/ixgbe0",
    "v4addrs": ["ipaddr-1"]
  }, {
    "href": "/api/network/v1/interfaces/ixgbe1",
    "v4addrs": ["ipaddr-2"]
  }, {
    "href": "/api/network/v1/interfaces/ixgbe2",
    "v4addrs": ["ipaddr-3"]
  }, {
    "href": "/api/network/v1/interfaces/ixgbe3",
    "v4addrs": ["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 abcd1234MWE=
Host: zfs-storage.example.com:215
Accept: application/json
```

응답 예:

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

```
{
  "interface": {
    "admin": true,
    "class": "ip",
    "curaddrs": ["ipaddr-1"],
    "enable": true,
    "href": "/api/network/v1/interfaces/ixgbe0",
    "interface": "ixgbe0",
    "label": "Untitled Interface",
    "links": ["ixgbe0"],
    "state": "up",
    "v4addrs": ["ipaddr-1"],
    "v4dhcp": false,
    "v6addrs": [],
    "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"
}
```

응답 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"
}
```

응답 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 abcd1234MWE=
Host: zfs-storage.example.com:215

```

결과 예:

HTTP/1.1 204 No Content

네트워크 경로

해당 명령은 네트워크 경로를 관리합니다.

표 41 네트워크 경로 관리

요청	경로 <i>/api/network/v1</i> 에 추가	설명
POST	<i>/routes</i>	새 네트워크 경로를 만듭니다.
GET	<i>/routes/route</i>	지정된 네트워크 경로 등록 정보를 가져옵니다.
GET	<i>/routes</i>	모든 네트워크 경로 객체를 나열합니다.
DELETE	<i>/routes/route</i>	지정된 경로 객체를 삭제합니다.
GET	<i>/routing</i>	네트 경로 지정 등록 정보를 가져옵니다.
PUT	<i>/routing</i>	네트 경로 지정 등록 정보를 수정합니다.

표 42 네트워크 경로 등록 정보 관리

등록 정보	설명
type	"system" 또는 "static"과 같은 경로의 유형(변경할 수 없음)
family	주소 그룹(IPv4 또는 IPv6)
destination	경로 대상 주소
gateway	게이트웨이 주소
interface	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 abcd1234MWE=
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 abcd1234MWE=
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 abcd1234MWE=
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 abcd1234MWE=
Host: zfs-storage.example.com:215
```

결과 예:

HTTP/1.1 204 No Content

RESTful API 문제 서비스

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

문제 서비스 명령

표 43 문제 서비스 명령

요청	경로 /problem/v1에 추가	설명
GET	/problem/v1만 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.

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
Accept: application/json
```

응답 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
Accept: application/json

```

응답 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
Accept: application/json
```

성공한 응답은 HTTP 상태 202(Accepted)를 반환합니다.

```
HTTP/1.1 202 Accepted
```


RESTful API 역할 서비스

역할은 사용자에게 지정할 수 있는 권한 모음입니다. 관리자 역할과 운영자 역할을 만들고 각각에 서로 다른 권한 부여 레벨을 지정하는 것이 좋습니다. 직원에게는 불필요한 권한을 지정하지 않고 필요에 맞는 역할이 지정될 수 있습니다. 역할을 사용하는 것이 공유 관리자 암호를 사용(예: 모든 사람에게 루트 암호 제공)하는 것보다 훨씬 안전합니다. 역할은 사용자에게 꼭 필요한 권한만 부여하며 감사 로그에서 사용자 작업을 해당 사용자의 사용자 이름과 함께 표시합니다. 기본적으로 매우 기본적인 권한 부여를 포함하는 "기본 관리"라는 역할이 있습니다.

RESTful API 역할 서비스를 사용하여 시스템 역할 및 권한 부여를 관리합니다.

역할 서비스 명령 개요

다음 목록에는 역할 명령이 나와 있습니다.

표 44 역할 서비스 명령

요청	경로 <i>/role/v1</i> 에 추가	설명
GET	<i>/role/v1</i> 만 사용	역할 서비스 명령을 나열합니다.
GET	<i>/roles/role</i>	지정된 관리 역할 등록 정보를 가져옵니다.
GET	<i>/roles</i>	모든 관리 역할 객체를 나열합니다.
PUT	<i>/roles/role</i>	지정된 관리 역할 객체를 수정합니다.
DELETE	<i>/roles/role</i>	지정된 역할 객체를 삭제합니다.
POST	<i>/roles</i>	새 역할을 만들거나 기존 역할을 복제합니다.
PUT	<i>/roles/role/revoke</i>	모든 사용자에서 지정된 역할을 제거합니다.
POST	<i>/roles/role/authorizations</i>	새 역할 권한 부여를 만듭니다.
GET	<i>/roles/role/authorizations/auth</i>	지정된 역할 권한 부여 등록 정보를 가져옵니다.
GET	<i>/roles/role/authorizations</i>	모든 역할 권한 부여 객체를 나열합니다.
PUT	<i>/roles/role/authorizations/auth</i>	지정된 역할 권한 부여 객체를 수정합니다.
DELETE	<i>/roles/role/authorizations/auth</i>	지정된 권한 부여 객체를 삭제합니다.

역할 나열

각 역할에는 다음과 같은 요약 등록 정보가 있습니다. 역할 등록 정보의 전체 설명은 CLI 도움말 말을 참조하십시오.

표 45 역할 등록 정보

등록 정보	유형	설명
name	string	역할 이름(만든 후 변경할 수 없음)
description	string	역할의 설명

요청 예:

```
GET /api/role/v1/roles HTTP/1.1
Authorization: Basic abcefgMWE=
Host: zfs-storage.example.com:215
Accept: application/json
```

응답 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 필의 매개변수를 true로 설정합니다.

요청 예:

```
GET /api/role/v1/roles/basic?props=true HTTP/1.1
Authorization: Basic abcefgMWE=
Host: zfs-storage.example.com:215
Accept: application/json
```

응답 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 46 새 역할 만들기 등록 정보

등록 정보	유형	설명
name	string	새 역할의 이름(필수)
clone	string	원래 등록 정보를 복제할 역할의 이름(선택적)
description	string	역할 설명(필수)

요청 예:

```
POST /api/role/v1/roles HTTP/1.1
Authorization: Basic abcefgMWE=
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 abcefgMWE=
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 abcefgMWE=
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=true를 DELETE 명령에 추가합니다.

요청 예:

```
DELETE /api/role/v1/roles/rola?confirm=true HTTP/1.1
Authorization: Basic abcefgMWE=
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 abcefgMWE=
Host: zfs-storage.example.com:215
Accept: application/json
```

응답 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 abcefgMWE=
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 abcefgMWE=
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 abcefgMWE=  
Host: zfs-storage.example.com:215  
Accept: */*
```

결과 예:

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


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:

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

initiator 개시자의 IQN, WWN 또는 EUI

표 47 개시자 명령

요청	경로 <i>/san/v1.0</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 48 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 auth protocol) 사용자 이름
<i>chapsecret</i>	<i>iscsi</i>	CHAP(Challenge handshake auth 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
Accept: application/json
```

응답 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
Accept: application/json
```

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

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

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

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

응답 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 49 개시자 그룹 명령

요청	경로 / <i>san/v1.0</i> 에 추가	설명
GET	/protocol/initiator-groups	지정된 프로토콜(<i>fc</i> , <i>iscsi</i> , <i>srp</i>) 객체에 대한 모든 SAN 개시자 그룹을 나열합니다.
GET	/protocol/initiator-groups/ <i>name</i>	지정된 프로토콜(<i>fc</i> , <i>iscsi</i> , <i>srp</i>) 등록 정보에 대한 지정된 SAN 개시자 그룹을 가져옵니다.
POST	/protocol/initiator-groups	지정된 프로토콜(<i>fc</i> , <i>iscsi</i> , <i>srp</i>)에 대한 새 SAN 개시자 그룹을 만듭니다.
PUT	/protocol/initiator-groups/ <i>name</i>	지정된 프로토콜(<i>fc</i> , <i>iscsi</i> , <i>srp</i>) 객체에 대한 지정된 SAN 개시자 그룹을 수정합니다.

요청	경로 <code>/san/v1.0</code> 에 추가	설명
DELETE	<code>/protocol/initiator-groups/name</code>	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
Accept: application/json
```

응답 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 세부정보를 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
Accept: application/json
```

응답 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 50 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
Content-Type: application/json
Content-Length: 64
Accept: application/json

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

응답 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 51 대상 명령

요청	경로 <i>/san/v1.0</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 52 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 53 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 abcd123MWE=
Accept: application/json
```

응답 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 abcd123MWE=
Accept: application/json
```

응답 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 abcd123MWE=
Content-Type: application/json
Content-Length: 23
Accept: application/json

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

응답 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 abcd123MWE=
Host: zfs-storage.example.com
Content-Type: application/json
Content-Length: 54
Accept: application/json
```

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

응답 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 abcd123MWE=
```

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 54 대상 그룹 명령

요청	경로 <i>/san/v1.0</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 abcd123MWE=
Accept: application/json
```

응답 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 abcd123MWE=
Accept: application/json
```

응답 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 abcd123MWE
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"]}]}
```

응답 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/nas/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 55 서비스 명령

요청	경로 /service/v1에 추가	설명
GET	/service/v1만 사용	서비스 명령을 나열합니다.
GET	/services	모든 서비스를 나열합니다.
GET	/services/service	지정된 서비스에 대한 구성 및 상태를 가져옵니다.
PUT	/services/service	지정된 서비스에 대한 구성을 수정합니다.
PUT	/services/service/enable	지정된 서비스를 사용으로 설정합니다.
PUT	/services/service/disable	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
Accept: application/json
```

응답 예:

