Sun Java System Web Server 7.0
관리자 설명서

Sun Microsystems, Inc.
4150 Network Circle
Santa Clara, CA 95054
U.S.A.

부품 번호: 820–0874
2006년 10월

070531@17466

목차

그림

표

코드 예

머리말

이 설명서에서는 **Sun Java™ System Web Server 7.0**을 구성 및 관리하는 방법에 대해 설명합니다.

본 설명서의 대상

이 설명서는 프로덕션 환경에서 서버를 관리하는 Sun Java System Web Server 관리자를 대상으로 합니다. 이 설명서에서는 사용자가 다음 사항에 대해 잘 알고 있다고 가정합니다.

- 소프트웨어 설치
- 웹 브라우저 사용
- 기본 시스템 관리 작업 수행
- 단말기 창에서 명령 실행

이 설명서를 읽기 전에

Sun Java System Web Server 7.0은 독립 실행형 제품으로 설치할 수도 있고 네트워크 또는 인터넷 환경에서 배포된 엔터프라이즈 응용 프로그램을 지원하는 소프트웨어 인프라인 Sun Java Enterprise System(Java ES)의 구성 요소로 설치할 수도 있습니다. Sun Java System Web Server 7.0을 Java ES의 구성 요소로 설치하는 경우에는 `http://docs.sun.com/coll/1286.2` 및 `http://docs.sun.com/coll/1397.2`의 시스템 설명서를 잘 이해해야 합니다.

Sun Java System Web Server 7.0 설명서 세트

Sun Java System Web Server 7.0 설명서 세트에서는 Web Server를 설치하고 관리하는 방법에 대해 설명합니다. Sun Java System Web Server 7.0 설명서에 대한 URL은 http://docs.sun.com/coll/1308.3 및 http://docs.sun.com/coll/1410.2입니다. Sun Java System Web Server 7.0에 대한 소개 내용을 보려면 다음 표에 나열된 설명서를 순서대로 참조하십시오.

표 P–1 Sun Java System Web Server 7.0 설명서 세트에 포함된 설명서

설명서 제목	목차
Sun Java System Web Server 7.0 Documentation Center	작업 및 주제별로 정리된 Web Server 설명서 항목
Sun Java System Web Server 7.0 릴리스 노트	■ 소프트웨어 및 설명서에 대한 최신 정보 ■ Web Server 설치에 지원되는 플랫폼과 패치 요구 사항
Sun Java System Web Server 7.0 Installation and Migration Guide	설치 및 마이그레이션 작업 수행: ■ Web Server 및 다양한 구성 요소 설치 ■ Sun ONE Web Server 6.0 또는 6.1에서 Sun Java System Web Server 7.0으로 데이터 마이그레이션
Sun Java System Web Server 7.0 관리자 설명서	다음 관리 작업 수행: ■ 관리 및 명령줄 인터페이스 사용 ■ 서버 기본 설정 구성 ■ 서버 인스턴스 사용 ■ 서버 작동 모니터링 및 로깅 ■ 서버 보안을 위한 인증서 및 공용 키 암호화 사용 ■ 서버 보안을 위한 액세스 제어 구성 ■ JavaPlatform Enterprise Edition(Java EE) 보안 기능 사용 ■ 응용 프로그램 구현 ■ 가상 서버 관리 ■ 성능 요구에 맞춰 서버 작업 부하 정의 및 시스템의 규모 설정 ■ 서버 문서의 컨텐트 및 속성 검색, 텍스트 검색 인터페이스 작성 ■ 컨텐트 압축용으로 서버 구성 ■ WebDAV를 사용한 웹 게시 및 내용 저작용으로 서버 구성
Sun Java System Web Server 7.0 Developer's Guide	프로그래밍 기법 및 API를 사용하여 다음 작업 수행: ■ Sun Java System Web Server 확장 및 수정 ■ 클라이언트 요청에 따라 동적으로 내용을 생성하고 서버의 내용 수정
Sun Java System Web Server 7.0 Update 1 NSAPI Developer's Guide	사용자 정의 NSAPI(Netscape Server Application Programmer's Interface) 플러그인 생성
Sun Java System Web Server 7.0 Developer's Guide to Java Web Applications	Sun Java System Web Server에서 Java Servlet 및 JSP™(JavaServer Pages™) 기술 구현
Sun Java System Web Server 7.0 Administrator's Configuration File Reference	구성 파일 편집
Sun Java System Web Server 7.0 Performance Tuning, Sizing, and Scaling Guide	Sun Java System Web Server를 조정하여 성능 최적화
Sun Java System Web Server 7.0 Troubleshooting Guide	Web Server 문제 해결

관련 설명서

Sun Java Enterprise System(Java ES) 및 구성 요소와 관련된 모든 설명서에 대한 URL은 http://docs.sun.com/app/docs/prod/entsys.06q4 및 http://docs.sun.com/app/docs/prod/entsys.06q4?l=ko 입니다.