```
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"
}, {
  "<status>": "online",
  "href": "/api/service/v1/services/smb",
  "log": {
    "href": "/api/log/v1/logs/network-smb:default",
    "size": 2
  },
  "name": "smb"
}, {
  "<status>": "online",
  "href": "/api/service/v1/services/dns",
  "log": {
    "href": "/api/log/v1/logs/network-dns-client:default",
    "size": 4
  },
  "name": "dns"
}, {
  "<status>": "online",
  "href": "/api/service/v1/services/dynrouting",
  "log": {
    "href": "/api/log/v1/logs/network-routing-route:default",
    "size": 81
  },
  "name": "dynrouting"
}, {
  "<status>": "disabled",
  "href": "/api/service/v1/services/ftp",
  "log": {
    "href": "/api/log/v1/logs/network-ftp:proftpd",
    "size": 40
  },
  "name": "ftp"
}, {
  "<status>": "disabled",
  "href": "/api/service/v1/services/http",
  "name": "http"
}, {
  "<status>": "online",
  "href": "/api/service/v1/services/identity",
  "log": {
    "href": "/api/log/v1/logs/system-identity:node",
    "size": 4
  },
  "name": "identity"
}, {
  "<status>": "online",
  "href": "/api/service/v1/services/idmap",
  "log": {
    "href": "/api/log/v1/logs/system-idmap:default",
    "size": 15
  },
  "name": "idmap"
}, {
  "<status>": "online",
  "href": "/api/service/v1/services/ipmp",
  "log": {
    "href": "/api/log/v1/logs/network-ipmp:default",
    "size": 3
  },
  "name": "ipmp"
}
```

```

}, {
  "<status>": "online",
  "href": "/api/service/v1/services/iscsi",
  "log": {
    "href": "/api/log/v1/logs/network-iscsi-target:default",
    "size": 3
  },
  "name": "iscsi"
}, {
  "<status>": "disabled",
  "href": "/api/service/v1/services/ldap",
  "name": "ldap"
}, {
  "<status>": "online",
  "href": "/api/service/v1/services/ndmp",
  "log": {
    "href": "/api/log/v1/logs/system-ndmpd:default",
    "size": 11
  },
  "name": "ndmp"
}, {
  "<status>": "online",
  "href": "/api/service/v1/services/nfs",
  "log": {
    "href": "/api/log/v1/logs/appliance-kit-nfsconf:default",
    "size": 6
  },
  "name": "nfs"
}, {
  "<status>": "disabled",
  "href": "/api/service/v1/services/nis",
  "log": {
    "href": "/api/log/v1/logs/network-nis-domain:default",
    "size": 3
  },
  "name": "nis"
}, {
  "<status>": "disabled",
  "href": "/api/service/v1/services/ntp",
  "name": "ntp"
}, {
  "<status>": "online",
  "href": "/api/service/v1/services/replication",
  "name": "replication"
}, {
  "<status>": "online",
  "href": "/api/service/v1/services/rest",
  "log": {
    "href": "/api/log/v1/logs/appliance-kit-akrestd:default",
    "size": 10
  },
  "name": "rest"
}, {
  "<status>": "disabled",
  "href": "/api/service/v1/services/scrk",
  "name": "scrk"
}, {
  "<status>": "disabled",
  "href": "/api/service/v1/services/sftp",
  "name": "sftp"
}, {
  "<status>": "online",
  "href": "/api/service/v1/services/shadow",
  "name": "shadow"
}

```

```

    }, {
      "<status>": "online",
      "href": "/api/service/v1/services/smtp",
      "log": {
        "href": "/api/log/v1/logs/network-smtp:sendmail",
        "size": 6
      },
      "name": "smtp"
    }, {
      "<status>": "disabled",
      "href": "/api/service/v1/services/snmp",
      "name": "snmp"
    }, {
      "<status>": "disabled",
      "href": "/api/service/v1/services/srp",
      "name": "srp"
    }, {
      "<status>": "online",
      "href": "/api/service/v1/services/ssh",
      "log": {
        "href": "/api/log/v1/logs/network-ssh:default",
        "size": 3
      },
      "name": "ssh"
    }, {
      "<status>": "disabled",
      "href": "/api/service/v1/services/syslog",
      "name": "syslog"
    }, {
      "<status>": "online",
      "href": "/api/service/v1/services/tags",
      "name": "tags"
    }, {
      "<status>": "disabled",
      "href": "/api/service/v1/services/tftp",
      "name": "tftp"
    }, {
      "<status>": "disabled",
      "href": "/api/service/v1/services/vscan",
      "log": {
        "href": "/api/log/v1/logs/vscan",
        "size": 0
      },
      "name": "vscan"
    }
  ]
}

```

서비스 가져오기

이 명령은 해당 상태 및 구성을 포함하여 단일 서비스의 세부정보를 가져옵니다.

요청 예:

```

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

```

응답 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
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
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.

요청 예:

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

```
{"port": 218}
```

성공한 응답은 HTTP 상태 202(Accepted)를 반환합니다.

```
HTTP/1.1 202 Accepted
Content-Length: 162
Content-Type: application/json; charset=utf-8
X-Zfssa-Service-API: 1.0

{
  "service": {
    "<status>": "disabled",
    "href": "/api/service/v1/services/sftp",
    "keys": [],
    "listen_port": 218,
    "logging_verbosity": "INFO",
    "root_login": false
  }
}
```

서비스 리소스

일부 서비스에 하위 리소스가 있습니다. 각 서비스 또는 서비스 명령 목록에 대해 반환된 데이터를 확인하여 사용할 수 있는 하위 리소스를 검토합니다.

표 56 서비스 하위 리소스 명령

요청	경로	설명
GET	<code>/services/service/resource</code>	서비스 하위 리소스 나열
PUT	<code>/services/service/resource/href</code>	하위 리소스 수정
POST	<code>/services/service/resource</code>	새 하위 리소스를 만듭니다.
DELETE	<code>/services/service/resource/href</code>	하위 리소스를 삭제합니다.

이러한 명령 각각은 다른 RESTful API 명령과 동일한 패턴을 사용합니다. 여기서 `GET`은 지정된 하위 리소스 유형을 나열하거나 가져오는 데 사용되고, `POST`는 새 하위 리소스 유형을 만드는 데 사용되고, `PUT`은 하위 리소스를 수정하는 데 사용되고, `DELETE`는 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 Storage Appliance에서 NAS는 모든 LUN 및 파일 시스템 공유에서 동일한 데이터 중복 특성을 특성화하는 풀에서 구성됩니다. 이 버전의 NAS API에서 풀 작업은 어플라이언스 스토리지 구성을 얻는 데 사용됩니다.

표 57 스토리지 풀 명령

요청	경로 <code>/api/storage/v1</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 58 스토리지 풀 등록 정보

등록 정보	유형	설명
pool	string	대상 풀 이름
profile	string	데이터 장치 프로파일
state	string	풀 상태(online, offline, exported)
asn	string	풀을 소유하는 어플라이언스의 일련 번호
peer	string	클러스터화된 시스템의 경우 피어 노드의 ASN
owner	string	풀을 소유하는 시스템의 호스트 이름

요청 예:

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

응답 예:

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

{
  "pools": [{
    "profile": "mirror3",
    "name": "m1",
    "peer": "00000000-0000-0000-0000-000000000000",
    "state": "online",
    "owner": "zfs-storage",
    "asn": "2f4aeeb3-b670-ee53-e0a7-d8e0ae410749"
  }, {
    "profile": "raidz1",
    "name": "r1",
    "peer": "00000000-0000-0000-0000-000000000000",
    "state": "online",
    "owner": "zfs-storage",
    "asn": "2f4aeeb3-b670-ee53-e0a7-d8e0ae410749"
  }]
}
```

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
Accept: application/json
```

응답 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
Authorization: Basic abhadbfsMWE=
Content-Type: application/json
Accept: application/json
```

```
{
  "name": "p1",
}
```

응답 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
Authorization: Basic abhadbfsmWE=
Content-Type: application/json
Accept: application/json

```

```

{
  "name": "p1",
  "profile": "stripe",
  "1-data": 8
}

```

응답 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. 스토리지 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
Authorization: Basic abhadbfsMWE=
Content-Type: application/json
Accept: application/json
```

```
{
  "2-data": 8
}
```

응답 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
Authorization: Basic abhadbfsMWE=
Content-Type: application/json
Accept: application/json
{
  "0-cache" : 2
}
```

응답 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 abhadbfsMWE=
Content-Type: application/json
Accept: application/json
```

응답 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` 또는 `DELETE` 요청을 전송하면 각각 풀 스크러빙이 시작되거나 실행 중인 스크러빙 작업이 중지됩니다. 자세한 내용은 CLI 명령 "구성 스토리지 스크러빙"을 참조하십시오.

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
Authorization: Basic abhadbfsMWE=
```

응답 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 `/projects`를 추가하고 단일 프로젝트에서 작동하는 명령은 `/projects/project`를 추가합니다.

표 59 프로젝트 명령

요청	경로 <i>/api/storage/v1</i> 에 추가	설명
GET	<i>/projects</i>	모든 프로젝트를 나열합니다.
GET	<i>/pools/pool/projects</i>	프로젝트를 나열합니다.
GET	<i>/pools/pool/projects?snaps=true</i>	스냅샷을 포함하여 모든 프로젝트를 나열합니다.
GET	<i>/pools/pool/projects/project</i>	프로젝트 세부정보를 가져옵니다.
POST	<i>/pools/pool/projects</i>	프로젝트를 만듭니다.
PUT	<i>/pools/pool/projects/project</i>	프로젝트를 수정합니다.
DELETE	<i>/pools/pool/projects/project</i>	프로젝트를 삭제합니다.
GET	<i>/pools/pool/projects/project/usage/groups</i>	프로젝트 그룹 사용량을 가져옵니다.
GET	<i>/pools/pool/projects/project/usage/groups/group</i>	지정된 그룹에 대한 프로젝트 사용량을 가져옵니다.
GET	<i>/pools/pool/projects/project/usage/users</i>	프로젝트 사용자 사용량을 가져옵니다.
GET	<i>/pools/pool/projects/project/usage/users/user</i>	지정된 사용자에 대한 프로젝트 사용량을 가져옵니다.

다음 표는 프로젝트 리소스 내에 있는 편집 가능한 등록 정보의 목록을 보여 줍니다.

표 60 프로젝트 등록 정보

등록 정보	유형	설명
<code>aclinherit</code>	string	ACL 상속 동작("discard", "noallow", "restricted", "passthrough", "passthrough-x", "passthrough-mode-preserve")
<code>aclmode</code>	string	모드 변경 시 ACL 동작("discard", "mask", "passthrough")
<code>atime</code>	boolean	읽기 시 액세스 시간 업데이트 플래그
<code>canonical_name</code>	string	표준 이름
<code>checksum</code>	string	블록 체크섬("fletcher2", "fletcher4", "sha256")
<code>compression</code>	string	데이터 압축 설정("off", "lzb", "gzip-2", "gzip", "gzip-9")
<code>copies</code>	number	추가 복제 복사본 수
<code>creation</code>	datetime	프로젝트(또는 LUN, 파일 시스템) 생성 날짜 및 시간
<code>dedup</code>	boolean	데이터 중복 플래그
<code>default_group</code>	string	프로젝트 기본 파일 시스템 그룹: "other"
<code>default_permissions</code>	string	프로젝트 기본 파일 시스템 권한 "700"
<code>default_sparse</code>	boolean	프로젝트 기본 LUN 스파스 데이터 플래그
<code>default_user</code>	string	프로젝트 기본 파일 시스템 사용자: "nobody"
<code>default_volblocksize</code>	number	프로젝트 기본 LUN 블록 크기: 8192
<code>default_volsize</code>	number	프로젝트 기본 LUN 크기

등록 정보	유형	설명
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")
sharetftp	string	TFTP 공유("off", "rw", "ro")
snapdir	string	.zfs/스냅샷 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
Accept: application/json
```

get에 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.

backup을 해당 등록 정보의 일부로 사용하여 모든 프로젝트를 가져오는 요청의 예:

```
GET /projects?filter=backup HTTP/1.1
Host: zfs-storage.example.com
Accept: application/json
```

프로젝트 등록 정보 가져오기

이 명령은 지정된 풀의 단일 프로젝트에 대한 등록 정보를 나열합니다. 성공한 get은 JSON 형식의 프로젝트 등록 정보와 함께 HTTP 코드 200(OK)을 반환합니다.

zfs-storage-1 풀에 이름이 proj-01인 프로젝트를 나열하는 요청의 예:

```
GET /api/storage/v1/pools/p1/projects/proj-01 HTTP/1.1
Host: zfs-storage.example.com
Accept: application/json
```

응답 예:

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

{
  "project": {
    "default_volblocksize": 8192.0,
    "logbias": "latency",
    "creation": "20130411T20:02:35",
    "nodeldestroy": 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",
    "sharesftp": "",
    "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
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-01에서 new-name으로 변경하는 요청의 예:

```
POST /api/storage/v1/pools/p1/projects/proj-01 HTTP/1.1
Host: zfs-storage.example.com
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.

응답 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 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 HTTP/1.1
Host: zfs-storage.example.com
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 61 파일 시스템 명령

요청	경로 <i>/api/storage/v1</i> 에 추가	설명
GET	<i>/filesystems</i>	모든 파일 시스템을 나열합니다.

요청	경로 <i>/api/storage/v1</i> 에 추가	설명
GET	<i>/pools/pool/projects/project/filesystems</i>	지정된 파일 시스템을 나열합니다.
GET	<i>/pools/pool/projects/project/filesystems?snaps=true</i>	스냅샷을 포함하여 모든 파일 시스템을 나열합니다.
GET	<i>/pools/pool/projects/project/filesystems/filesystem</i>	파일 시스템 세부정보를 가져옵니다.
POST	<i>/pools/pool/projects/project/filesystems</i>	파일 시스템을 만듭니다.
PUT	<i>/pools/pool/projects/project/filesystems/filesystem</i>	파일 시스템을 수정합니다.
DELETE	<i>/pools/pool/projects/project/filesystems/filesystem</i>	파일 시스템을 삭제합니다.
GET	<i>/pools/pool/projects/project/filesystems/filesystem/usage/groups</i>	파일 시스템 그룹 사용량을 가져옵니다.
GET	<i>/pools/pool/projects/project/filesystems/filesystem/usage/groups/group</i>	지정된 그룹에 대한 파일 시스템 사용량을 가져옵니다.
POST	<i>/pools/pool/projects/project/filesystems/filesystem/usage/groups</i>	파일 시스템 그룹 쿼터를 만듭니다.
PUT	<i>/pools/pool/projects/project/filesystems/filesystem/usage/groups/name</i>	파일 시스템 그룹 쿼터를 수정합니다.
GET	<i>/pools/pool/projects/project/filesystems/filesystem/usage/users</i>	파일 시스템 사용자 사용량을 가져옵니다.
GET	<i>/pools/pool/projects/project/filesystems/filesystem/usage/users/user</i>	지정된 사용자에 대한 파일 시스템 사용량을 가져옵니다.
POST	<i>/pools/pool/projects/project/filesystems/filesystem/usage/users</i>	파일 시스템 사용자 쿼터를 만듭니다.
PUT	<i>/pools/pool/projects/project/filesystems/filesystem/usage/users/name</i>	파일 시스템 사용자 쿼터를 수정합니다.
GET	<i>/pools/pool/projects/project/filesystems/filesystem/shadow/errors</i>	새도우 마이그레이션 오류를 나열합니다.

각 파일 시스템은 프로젝트의 등록 정보를 포함하고 있으며 다음 파일 시스템별 등록 정보도 포함하고 있습니다.

표 62 파일 시스템 등록 정보

등록 정보	유형	설명
<i>casesensitivity</i>	string	대소문자 구분 설정(<i>mixed</i> , <i>sensitive</i> 또는 <i>insensitive</i>)
<i>group</i>	string	그룹 이름
<i>normalization</i>	string	정규화
<i>permissions</i>	string	파일 시스템 권한
<i>project</i>	string	프로젝트 이름
<i>quota_snap</i>	boolean	쿼터에 스냅샷을 포함하는 플래그
<i>reservation_snap</i>	boolean	예약에 스냅샷을 포함하는 플래그
<i>shadow</i>	string	데이터 마이그레이션 소스
<i>errors</i>	string	데이터 마이그레이션 오류
<i>sharesmb_name</i>	string	SMB 공유의 이름
<i>source</i>	object	프로젝트 상속 등록 정보
<i>usage</i>	object	파일 시스템 사용량 정보
<i>user</i>	string	공유를 소유하는 사용자 이름

등록 정보	유형	설명
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
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/v1/filesystems가 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
Accept: application/json
```

파일 시스템 가져오기

파일 시스템 가져오기 명령은 지정된 풀 또는 프로젝트의 단일 파일 시스템 등록 정보를 반환합니다. 다음 URI 매개변수가 사용됩니다.

pool 스토리지 풀 이름

project 프로젝트 이름

filesystem 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
Accept: application/json
```

성공한 `get`은 JSON 형식의 파일 시스템 등록 정보와 함께 HTTP 상태 200(OK)을 반환합니다.

응답 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",
      "sharetftp": "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",
    "sharesftp": "",
    "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.

pool 스토리지 풀 이름

project 프로젝트 이름

filesystem 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
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.

응답 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.

pool 스토리지 풀 이름

project 프로젝트 이름

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

```
{
  "name": "new-name",
  "root_user": "nobody",
}
```

응답 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.

pool 스토리지 풀 이름

project 프로젝트 이름

filesystem 파일 시스템 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 /api/storage/v1/pools/p1/projects/proj-01/filesystems/share-01 HTTP/1.1
Host: zfs-storage.example.com
Accept: application/json
```

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

응답 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 63 볼륨 명령

요청	경로 <code>/api/storage/v1</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 64 볼륨 등록 정보

등록 정보	유형	설명
<code>assignednumber</code>	number 또는 list of numbers	지정된 LU 번호 다중 개시자 그룹에 제공된 경우 유형은 list of numbers입니다. 다중 개시자 그룹에 제공된 경우 <code>assignednumber</code> 및 <code>initiatorgroups</code> 의 순서가 조정됩니다. 예를 들어, <code>assignednumber</code> 목록