기본 경로 및 파일 이름

다음 표에서는 이 설명서에서 사용되는 기본 경로 및 파일 이름을 설명합니다.

표 P–2 기본 경로 및 파일 이름

자리 표시자	설명	기본값

install_dir	Sun Java System Web Server 7.0의 기본 설치 디렉토리를 나타냅니다.	Solaris™ 플랫폼에 Sun Java Enterprise System(Java ES) 설치:
		`/opt/SUNWwbserver7`
		Linux 및 HP-UX 플랫폼에 Java ES 설치:
		`/opt/sun/webserver/`
		Windows 플랫폼에 Java ES 설치:
		System Drive`:\Program Files\Sun\JavaES5\WebServer7`
		기타 Solaris, Linux 및 HP-UX 설치(루트가 아닌 사용자의 경우):
		user's home directory`/sun/webserver7`
		기타 Solaris, Linux 및 HP-UX 설치(루트 사용자의 경우):
		`/sun/webserver7`
		Windows(모든 설치):
		SystemDrive`:\Program Files\Sun\WebServer7`

표 P–2 기본 경로 및 파일 이름 *(계속)*

자리 표시자	설명	기본값
instance_root	인스턴스별 하위 디렉토리를 포함하는 디렉토리입니다.	Solaris에서 인스턴스의 기본 위치: `/var/opt/SUNWwbsvr7`. Linux 및 HP-UX에서 인스턴스의 기본 위치: `/var/opt/sun/webserver7` Windows에서 인스턴스의 기본 위치: *System Drive*`:\Program Files\sun\WebServer7` Java ES 설치의 경우, Windows에서 인스턴스의 기본 위치: *System Drive*`:\Program Files\Sun\JavaES5\WebServer7`

활자체 규칙

다음 표는 이 책에서 사용된 활자체 변경 사항에 대하여 설명합니다.

표 P–3 활자체 규칙

서체	의미	예
`AaBbCc123`	명령, 파일 및 디렉토리와 화면 상의 컴퓨터 출력을 나타냅니다.	`.login` 파일을 편집합니다. 모든 파일을 나열하려면 `ls -a`를 사용합니다. `machine_name% you have mail.`
`AaBbCc123`	컴퓨터 화면 상의 출력과는 달리 사용자가 직접 입력하는 사항입니다.	`machine_name%` **`su`** `Password:`
AaBbCc123	명령줄 자리 표시자: 실제 이름이나 값으로 대체됩니다.	파일을 제거하는 명령은 `rm` *filename*입니다.
AaBbCc123	책 제목, 새로 나오는 용어, 강조 표시할 단어입니다. (일부 강조된 항목은 온라인상에서 볼드로 표시됩니다.)	**사용자 설명서**의 6장을 읽으십시오. **캐시**는 로컬로 저장된 복사본입니다. 파일을 저장하지 **마십시오**.

기호 규칙

다음 표는 이 설명서에서 사용되는 기호를 설명합니다.

표 P–4 기호 규칙

기호	설명	예	의미
`[ ]`	선택 인수 및 명령 옵션을 포함합니다.	`ls [-l]`	`-l` 옵션은 필수가 아닙니다.
`{ | }`	필수 명령 옵션에 대한 일련의 선택 항목을 포함합니다.	`-d {y|n}`	`-d` 옵션에서는 `y` 인수나 `n` 인수를 사용해야 합니다.
`${ }`	변수 참조를 나타냅니다.	`${com.sun.javaRoot}`	`com.sun.javaRoot` 변수 값을 참조합니다.
-	동시에 입력하는 여러 키를 결합합니다.	Ctrl-A	A 키를 누른 채로 Ctrl 키를 누릅니다.
+	연속해서 입력하는 여러 키를 결합합니다.	Ctrl+A+N	Ctrl 키를 눌렀다가 놓은 다음 후속 키를 누릅니다.
→	그래픽 사용자 인터페이스의 메뉴 항목 선택을 나타냅니다.	파일 → 새로 만들기 → 템플리트	파일 메뉴에서 새로 만들기를 선택합니다. 새로 만들기 하위 메뉴에서 템플리트를 선택합니다.