등록 정보	유형	설명
		록의 첫번째 항목은 <code>initiatorgroups</code> 목록의 첫번째 항목과 관련이 있습니다.
<code>fixednumber</code>	<code>boolean</code>	LU 번호를 현재 값에 고정하는 플래그.
<code>initiatorgroups</code>	<code>list of strings</code>	개시자 그룹입니다. LUN이 다중 개시자 그룹에 제공된 경우 <code>assignednumber</code> 및 <code>initiatorgroups</code> 의 순서가 조정됩니다. 예를 들어, <code>assignednumber</code> 목록의 첫번째 항목은 <code>initiatorgroups</code> 목록의 첫번째 항목과 관련이 있습니다.
<code>lunguid</code>	<code>string</code>	STMF GUID입니다.
<code>lunumber</code>	<code>number</code> 또는 <code>string</code>	LU 번호 <code>number</code> 또는 <code>auto</code> 입니다.
<code>project</code>	<code>string</code>	프로젝트 이름(변경할 수 없음).
<code>source</code>	<code>object</code>	등록 정보의 소스(<code>local</code> 또는 <code>inherited</code>)를 나열합니다.
<code>sparse</code>	<code>boolean</code>	Thin 프로비저닝을 사용으로 설정하는 플래그.
<code>status</code>	<code>string</code>	논리 장치 상태(<code>online</code> 또는 <code>offline</code>)입니다.
<code>targetgroup</code>	<code>string</code>	대상 그룹
<code>usage</code>	<code>object</code>	LUN 사용량 통계 나열
<code>volblocksize</code>	<code>number</code>	볼륨 블록 크기
<code>volsize</code>	<code>number</code>	볼륨 크기
<code>writocache</code>	<code>boolean</code>	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"]}
```

LUN 나열

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.

pool 스토리지 풀 이름

project 프로젝트 이름

filesystem 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
Accept: application/json
```

성공한 *get*은 JSON 형식의 LUN 등록 정보와 함께 HTTP 상태 200(OK)을 반환합니다.

응답 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.

pool 스토리지 풀 이름

project 프로젝트 이름

lun 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
Accept: application/json
```

성공한 *get*은 JSON 형식의 LUN 등록 정보와 함께 HTTP 상태 200(OK)을 반환합니다.

응답 예:

```
HTTP/1.1 200 OK
Content-Type: application/json
{
  "lun": {
    "logbias": "latency",
    "creation": "20130423T21:31:17",
    "nodeestroy": 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
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
  lunnumber : 'auto',         // Volume LUN 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
Content-Type: application/json
Accept: application/json
```

```
{
  "name": "new-name",
}
```

성공한 응답은 HTTP 상태 202(Accepted)를 반환하고 모든 LUN 등록 정보를 나열합니다.

응답 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
Accept: application/json
```

성공한 *get*은 HTTP 상태 204(No Content)를 반환합니다.

응답 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 URI `/api/storage/v1/pools/pool/projects/project`로 시작합니다.
- 모든 파일 시스템 기반 스냅샷 작업에 대한 URI는 `/api/storage/v1/pools/pool/projects/project/filesystems/filesystem`으로 시작합니다.
- 모든 LUN 기반 스냅샷 작업에 대한 URI는 `/api/storage/v1/pools/pool/projects/project/luns/lun`으로 시작합니다.

표 65 스냅샷 및 복제 명령

요청	경로 <code>/api/storage/v1</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.
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>	레이터를 지정된 파일 시스템 스냅샷으로 롤백합니다.

요청	경로 <i>/api/storage/v1</i> 에 추가	설명
PUT	<i>/pools/pool/projects/project/lun/lun/snapshots/snapshot/rollback</i>	데이터를 지정된 LUN 스냅샷으로 롤백합니다.
DELETE	<i>/pools/pool/projects/project/snapshots/snapshot</i>	프로젝트 스냅샷을 삭제합니다.
DELETE	<i>/pools/pool/projects/project/filesystems/filesystem/snapshots/snapshot</i>	파일 시스템 스냅샷을 삭제합니다.
DELETE	<i>/pools/pool/projects/project/luns/lun/snapshots/snapshot</i>	LUN 스냅샷을 삭제합니다.
GET	<i>/pools/pool/projects/project/snapshots/snapshot/dependents</i>	프로젝트 스냅샷 종속 항목을 나열합니다.
GET	<i>/pools/pool/projects/project/filesystems/filesystem/snapshots/snapshot/dependents</i>	파일 시스템 스냅샷 종속 항목을 나열합니다.
GET	<i>/pools/pool/projects/project/lun/lun/snapshots/snapshot/dependents</i>	LUN 스냅샷 종속 항목을 나열합니다.
POST	<i>/pools/pool/projects/project/automatic</i>	새 프로젝트 자동 스냅샷 객체를 만듭니다.
POST	<i>/pools/pool/projects/project/automatic?convert=true</i>	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	<i>/pools/pool/projects/project/automatic/automatic</i>	지정된 프로젝트 자동 스냅샷 등록 정보를 가져옵니다.
GET	<i>/pools/pool/projects/project/automatic</i>	모든 프로젝트 자동 스냅샷 객체를 나열합니다.
PUT	<i>/pools/pool/projects/project/automatic/automatic</i>	지정된 프로젝트 자동 스냅샷 객체를 수정합니다.
PUT	<i>/pools/pool/projects/project/automatic/automatic?convert=true</i>	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	<i>/pools/pool/projects/project/automatic/automatic</i>	지정된 자동 객체를 삭제합니다.
DELETE	<i>/pools/pool/projects/project/automatic/automatic?convert=true</i>	지정된 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/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.

요청	경로 /api/storage/v1에 추가	설명
		convert 등록 정보를 제외하면 자동 생성된 기존 스냅샷이 삭제됩니다.
GET	/pools/pool/projects/project/filesystems/filesystem/automatic/automatic	지정된 파일 시스템 자동 스냅샷 등록 정보를 가져옵니다.
GET	/pools/pool/projects/project/filesystems/filesystem/automatic	모든 파일 시스템 자동 스냅샷 객체를 나열합니다.
PUT	/pools/pool/projects/project/filesystems/filesystem/automatic/automatic	지정된 파일 시스템 자동 스냅샷 객체를 수정합니다.
PUT	/pools/pool/projects/project/filesystems/filesystem/automatic/automatic?convert=true	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	/pools/pool/projects/project/filesystems/filesystem/automatic/automatic	지정된 자동 스냅샷 일정 객체를 삭제합니다.
DELETE	/pools/pool/projects/project/filesystems/filesystem/automatic/automatic?convert=true	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	/pools/pool/projects/project/luns/lun/automatic	새 LUN 자동 스냅샷을 만듭니다.
POST	/pools/pool/projects/project/luns/lun/automatic?convert=true	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	/pools/pool/projects/project/luns/lun/automatic/automatic	지정된 LUN 자동 스냅샷 등록 정보를 가져옵니다.
GET	/pools/pool/projects/project/luns/lun/automatic	모든 LUN 자동 스냅샷 객체를 나열합니다.
PUT	/pools/pool/projects/project/luns/lun/automatic/automatic	지정된 LUN 자동 스냅샷 객체를 수정합니다.
PUT	/pools/pool/projects/project/luns/lun/automatic/automatic?convert=true	지정된 LUN 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	/pools/pool/projects/project/luns/lun/automatic/automatic	지정된 LUN 자동 객체를 삭제합니다.
DELETE	/pools/pool/projects/project/luns/lun/automatic/automatic?convert=true	지정된 LUN 자동 스냅샷 일정 객체를 삭제합니다. 새 일정에 맞지 않는 자동 생성된 기존 스냅샷이 수동 스냅샷으로 변환됩니다.

요청	경로 <code>/api/storage/v1</code> 에 추가	설명
		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 66 스냅샷 나열 명령 형식

명령	경로 <code>/api/storage/v1/pools/pool/projects/project</code> 에 추가
프로젝트 스냅샷 나열	<code>/snapshots</code>
List filesystem snapshots	<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
```

응답 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
```

응답 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"}
```

응답 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"}

```

응답 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.

<i>pool</i>	소스 풀 이름
<i>project</i>	소스 프로젝트 이름
<i>filesystem</i>	파일 시스템 스냅샷에 대한 소스 공유 이름
<i>lun</i>	LUN 스냅샷에 대한 소스 공유 이름
<i>snapshot</i>	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 67 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"}
```

응답 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
```

응답 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/projectsnapshots
```

스냅샷에 NDMP 보류가 있는 경우 `?confirm=true`를 DELETE 명령에 추가합니다. 하지만 이렇게 하면 NDMP 작업에 부정적인 영향이 미칠 수 있습니다. 자세한 내용은 [Oracle ZFS Storage Appliance 관리 설명서, 릴리스 OS8.8.0의 "NDMP 구성"](#)을 참조하십시오.

요청 예:

```
DELETE /pools/p1/projects/default/filesystems/fs-01/snapshots/initial-backup?confirm=true
```

`?confirm=true`가 추가되지 않은 경우 결과 예:

스냅샷에 NDMP 보류가 있을 때 `?confirm=true`가 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\nneeded for scripted command (scripted commands must be prefixed with \"confirm\"\nto automatically confirm or \"deny\" to automatically deny) (encountered while\nattempting to run command \"confirm destroy snap\"))", "code": 409, "name":\n"ERR_CONFIRM_REQUIRED"}}
```

스냅샷 종속 항목 나열

파일 시스템 또는 볼륨에 대한 종속 항목을 나열합니다. 다음 URI 매개변수가 사용됩니다.

<i>pool</i>	시스템 스토리지 풀 이름
<i>project</i>	프로젝트 이름
<i>filesystem</i>	파일 시스템 이름

lun LUN 이름

snapshot 스냅샷 이름

파일 시스템 종속 항목 나열:

```
GET /api/storage/v1/
pools/pool/projects/project/filesystems/filesystem/snapshots/snapshot/dependents
```

볼륨 종속 항목 나열:

```
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
```

응답 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

{
  "dependents": [
    {
      "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 68 스키마 명령

요청	경로 <i>/api/storage/v1</i> 에 추가	설명
GET	<i>/schema</i>	모든 NAS 스키마 등록 정보 객체를 나열합니다.
GET	<i>/schema/property</i>	지정된 NAS 스키마 등록 정보 등록 정보를 가져옵니다.

요청	경로 <code>/api/storage/v1</code> 에 추가	설명
POST	<code>/schema</code>	새 NAS 스키마 등록 정보를 만듭니다.
PUT	<code>/schema/property</code>	지정된 NAS 스키마 등록 정보 객체를 수정합니다.
DELETE	<code>/schema/property</code>	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 69 스키마 매개변수

매개변수	설명
<code>property</code>	등록 정보의 이름(변경할 수 없음)
<code>description</code>	등록 정보 설명(브라우저 인터페이스용)
<code>type</code>	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 Storage Appliance의 특정 모델에 대해 사용이 허가된 기능이며 복제 RESTful API가 해당 기능을 관리합니다. 서비스는 URI <https://hostname:215/api/storage/v1/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 70 복제 서비스 명령

요청	경로 /api/service/v1/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 가져오기

복제 서비스의 상태를 가져옵니다.

요청 예:

```
GET /api/service/v1/services/replication HTTP/1.1
Host: zfs-storage.example.com:215
Authorization: Basic ab6rt4psMWE=
Accept: application/json
```

결과 예:

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

{
  "service": {
    "status": "online",
    "href": "/service/v1/services/replication",
    "sources": [],
    "targets": []
  }
}
```

복제 서비스 상태 수정

다른 서비스와 마찬가지로 복제 서비스 상태도 수정할 수 있습니다. 자세한 내용은 서비스 RESTful API를 참조하십시오.

복제 대상

다음 표는 사용 가능한 복제 대상 명령을 보여 줍니다.

표 71 복제 대상 명령

요청	경로 <code>/api/storage/v1</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.
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.

복제 대상 나열

시스템에서 사용할 수 있는 모든 복제 대상을 나열합니다.

요청 예:

```
GET /api/storage/v1/replication/targets HTTP/1.1
Host: zfs-storage.example.com:215
Authorization: Basic ab6rt4psMWE=
Accept: application/json
```

응답 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/v1/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/v1/replication/targets/zfs-storage-2"
  }]
}
```

```
}
```

복제 대상 가져오기

이 명령은 단일 복제 대상의 세부정보를 나열합니다. 대상은 해당 호스트 이름으로 액세스합니다.

요청 예:

```
GET /api/storage/v1/replication/targets/zfs-storage-1 HTTP/1.1
Authorization: Basic ab6rt4psMWE=
Host: zfs-storage.example.com:215
Accept: application/json
```

응답 예:

```
HTTP/1.1 200 OK
X-Zfssa-Replication-API: 1.0
Content-Type: application/json
Content-Length: 337

{
  "target": {
    "href": "/api/storage/v1/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

원격 복제에 대한 새 복제 대상을 만듭니다.

요청 예:

```
POST /api/replication/v1/targets HTTP/1.1
Authorization: Basic ab6rt4psMWE=
Host: zfs-storage.example.com:215
Accept: application/json
Content-Type: application/json
Content-Length: 54

{"hostname":"example", "root_password":"root-password", "label":"zfs-storage-3"}
```

응답 예:

```
HTTP/1.1 201 Created
Content-Length: 135
Content-Type: application/json
Location: /service/v1/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",
    "hostname": "example",
    "href": "/service/v1/services/replication/targets/target-000",
    "label": "zfs-storage-3"
  }
}
```

복제 대상 삭제

이 명령은 기존 복제 대상을 삭제합니다.

요청 예:

```
DELETE /service/v1/services/replication/targets/target-000 HTTP/1.1
Host: zfs-storage.example.com
Authorization: Basic ab6rt4psMWE=
```

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

응답 예:

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

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 72 기본 작업 인터페이스

요청	경로 <code>/api/storage/v1</code> 에 추가	설명
GET	<code>/replication/actions</code>	모든 복제 작업 객체를 나열합니다.

요청	경로 <code>/api/storage/v1</code> 에 추가	설명
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 73 작업 일정 액세스

요청	경로 <code>/api/storage/v1</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 74 복제 자동 스냅샷 구성 액세스

요청	경로 <code>/api/storage/v1</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 75 복제 작업 통계 액세스

요청	경로 <i>/api/storage/v1</i> 에 추가	설명
GET	<i>/replication/actions/ra_id/stats</i>	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/v1/pools/pool/projects/project

- 파일 시스템 기반 작업 URI는 다음으로 시작합니다.

/api/storage/v1/pools/pool/projects/project/filesystems/filesystem

- LUN 기반 작업 URI는 다음으로 시작합니다.

/api/storage/v1/pools/pool/projects/project/luns/lun

복제 작업을 관리하려면 위에 나열된 원하는 컨텍스트 URI에 다음 기본 명령을 추가하십시오.

표 76 프로젝트, 파일 시스템 또는 LUN 기본 복제 작업 인터페이스

요청	위에 나열된 프로젝트, 파일 시스템 또는 LUN URI에 추가	설명
GET	<i>/replication/actions</i>	모든 복제 작업 객체를 나열합니다.
GET	<i>/replication/actions/ra_id</i>	지정된 복제 작업 등록 정보를 가져옵니다.
POST	<i>/replication/actions</i>	새 복제 작업 만들기
PUT	<i>/replication/actions/ra_id</i>	지정된 복제 작업 객체를 수정합니다.
DELETE	<i>/replication/actions/ra_id</i>	지정된 복제 작업 객체를 삭제합니다.
PUT	<i>/replication/actions/ra_id/sendupdate</i>	선택된 복제 작업을 시작합니다.
PUT	<i>/replication/actions/ra_id/cancelupdate</i>	선택된 복제 작업을 중지합니다.

다음 표에는 복제 일정을 관리하는 기본 명령이 나와 있습니다.

- 프로젝트 기반 작업 URI는 다음으로 시작합니다.

/api/storage/v1/pools/pool/projects/project

- 파일 시스템 기반 작업 URI는 다음으로 시작합니다.

/api/storage/v1/pools/pool/projects/project/filesystems/filesystem

- LUN 기반 작업 URI는 다음으로 시작합니다.

/api/storage/v1/pools/pool/projects/project/luns/lun

복제 일정을 관리하려면 위에 나열된 원하는 컨텍스트 URI에 다음 기본 명령을 추가하십시오.

표 77 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.
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/v1/pools/pool/projects/project

- 파일 시스템 기반 작업 URI는 다음으로 시작합니다.

/api/storage/v1/pools/pool/projects/project/filesystems/filesystem

- LUN 기반 작업 URI는 다음으로 시작합니다.

/api/storage/v1/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 78 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.

요청	위에 나열된 프로젝트, 파일 시스템 또는 LUN URI에 추가	설명
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/v1/pools/pool/projects/project
```

- 파일 시스템 기반 작업 URI는 다음으로 시작합니다.

```
/api/storage/v1/pools/pool/projects/project/filesystems/filesystem
```

- LUN 기반 작업 URI는 다음으로 시작합니다.

```
/api/storage/v1/pools/pool/projects/project/luns/lun
```

복제 작업 통계에 액세스하려면 위에 나열된 원하는 컨텍스트 URI에 다음 기본 명령을 추가하십시오.

표 79 복제 작업 통계 액세스

요청	위에 나열된 프로젝트, 파일 시스템 또는 LUN URI에 추가	설명
GET	/replication/actions/ra_id/stats	선택한 복제 작업에 대한 읽기 전용 복제 통계를 검색합니다.

복제 작업 나열

사용 가능한 복제 작업 목록을 가져옵니다.

요청 예:

```
GET /api/storage/v1/replication/actions HTTP/1.1
Authorization: Basic ab6rt4psMWE=
Accept: application/json
```

응답 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/v1/replication/actions/1438ed7f-aad3-c631-d869-9e85cd7f15b4 HTTP/1.1
Authorization: Basic ab6rt4psMWE=
Accept: application/json
```

응답 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/v1/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/v1/replication/actions HTTP/1.1
Authorization: Basic ab6rt4psMWE=
Content-Type: application/json
```

응답 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 정보 만들기:

```
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/v1/replication/actions HTTP/1.1
Host: zfs-storage.example.com:215
Authorization: Basic ab6rt4psMWE=
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"
}
```

응답 예:

```
HTTP/1.1 201 Created
```

```
Content-Length: 506
Content-Type: application/json
Location: /api/storage/v1/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/v1/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/v1/replication/actions/b77bd8cd-17ed-69da-9e4b-aebe3cc63755/schedules
HTTP/1.1
Host: zfs-storage.example.com:215
Authorization: Basic cm9vdDpsMWE=
Accept: */*
Content-Type: application/json
Content-Length: 65

{"frequency": "month", "day": "5th", "hour": "auto", "minute": "auto"}
```

응답 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/v1/replication/actions/c141d88d-ffd2-6730-d489-b71905f340cc HTTP/1.1
Host: zfs-storage.example.com:215
Authorization: Basic ab6rt4psMWE=
Content-Type: application/json
```

```
{"use_ssl": false}
```

응답 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/v1/replication/actions/action_id -d '{"recovery_point_objective":
60}' HTTP/1.1
Host: zfs-storage.example.com:215
Authorization: Basic ab6rt4psMWE=
Content-Type: application/json
```