Sun 자원 온라인 액세스

`http://docs.sun.com`(docs.sun.com℠) 웹 사이트에서는 Sun 기술 관련 설명서를 온라인으로 이용할 수 있습니다. docs.sun.com 아카이브를 탐색하거나 특정 책 제목 또는 주제를 검색할 수 있습니다. 설명서는 PDF 및 HTML 형식의 온라인 파일로 제공됩니다. 장애가 있는 사용자도 보조 기술을 이용하여 두 형식을 모두 읽을 수 있습니다.

다음의 Sun 자원에 액세스하려면 `http://www.sun.com`을 방문하십시오.

- Sun 제품 다운로드
- 서비스 및 솔루션
- 지원(패치 및 업데이트 포함)
- 교육
- 리서치
- 커뮤니티(예: Sun 개발자 네트워크)

Sun 제품 설명서 검색

docs.sun.com 웹 사이트에서 Sun 제품 설명서를 검색하는 것 외에도 검색 필드에서 다음 구문을 입력하여 검색 엔진을 사용할 수 있습니다.

search-term `site:docs.sun.com`

예를 들어 "Web Server"를 검색하려면 다음을 입력합니다.

```
Web Server site:docs.sun.com
```

검색에 다른 Sun 웹 사이트(예: java.sun.com, www.sun.com, developers.sun.com)를 포함하려면 검색 필드에서 "`docs.sun.com`" 대신 "`sun.com`"을 사용합니다.

타사 웹 사이트 참조

이 문서에서는 추가적인 관련 정보를 제공하기 위해 타사 URL을 참조하기도 합니다.

주 – Sun은 이 설명서에 언급된 타사 웹 사이트의 가용성에 대해 책임지지 않습니다. Sun은 이러한 사이트나 자원을 통해 사용할 수 있는 내용, 광고, 제품 또는 기타 자료에 대해서는 보증하지 않으며 책임지지 않습니다. Sun은 해당 사이트 또는 자원을 통해 사용 가능한 내용, 제품 또는 서비스의 사용과 관련해 발생한 사실이 있거나 발생했다고 주장이 제기되는 손해나 손실에 대해 책임이나 의무를 지지 않습니다.

사용자 의견 환영

Sun은 설명서의 내용 개선에 노력을 기울이고 있으며, 여러분의 의견과 제안을 환영합니다. 사용자 의견을 보내시려면 `http://docs.sun.com`에서 Send Comments(의견 보내기)를 누릅니다. 온라인 양식에서 전체 설명서 제목과 부품 번호를 입력합니다. 부품 번호는 해당 설명서의 제목 페이지나 문서의 URL에 있으며 7자리 또는 9자리 숫자로 되어 있습니다. 예를 들어, 이 설명서의 부품 번호는 820-0874입니다.

◆ ◆ ◆ 1 장

1 시작하기

이 장에서는 이 설명서에 사용되는 용어를 간략하게 소개하여 Sun Java System Web Server 7.0의 기본 사항에 대해 설명합니다.

개요

Sun Java System Web Server 7.0은 업계 표준을 기반으로 구축된 다중 프로세스, 다중 스레드의 보안 웹 서버로, 중/대기업을 위해 고성능, 안정성, 확장 가능성 및 관리 용이성을 제공합니다.

Web Server 7.0은 포괄적인 명령줄 인터페이스 지원, 통합 구성, Elliptic Curve Cryptography 지원을 통한 향상된 보안 및 클러스터링 지원을 제공합니다. 또한 Web Server 6.0 및 6.1의 응용 프로그램과 구성을 Sun Java System Web Server 7.0으로 마이그레이션하는 데 도움이 되는 견고한 내장 마이그레이션 도구가 함께 제공됩니다.

새로운 기능

Sun Java System Web Server 7.0의 새로운 기능과 향상된 기능에 대한 자세한 내용은 **Sun Java System Web Server 7.0 릴리스 노트**의 1 장, “Sun Java System Web Server 릴리스 노트”를 참조하십시오.

Administration Server 시작

관리 인터페이스를 사용하려면 Administration Server를 시작해야 합니다.

Unix/Linux에서 Administration Server 시작

Administration Server를 시작하려면 다음 작업을 수행하십시오.

▼ Unix/Linux에서 Administration Server 시작

1 `install_root/admin-server/bin` 디렉토리로 이동합니다(예: `/usr/sjsws7.0/admin-server/bin`).

2 `./startserv`를 입력합니다.

이 명령에 따라 Administration Server가 시작되며 설치 시에 지정한 포트 번호가 사용됩니다.

Windows에서 Administration Server 시작

Sun Java System Web Server 설치 프로그램은 Windows 플랫폼용으로 여러 개의 아이콘이 있는 프로그램 그룹을 만듭니다. 이 프로그램 그룹에는 다음과 같은 아이콘이 포함됩니다.

- 릴리스 노트
- Administration Server 시작
- Web Server 제거

Administration Server는 서비스 애플릿으로 실행되기 때문에 제어판을 사용하여 이 서비스를 직접 시작할 수도 있습니다.

여러 가지 서버 관리 방법

다음 사용자 인터페이스를 사용하여 Sun Java System Web Server를 관리할 수 있습니다.

- 관리 콘솔(GUI)
- 명령줄 인터페이스(`wadm` 쉘).

`wadm` 쉘 인터페이스(이 장의 뒷부분에서 설명됨) 또는 웹 기반 관리 콘솔을 사용하여 인스턴스를 관리할 수 있습니다. 관리 노드에서는 특정 구성의 인스턴스가 하나만 실행될 수 있습니다.

관리 콘솔 사용

Sun Java System Web Server를 설치한 후에 브라우저를 사용하여 관리 콘솔에 액세스합니다.

Administration Server 페이지로 이동하는 데 사용하는 URL은 Sun Java System Web Server를 설치할 때 Administration Server용으로 선택한 컴퓨터 호스트 이름과 포트 번호에 따라 다릅니다. 예를 들어 Administration Server를 SSL 포트 1234에 설치한 경우 URL은 다음과 같습니다.

```
https://myserver.sun.com:1234/
```

서버 관리를 수행하려면 관리 콘솔에 로그인해야 합니다. 컴퓨터에 Sun Java System Web Server를 설치할 때 관리자 이름과 비밀번호를 설정합니다. 다음 그림은 인증 화면을 나타냅니다.

Administration Server에 액세스하면 일반 작업 페이지가 처음에 표시됩니다. 이 페이지에 있는 버튼을 사용하여 Sun Java System Web Server를 관리, 추가, 제거 및 마이그레이션할 수 있습니다. 다음 그림은 일반 작업 페이지를 나타냅니다.

주 – 이 탭 중 하나를 누르면 페이지에 하위 탭이 나타날 수 있습니다. 하위 탭에서 제공되는 작업은 상위 탭 기능에 따라 결정됩니다.

다음 그림은 선택한 탭의 하위 탭을 보여줍니다.

탭을 누르면 같은 창에 페이지가 열립니다. 일련의 단계를 통해 사용자로부터 데이터를 수집하는 작업이 있습니다. 관리 콘솔에는 이러한 작업을 위한 마법사 인터페이스가 있습니다. 마법사는 항상 새 창에서 열립니다.

관리 콘솔 GUI 화면에 대한 도움말

모든 형식 요소와 GUI 구성 요소에는 확인 및 선택 매개 변수에 대한 정보를 제공하는 자세한 인라인 도움말이 있습니다. 마법사 인터페이스의 경우 마법사의 어느 단계에서나 도움말 탭을 누르면 현재 작업에 해당되는 도움말을 읽을 수 있습니다.

CLI 사용

이 절에서는 Sun Java System Web Server 7.0의 명령줄 인터페이스에 대해 설명하며 서버 구성 및 관리를 위해 지원되는 모든 명령을 정의합니다.

Sun Java System Web Server 7.0에는 `wadm`이라는 새 CLI가 도입되었습니다.

이전 버전의 서버는 몇 개의 개별적인 명령줄만 지원했는데, 이는 GUI에 제공되는 관리 기능 중 일부 하위 기능만 처리할 수 있었습니다. Sun Java System Web Server 6.1에서 지원되는 명령줄 인터페이스는 `HttpServerAdmin`, `wdeploy` 및 `flexanlg`입니다. 새 CLI(`wadm`)는 다음 기능을 제공합니다.