응답 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/v1/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/v1/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/v1/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/v1/replication/actions/1438ed7f-aad3-c631-d869-9e85cd7f15b4 HTTP/1.1
Authorization: Basic ab6rt4psMWE=
Accept: application/json
```

응답 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/v1/replication/actions/c141d88d-ffd2-6730-d489-b71905f340cc/cancelupdate
HTTP/1.1
Host: zfs-storage.example.com
Authorization: Basic ab6rt4psMWE=
```

응답 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/v1/replication/actions/c141d88d-ffd2-6730-d489-b71905f340cc/sendupdate
HTTP/1.1
Authorization: Basic ab6rt4psMWE=
```

응답 예:

```
HTTP/1.1 202 Accepted
X-Zfssa-Replication-API: 1.0
```

복제 작업 삭제

기존 복제 작업을 삭제합니다.

요청 예:

```
DELETE /api/storage/v1/replication/actions/e7e688b1-ff07-474f-d5cd-cac08293506e HTTP/1.1
Host: zfs-storage.example.com
Authorization: Basic ab6rt4psMWE=
```

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

응답 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 80 복제 패키지 명령

요청	경로 <code>/api/storage/v1/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.

복제 패키지

요청	경로 <code>/api/storage/v1/replication</code> 에 추가	설명
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.
GET	<code>/packages/package/projects/project/filesystems/filesystem/automatic</code>	모든 패키지 파일 시스템 자동 스냅샷 객체를 나열합니다.
GET	<code>/packages/package/projects/project/filesystems/filesystem/automatic/automatic</code>	지정된 패키지 파일 시스템 자동 스냅샷 등록 정보를 가져옵니다.
GET	<code>/packages/package/projects/project/luns</code>	패키지 LUN을 나열합니다.
GET	<code>/packages/package/projects/project/luns/lun</code>	패키지 LUN을 가져옵니다.
PUT	<code>/packages/package/projects/project/luns/lun</code>	패키지 LUN을 수정합니다.
GET	<code>/packages/package/projects/project/luns/lun/usage/groups</code>	패키지 LUN 그룹 사용량을 가져옵니다.
GET	<code>/packages/package/projects/project/luns/lun/usage/users</code>	패키지 LUN 사용자 사용량을 가져옵니다.
GET	<code>/packages/package/projects/project/luns/lun/snapshots/snapshot</code>	지정된 스냅샷 등록 정보를 가져옵니다.
GET	<code>/packages/package/projects/project/luns/lun/snapshots</code>	모든 스냅샷 객체를 나열합니다.
DELETE	<code>/packages/package/projects/project/luns/lun/snapshots/snapshot</code>	지정된 스냅샷 객체를 삭제합니다.
PUT	<code>/packages/package/projects/project/luns/lun/snapshots/snapshot</code>	패키지 LUN 스냅샷의 이름을 바꿉니다.
GET	<code>/packages/package/projects/project/luns/lun/automatic</code>	모든 패키지 LUN 자동 스냅샷 객체를 나열합니다.
GET	<code>/packages/package/projects/project/luns/lun/automatic/automatic</code>	지정된 패키지 LUN 자동 스냅샷 등록 정보를 가져옵니다.

다음 명령을 사용하여 복제 소스 및 해당하는 패키지에도 액세스할 수 있습니다.

표 81 복제 소스 명령

요청	경로 <code>/api/storage/v1/replication/sources</code> 에 추가	설명
GET	<code>/api/storage/v1/replication/sources</code> 만 사용	복제 소스를 나열합니다.
GET	<code>/source</code>	복제 소스 세부정보를 나열합니다.
GET	<code>/source/packages/package</code>	지정된 복제 패키지를 가져옵니다.
PUT	<code>/source/packages/package</code>	지정된 복제 패키지를 수정합니다.
DELETE	<code>/source/packages/package</code>	지정된 복제 패키지를 삭제합니다.
PUT	<code>/source/packages/package/cancelupdate</code>	지정된 패키지에 대한 <code>cancelupdate</code> 를 실행합니다.

요청	경로 <i>/api/storage/v1/replication/sources</i> 에 추가	설명
PUT	<i>/source/packages/package/sever</i>	지정된 패키지에 대한 sever를 실행합니다.
PUT	<i>/source/packages/package/pkgreverse</i>	지정된 패키지에 대한 reverse를 실행합니다.
PUT	<i>/source/packages/package/clone</i>	지정된 패키지를 복제합니다.
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.

요청	경로 <code>/api/storage/v1/replication/sources</code> 에 추가	설명
GET	<code>/source/packages/package/projects/project/filesystems/filesystem/automatic</code>	모든 패키지 파일 시스템 자동 스냅샷 객체를 나열합니다.
GET	<code>/source/packages/package/projects/project/filesystems/filesystem/automatic/automatic</code>	지정된 패키지 파일 시스템 자동 스냅샷 등록 정보를 가져옵니다.
GET	<code>/source/packages/package/projects/project/luns</code>	패키지 LUN을 나열합니다.
GET	<code>/source/packages/package/projects/project/luns/lun</code>	패키지 LUN을 가져옵니다.
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/v1/replication/sources HTTP/1.1
Host: zfs-storage.example.com
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/v1/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/v1/replication/packages HTTP/1.1
Host: zfs-storage.example.com
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/v1/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": "ipaddr-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/v1/replication/packages/8373d331-de60-e590-90e8-9ad69fcb4aec HTTP/1.1
Host: zfs-storage.example.com
Authorization: Basic ab6rt4psMWE=
Content-Type: application/json
```

```
{"enabled": false}
```

결과 예:

```
HTTP/1.1 202 Accepted
X-Zfssa-Replication-API: 1.0
```

요청 예:

```
PUT /api/storage/v1/replication/packages/8373d331-de60-e590-90e8-9ad69fcb4aec/pkgreverse
HTTP/1.1
Host: zfs-storage.example.com
Authorization: Basic ab6rt4psMWE=
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/v1/replication/packages/8373d331-de60-e590-90e8-9ad69fcb4aec HTTP/1.1
Host: zfs-storage.example.com
Authorization: Basic ab6rt4psMWE=
```

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

응답 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/v1/replication/packages/8373d331-de60-e590-90e8-9ad69fcb4aec/cancelupdate
HTTP/1.1
```

```
Host: zfs-storage.example.com
Authorization: Basic ab6rt4psMWE=
```

진행 중인 업데이트가 없는 경우 HTTP 상태 409(Conflict)가 반환됩니다.

응답 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"
  }
}
```

응답 예:

```
HTTP/1.1 202 Accepted
X-Zfssa-Replication-API: 1.0
```

패키지 복제

패키지 프로젝트를 복제합니다.

요청 예:

```
PUT /api/v1/storage/replication/packages/8373d331-de60-e590-90e8-9ad69fcb4aec/clone
HTTP/1.1
Host: zfs-storage.example.com
Authorization: Basic ab6rt4psMWE=
```

응답 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/v1/replication/packages/8373d331-de60-e590-90e8-9ad69fcb4aec/clone/
conflicts HTTP/1.1
Host: zfs-storage.example.com
Authorization: Basic ab6rt4psMWE=
```

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/v1/replication/packages/8373d331-de60-e590-90e8-9ad69fcb4aec/sever
HTTP/1.1
Host: zfs-storage.example.com
Authorization: Basic ab6rt4psMWE=

{"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/v1/replication/packages/8373d331-de60-e590-90e8-9ad69fcb4aec/reverse
HTTP/1.1
Host: zfs-storage.example.com
Authorization: Basic ab6rt4psMWE=

{"projname":"restrev"}
```

성공 응답:

HTTP/1.1 202 Accepted
X-Zfssa-Replication-API: 1.0

암호화

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 Storage Appliance는 개별 공유(파일 시스템 및 LUN) 레벨 및 프로젝트에서 투명한 데이터 암호화를 제공합니다. 어플라이언스는 내장 로컬 키 저장소를 포함하고 있으며 OKM(Oracle Key Manager) 시스템에 연결할 수도 있습니다. 암호화된 각 프로젝트 또는 공유에는 로컬 또는 OKM 키 저장소의 래핑 키가 필요합니다. 데이터 암호화 키는 스토리지 어플라이언스에 의해 관리되며 로컬 또는 OKM 키 저장소의 래핑 키에 의해 암호화되어 영구적으로 저장됩니다.

다음 표에서는 로컬 및 OKM 암호화를 관리하는 데 사용할 수 있는 RESTful API 요청에 대해 설명합니다.

표 82 로컬 암호화

요청	경로 <code>/api/storage/v1</code> 에 추가	설명
GET	<code>/encryption/local</code>	로컬 키 저장소 등록 정보를 가져옵니다.
PUT	<code>/encryption/local</code>	로컬 키 저장소 등록 정보를 수정합니다.
GET	<code>/encryption/local/keys</code>	로컬 키를 가져옵니다.
GET	<code>/encryption/local/keys/key</code>	로컬 키 세부정보를 가져옵니다.
POST	<code>/encryption/local/keys</code>	로컬 키를 만듭니다.
DELETE	<code>/encryption/local/keys/key</code>	로컬 키를 삭제합니다.
GET	<code>/encryption/local/keys/key/dependents</code>	이 키에 종속된 공유를 나열합니다.

표 83 OKM 암호화

요청	경로 <code>/api/storage/v1</code> 에 추가	설명
GET	<code>/encryption/okm</code>	OKM 키 저장소 등록 정보를 가져옵니다.
PUT	<code>/encryption/okm</code>	OKM 키 저장소 등록 정보를 수정합니다.
GET	<code>/encryption/okm/keys</code>	OKM 키를 가져옵니다.
GET	<code>/encryption/okm/keys/key</code>	OKM 키 세부정보를 가져옵니다.
POST	<code>/encryption/okm/keys</code>	OKM 키를 만듭니다.
DELETE	<code>/encryption/okm/keys/key</code>	OKM 키를 삭제합니다.
GET	<code>/encryption/okm/keys/key/dependents</code>	이 키에 종속된 공유를 나열합니다.