- 스크립트 작성을 위해 내장된 JACL 쉘
- 확장 가능한 CLI — 타사 플러그인을 사용하여 CLI에 명령을 더 추가할 수 있습니다.

주 – Sun Java System Web Server 7.0은 `HttpServerAdmin`을 지원하지 않습니다.

주 – `wdeploy`는 Sun Java System Web Server 7.0에서 6.x 버전과의 역방향 호환성을 위해서만 지원되며 Administration Server 노드에서만 작동합니다.

CLI의 모드

`wadm`은 서로 다른 3가지 모드로 호출을 지원합니다. 해당 모드는 다음과 같습니다.

- **독립 실행형 모드** — 이 모드에서는 원하는 명령, 옵션 및 피연산자를 지정하여 명령 쉘에서 `wadm`을 호출합니다. 명령 실행이 끝나면 CLI가 종료되고 쉘로 돌아갑니다. 이 모드는 대화형 및 비대화형 명령 실행을 모두 지원합니다. 기본 설정인 대화형 실행은 비밀번호 파일(`--password-file` 옵션을 통해 전달됨)에 비밀번호가 지정되어 있지 않은 경우 비밀번호를 묻는 프롬프트를 표시합니다. 비대화형 실행의 경우 `--password-file` 옵션이 지정되지 않으면 오류가 발생합니다. 예를 들면 다음과 같습니다

```
wadm> create-config --user=admin --password-file=./admin.pwd
--http-port=2222 --server-name=syrinx myconfig
```

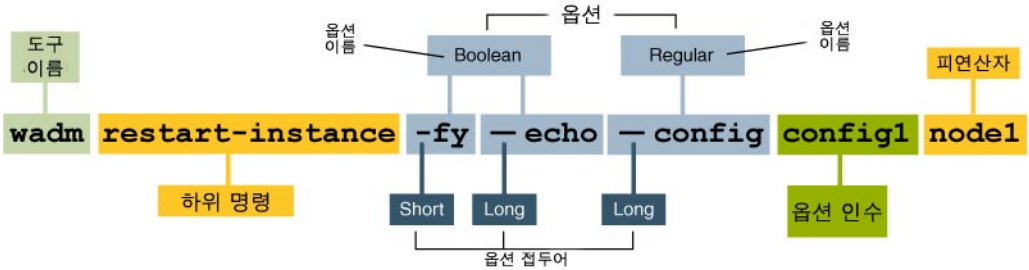
- **쉘 모드** — 이 모드에서는 명령 없이 명령 쉘로부터 `wadm`을 호출합니다. `wadm`에서 명령 입력 프롬프트를 표시합니다. 명령이 실행된 후 쉘로 돌아갑니다. exit 또는 quit 명령을 입력하면 이 쉘을 종료할 수 있습니다. 이 모드에서 대화형 및 비대화형 실행을 사용할 수 있습니다. 예를 들면 다음과 같습니다

```
wadm -user=admin -host=serverhost --password-file=admin.pwd --port=8989
```

- **파일 모드** — 이 모드에서는 파일에 명령 목록을 추가하고 `wadm`의 인수로 파일을 전달할 수 있습니다. 예를 들면 다음과 같습니다

```
wadm -user=admin -host=serverhost --password-file=admin.pwd
--port=8989 -commands-file=/space/scripts/admscr
```

아래 그림은 `wadm` 명령을 호출하는 구문을 나타냅니다.



주 – `wadm` CLI를 사용하면 관리 콘솔로 수행할 수 있는 모든 작업을 수행할 수 있습니다.

`wadm` CLI를 찾을 수 있는 위치

질문: Sun Java System Web Server 7.0 관리에 사용할 CLI는 어디에서 찾을 수 있습니까?

응답: 관리 CLI는 `install-root/bin/wadm`에 있습니다. CLI를 사용하려면 다음에 대해 알고 있어야 합니다.

- Administration Server 호스트 이름(기본값은 `localhost`)
- Administration Server의 SSL 포트(기본값은 8989)
- Administration Server 사용자 이름(기본값은 `admin`)
- Administration Server 비밀번호

주 – CLI를 사용하려면 Administration Server를 실행하고 있어야 합니다. `install-root/admin-server/bin/startserv`를 실행하여 서버를 시작할 수 있습니다.