모든 로컬 키 나열

출력:

```
{
  "keys": [{
    "cipher": "AES",
    "keyname": "key-1",
    "href": "/api/storage/v1/encryption/local/keys/key-000"
  }, {
    "cipher": "AES",
    "keyname": "key-2",
    "href": "/api/storage/v1/encryption/local/keys/key-001"
  }, {
    "cipher": "AES",
    "keyname": "key-3",
    "href": "/api/storage/v1/encryption/local/keys/key-002"
  }]
}
```

로컬 키 나열

출력:

```
{
  "key": {
    "href": "/api/storage/v1/encryption/local/keys/key-000",
    "cipher": "AES",
    "keyname": "key-1"
  }
}
```

모든 OKM 키 나열

출력:

```
{
  "keys": [{
    "cipher": "AES",
    "keyname": "okm-key-1",
    "href": "/api/storage/v1/encryption/local/keys/key-000"
  }, {
    "cipher": "AES",
    "keyname": "okm-key-2",
    "href": "/api/storage/v1/encryption/local/keys/key-001"
  }, {
    "cipher": "AES",
    "keyname": "okm-key-3",
    "href": "/api/storage/v1/encryption/local/keys/key-002"
  }]
}
```

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 84 어플라이언스 시스템 명령

요청	경로 <code>/api/system/v1</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
Accept: application/json
```

응답 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 한 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
```

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
```

요청 예:

```
PUT /api/system/v1/reboot?skip_update=true HTTP/1.1
Host: zfs-storage.example.com
```

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
```

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
```

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
```

시스템 지원 번들

다음 지원 번들 명령을 사용할 수 있습니다.

표 85 지원 번들 명령

요청	경로 <code>/api/system/v1</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 abhdbfsmWE=
Host: zfs-storage.example.com:215
Accept: application/json
Content-Type: application/json
Content-Length: 23

{"srn": "3-0123456789"}
```

응답 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 abhadbfsmWE=
Host: zfs-storage.example.com:215
Accept: application/json
Content-Type: application/json
Content-Length: 23
```

응답 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 abhadbfsmWE=
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 abhadbfsmWE=
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 abhadbfsmWE=
Host: zfs-storage.example.com:215
```

응답 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 abhadbfsMWE=
Host: zfs-storage.example.com:215
```

응답 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 abhadbfsMWE=
Host: zfs-storage.example.com:215
```

```
{"srn": "3-0123456789"}
```

응답 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 abhadbfsMWE=
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 abhadbfsMWE=
Host: zfs-storage.example.com:215
```

응답 예:

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

시스템 업데이트

이러한 명령은 시스템 업데이트 이미지를 관리합니다.

표 86 업데이트 명령

요청	경로 <i>/api/system/v1</i> 에 추가	설명
GET	<i>/updates</i>	모든 시스템 업데이트를 나열합니다.
GET	<i>/updates/update</i>	지정된 시스템 업데이트 등록 정보를 가져옵니다.
GET	<i>/update/platform</i>	플랫폼 펌웨어(컨트롤러의 서비스 프로세서 및 시스템 보드 펌웨어를 칭함)의 업데이트 상태를 표시합니다.
GET	<i>/update/firmware</i>	구성요소 펌웨어(디스크 및 SSD 펌웨어와 디스크 인클로저 IOM 펌웨어를 칭함)의 업데이트 상태를 표시합니다.
PUT	<i>/updates/update</i>	업데이트 설정 수정
PUT	<i>/updates/update/upgrade</i>	지정된 업데이트 이미지로 업그레이드합니다.
PUT	<i>/updates/update/check</i>	지정된 업데이트 이미지에 대한 업그레이드 상태 검사를 실행합니다.

요청	경로 <code>/api/system/v1</code> 에 추가	설명
PUT	<code>/updates/update/rollback</code>	지정된 업데이트 이미지로 롤백합니다.
PUT	<code>/updates-apply</code>	호환되지 않는 지연 업데이트를 적용합니다.
DELETE	<code>/updates/update</code>	지정된 시스템 업데이트를 삭제합니다.
POST	<code>/updates</code>	업데이트 이미지를 어플라이언스로 로드합니다.

표 87 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 abcdefgMWE=
Host: zfs-storage.example.com:215
Accept: application/json
```

응답 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 abcefgMWE=
Host: zfs-storage.example.com:215
Accept: application/json
```

응답 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 abcefgMWE=
Host: zfs-storage.example.com:215
Accept: application/json
```

응답 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 abcefgMWE=
Host: zfs-storage.example.com:215
Accept: application/json
```

응답 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 abcefgMWE=
Content-Length: 0
```

응답 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 abcefgMWE=
Content-Length: 0
```

응답 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 abcefgMWE=
```

응답 예:

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


RESTful API 사용자 서비스

RESTful API 사용자 서비스는 어플라이언스에서 로컬 관리 사용자 및 사용자 환경 설정을 구성하는 데 사용됩니다.

사용자 서비스 명령

다음 사용자 서비스 명령을 사용할 수 있습니다.

표 88 사용자 서비스 명령

요청	경로 <i>/api/user/v1</i> 에 추가	설명
GET	<i>/api/user/v1</i> 만 사용	사용자 서비스 명령을 나열합니다.
GET	<i>/users</i>	모든 사용자에 대한 요약 정보를 나열합니다.
GET	<i>/users/user</i>	특정 사용자에 대한 세부정보를 가져옵니다.
DELETE	<i>/users/user</i>	시스템에서 로컬 사용자를 제거합니다.
POST	<i>/users</i>	새 로컬 사용자를 만들거나, 기존 사용자를 새 사용자로 복제하거나, 네트워크 디렉토리에서 관리자를 추가합니다.
PUT	<i>/users/user</i>	사용자 등록 정보를 수정합니다.
PUT	<i>/users/user/preferences</i>	사용자 환경 설정을 수정합니다.
GET	<i>/users/user/preferences</i>	사용자 환경 설정을 가져옵니다.
POST	<i>/users/user/exceptions</i>	새 사용자 권한 부여 예외를 만듭니다.
GET	<i>/users/user/exceptions/auth</i>	지정된 사용자 권한 부여 예외 등록 정보를 가져옵니다.
GET	<i>/users/user/exceptions</i>	모든 사용자 권한 부여 예외 객체를 나열합니다.
PUT	<i>/users/user/exceptions/auth</i>	지정된 사용자 권한 부여 예외 객체를 수정합니다.
DELETE	<i>/users/user/exceptions/auth</i>	지정된 권한 부여 객체를 삭제합니다.
POST	<i>/users/user/preferences/keys</i>	새 사용자 ssh 키를 만듭니다.
GET	<i>/users/user/preferences/keys/key</i>	지정된 사용자 ssh 키 등록 정보를 가져옵니다.
GET	<i>/users/user/preferences/keys</i>	모든 사용자 ssh 키 객체를 나열합니다.
PUT	<i>/users/user/preferences/keys/key</i>	지정된 사용자에 대한 지정된 ssh 키를 수정합니다.
DELETE	<i>/users/user/preferences/keys/key</i>	지정된 키 객체를 삭제합니다.

사용자 나열

각 사용자에는 다음과 같은 사용 가능한 요약 등록 정보가 있습니다.

표 89 사용자 등록 정보

등록 정보	유형	설명
logname	string	사용자 이름(만든 후 변경할 수 없음)
uid	number	사용자 ID, 디렉토리 사용자에게 대해 사용으로 설정되지 않음
fullname	string	전체 이름
initial_password	string	암호
require_annotation	boolean	세션 주석 필요 플래그
roles	string	이 사용자의 역할
kiosk_mode	boolean	키오스크 사용자
kiosk_screen	string	키오스크 화면

요청 예:

```
GET /api/user/v1/users HTTP/1.1
Authorization: Basic abcdefgMWE=
Host: zfs-storage.example.com:215
Accept: application/json
```

응답 예:

```
{
  "user": {
    {
      "href": "/api/user/v1/users/admin3",
      "logname": "admin3",
      "type": "local",
      "uid": 2000000000,
      "fullname": "Administrator",
      "initial_password": "password",
      "require_annotation": false,
      "roles": [
        "basic"
      ],
      "kiosk_mode": false,
      "kiosk_screen": "status/dashboard",
      "exceptions": [
      ],
      "preferences": {
        "href": "/api/user/v1/users/admin3/preferences",
        "locale": "C",
        "login_screen": "status/dashboard",
        "session_timeout": 15,
        "advanced_analytics": false,
        "keys": [
        ]
      }
    }
  }
}
```

}

사용자 가져오기

사용자에 대한 세부정보를 가져오고 사용자 환경 설정 및 권한 부여 예외를 포함합니다. 각 권한 부여 예외 유형은 자체 등록 정보를 정의합니다. 사용자 환경 설정 등록 정보가 표시됩니다.

표 90 사용자 환경 설정

등록 정보	유형	설명
locale	string	지역성
login_screen	string	초기 로그인 화면
session_timeout	string	세션 시간 초과(분)
advanced_analytics	string	고급 Analytics 통계 사용

각 사용자는 정의된 환경 설정의 일부로 지정된 ssh 키를 가질 수 있습니다.

표 91 SSH 키 등록 정보

등록 정보	유형	설명
type	string	SSH 키 유형: RSA 또는 DSA
key	string	SSH 키 내용
comment	string	이 SSH 키와 연관된 설명

요청 예:

```
GET /api/user/v1/users/admin1 HTTP/1.1
Authorization: Basic abcefgMWE=
Host: zfs-storage.example.com:215
Accept: application/json
```

응답 예:

```
HTTP/1.1 200 OK
X-Zfssa-Appliance-API: 1.0
Content-Type: application/json
Content-Length: 390

{
  "user": {
    "fullname": "Administrator",
    "href": "/api/user/v1/users/admin3",
    "initial_password": "password",
    "kiosk_mode": false,
    "kiosk_screen": "status/dashboard",
    "logname": "admin3",
    "require_annotation": false,
    "roles": ["basic"]
  }
}
```

```
}

```

사용자 만들기

사용자 및 사용자 유형에 대한 자세한 내용은 [Oracle ZFS Storage Appliance 관리 설명서, 릴리스 OS8.8.0의 "사용자 및 역할 이해"](#)를 참조하십시오.

이 명령은 다음 3가지 형식을 사용합니다.

- 새 사용자 만들기 – 새 사용자를 만듭니다.
- 기존 사용자 복제 – 기존 사용자에서 새 사용자를 복제합니다.
- 관리자 추가 – 엔터프라이즈 디렉토리에서 정의된 사용자가 어플라이언스를 관리할 수 있도록 허용합니다.

이 3가지 경우 모두에서 본문에 있는 JSON 형식의 등록 정보와 함께 사용자에게 대한 POST 요청이 전송됩니다.

표 92 새 사용자 만들기 등록 정보

등록 정보	유형	설명
logname	string	새 사용자의 로그인 이름(필수)
uid	number	선택적 사용자 ID
fullname	string	새 사용자의 전체 이름(필수)
type	string	"local", "data", "nologin"(기본값: "local")
initial_password	string	초기 사용자 암호("local" 및 "data" 만)
require_annotation	boolean	세션 주석 필요 선택적 플래그("local"만)

표 93 사용자 복제 등록 정보

등록 정보	유형	설명
user	string	소스 사용자 이름
uid	number	사용자 ID, 디렉토리 사용자에게 대해 사용으로 설정되지 않음
clonename	string	새 복제 로그인 이름
fullname	string	새 복제 사용자 전체 이름(디렉토리 사용자의 경우 해당하지 않음)
password	string	새 복제 사용자 암호(디렉토리 또는 비로그인 사용자의 경우 해당하지 않음)

표 94 관리자 추가 등록 정보

등록 정보	유형	설명
type	string	디렉토리 사용자

등록 정보	유형	설명
logname	string	디렉토리 사용자 로그인 이름

예 1 로컬 사용자 만들기

요청 예:

```
POST /api/user/v1/users HTTP/1.1
Host: zfs-storage.example.com:215
Authorization: Basic abcdefghijklmnop
Content-type: application/json
```

```
{
  "type": "local",
  "logname": "admin3",
  "initial_password": "password",
  "fullname": "Administrator"
}
```

결과 예:

```
{
  "user": {
    "href": "/api/user/v1/users/admin3",
    "logname": "admin3",
    "type": "local",
    "uid": 2000000002,
    "fullname": "Administrator",
    "initial_password": "password",
    "require_annotation": false,
    "roles": [
      "basic"
    ],
    "kiosk_mode": false,
    "kiosk_screen": "status/dashboard",
    "exceptions": [
    ],
    "preferences": {
      "href": "/api/user/v1/users/admin3/preferences",
      "locale": "C",
      "login_screen": "status/dashboard",
      "session_timeout": 15,
      "advanced_analytics": false,
      "keys": [
      ]
    }
  }
}
```

예 2 디렉토리 사용자 만들기

요청 예:

```
POST /api/user/v1/users
{
  "type": "directory",
  "logname": "admin3"
}
```

결과 예:

```
{
  "user": {
    "href": "/api/user/v1/users/admin3",
    "logname": "admin3",
    "type": "directory",
    "uid": 26718,
    "fullname": "Administrator",
    "require_annotation": false,
    "roles": [
      "basic"
    ],
    "kiosk_mode": false,
    "kiosk_screen": "status/dashboard",
    "exceptions": [
    ],
    "preferences": {
      "href": "/api/user/v1/users/admin3/preferences",
      "locale": "C",
      "login_screen": "status/dashboard",
      "session_timeout": 15,
      "advanced_analytics": false,
      "keys": [
      ]
    }
  }
}
```

예 3 데이터 전용 사용자 만들기

요청 예:

```
POST /api/user/v1/users
{
  "type": "data",
  "logname": "admin3",
  "initial_password": "password",
  "fullname": "Administrator",
  "uid": 5000000
}
```

결과 예:

```
{
  "user": {
    {
      "href": "/api/user/v1/users/data",
      "logname": "admin3",
      "type": "data",
      "uid": 5000000,
      "fullname": "Administrator",
      "initial_password": "password"
    }
  }
}
```

예 4 비로그인 사용자 만들기

요청 예:

```
POST /api/user/v1/users
{
  "type": "nologin",
  "logname": "admin3",
  "fullname": "Administrator",
  "uid": 5000001
}
```

결과 예:

```
{
  "user": {
    "href": "/api/user/v1/users/admin3",
    "logname": "admin3",
    "type": "nologin",
    "uid": 5000001,
    "fullname": "Administrator"
  }
}
```

사용자 수정

직접 사용자 등록 정보를 수정합니다. 사용자 리소스: 예외, 환경 설정 및 ssh 키가 추가, 수정 또는 제거될 수 있습니다. "UID" 및 "Type"은 만든 후 변경할 수 없습니다.

요청 예:

```
PUT /api/user/v1/users/admin1 HTTP/1.1
Authorization: Basic abcdefgMWE=
Host: zfs-storage.example.com:215
Accept: application/json
Content-Type: application/json
Content-Length: 24

{"require_annotation": true}
```

결과 예:

```
HTTP/1.1 202 Accepted
X-Zfssa-Appliance-API: 1.0
Content-Type: application/json
Content-Length: 236

{
  "user": {
    "href": "/api/user/v1/users/admin3",
    "logname": "admin3",
    "type": "local",
    "uid": 2000000000,
    "fullname": "Administrator",
    "initial_password": "password",
    "require_annotation": true,
    "kiosk_mode": false,
    "kiosk_screen": "status/dashboard",
    "roles": ["basic"]
  }
}
```

사용자 삭제

시스템에서 사용자를 삭제합니다.

요청 예:

```
DELETE /api/user/v1/users/admin1 HTTP/1.1
Authorization: Basic abcefgMWE=
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 95 워크플로우 서비스 명령

요청	경로 <i>/api/workflow/v1</i> 에 추가	설명
GET	<i>/api/workflow/v1</i> 만 사용	워크플로우 서비스 명령을 나열합니다.
POST	<i>/workflows</i>	새 워크플로우를 어플라이언스로 업로드합니다.
GET	<i>/workflows</i>	모든 워크플로우를 나열합니다.
GET	<i>/workflows/workflow</i>	지정된 워크플로우 등록 정보를 나열합니다.
PUT	<i>/workflows/workflow</i>	지정된 워크플로우 등록 정보를 수정합니다.
PUT	<i>/workflows/workflow/execute</i>	지정된 워크플로우를 실행합니다.
DELETE	<i>/workflows/workflow</i>	지정된 워크플로우를 삭제합니다.
POST	<i>/scripts</i>	스크립트를 업로드 및 실행합니다.
GET	<i>/scripts</i>	실행 중인 모든 스크립트를 나열합니다.
GET	<i>/scripts/script</i>	실행 중인 스크립트에 재연결합니다.
DELETE	<i>/scripts/script</i>	실행 중인 스크립트를 중지합니다.

워크플로우 업로드

워크플로우를 어플라이언스에 업로드합니다.

요청 예:

```
POST /api/workflow/v1/workflows HTTP/1.1
Authorization: Basic abcefgMWE=
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 abcefgMWE=
Host: zfs-storage.example.com:215
Accept: application/json
```

응답 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-cf5e527df92d",
    ...
  },
  {
    "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 cm9vdDpsMWE=
Host: zfs-storage.example.com:215
Accept: application/javascript

```

응답 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 cm9vdDpsMWE=
Host: zfs-storage.example.com:215
Accept: application/json

```

응답 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 abcefgMWE=
Host: zfs-storage.example.com:215
Accept: application/json
Content-Type: application/json
Content-Length: 28
```

```
{"setid": false}
```

응답 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 abcefgMWE=
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 abcefgMWE=
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 [Oracle ZFS Storage Appliance 관리 설명서, 릴리스 OS8.8.0의 “CLI 스크립트 작성 작업”](#)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 dDpScm9vMWE=
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 [Oracle ZFS Storage Appliance 관리 설명서, 릴리스 OS8.8.0의 "CLI 스크립트 작성 작업"](#)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 cm9vdDpsMWE=
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 [Oracle ZFS Storage Appliance 관리 설명서, 릴리스 OS8.8.0](#)의 “CLI 스크립트 작성 작업”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 cm9vdDpsMWE=
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 [Oracle ZFS Storage Appliance 관리 설명서, 릴리스 OS8.8.0의 "CLI 스크립트 작성 작업"](#)을 참조하십시오.

요청 예:

```

$ curl -kv -X DELETE --user root:l1a \
  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 cm9vdDpsMWE=
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/nas/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,
      "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,
    "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
```