CLI에서의 인증

wadm은 관리자의 사용자 이름과 비밀번호를 사용하여 Administration Server를 인증합니다. 단일 모드로 실행되는 각 명령에 대해 유효한 사용자 이름과 비밀번호 파일을 인수로 전달해야 합니다. 쉘 모드는 wadm 실행 파일이 호출될 때 사용자 이름과 비밀번호 파일을 받습니다. 쉘 모드로 호출되는 명령에는 연결 옵션(예: user, password-file,host, port 및 ssl)이 필요하지 않습니다. 연결 옵션을 지정해도 무시됩니다.

CLI에서 지원하는 명령 중에는 비밀번호 입력이 필요한 것도 있습니다. 예를 들면 bindpw, user-password, token-pin 등입니다. 사용자는 관리자 비밀번호가 포함된 동일한 파일에 이 비밀번호를 지정할 수 있습니다. 명령과 함께 password-file을 지정하지 않으면 사용자에게 비밀번호 입력 프롬프트가 표시됩니다.

Administration Server에서 SSL을 사용하는 경우 wadm은 SSL을 통해 Administration Server와 통신합니다. Administration Server에서 전달된 인증서는 *truststore(~/.wadmtruststore)*에 대해 확인됩니다. 인증서가 있으며 유효한 경우 명령이 정상적으로 처리됩니다. 그렇지 않은 경우 wadm에서 인증서를 표시하며 사용자가 수락 여부를 선택할 수 있습니다. 사용자가 인증서를 수락하면 인증서가 *truststore*에 추가되고 명령이 정상적으로 처리됩니다.

주 – *truststore*는 중요한 데이터를 포함하지 않기 때문에 비밀번호로 보호할 필요가 없습니다.

Web Server 7.0 이 해

Web Server에는 서버 팜의 서버 전체에 대해 향상된 분산 관리 기능을 제공하는 새로운 관리 프레임워크가 포함되어 있습니다. 강력한 관리 기능을 통해 그래픽 인터페이스와 명령줄 인터페이스를 모두 사용하여 Web Server를 원격으로 관리 및 배포할 수 있습니다. 서버 팜의 중앙 위치에서 서버를 관리하고 하나 이상의 노드에 배포하여 서버 인스턴스를 만들 수 있습니다. 이러한 서버에 대한 모니터링 및 라이프사이클 관리 기능도 제공됩니다.

Web Server는 다양한 기능을 설정 또는 해제하거나, 개별 클라이언트 요청에 대한 응답 방식을 결정하거나, 서버에서 실행되며 서버의 작업과 통합되는 프로그램을 작성할 수 있도록 구성됩니다. 이러한 옵션을 식별하는 지침(지시문)은 구성 파일에 저장됩니다. Sun Java System Web Server는 시작할 때와 클라이언트 요청 중에 구성 파일을 읽어 선택 사항을 원하는 서버 작동과 매핑합니다.

이 파일에 대한 자세한 내용은 Sun Java System Web Server 7.0 **관리자 구성 파일 참조 설명서**를 참조하십시오.

Web Server 7.0에서는 웹 응용 프로그램, 구성 파일, 검색 모음 색인과 같은 서버 인스턴스의 구성 가능한 모든 요소가 논리적으로 그룹화되며 **구성**이라고 불립니다. 구성은 CLI 또는 웹 기반 관리 인터페이스를 사용하여 작성, 수정 또는 삭제할 수

있습니다. 한 번에 두 개 이상의 구성을 관리할 수 있습니다. 구성이라는 용어는 서버의 런타임 서비스를 구성하는 메타데이터 집합을 나타내기도 합니다. 예를 들어 런타임 서비스는 구성된 문서 루트의 웹 페이지를 제공합니다. 구성 메타데이터는 서버 런타임이 내장 서비스와 타사 플러그인을 로드하고, 데이터베이스 드라이버와 같은 다른 서버 확장이 웹 페이지 및 동적 웹 응용 프로그램을 제공하도록 설정하는 데 사용됩니다.

주 – 구성 관련 파일은 모두 파일 시스템의 **구성 저장소**라고 하는 저장소에 저장됩니다. 이 설명서에 명시되어 있지 않은 한 이 저장소에 있는 파일을 직접 편집하면 안 됩니다.

Web Server에서 CLI를 사용하거나 웹 기반 관리 인터페이스를 통해 수행하는 구성 변경은 먼저 구성 저장소에서 이루어지며, 그 후에 구성이 배포됩니다. 이어서 변경 사항이 인스턴스 디렉토리에 복사됩니다. 웹 응용 프로그램은 다음 위치에 배포됩니다.

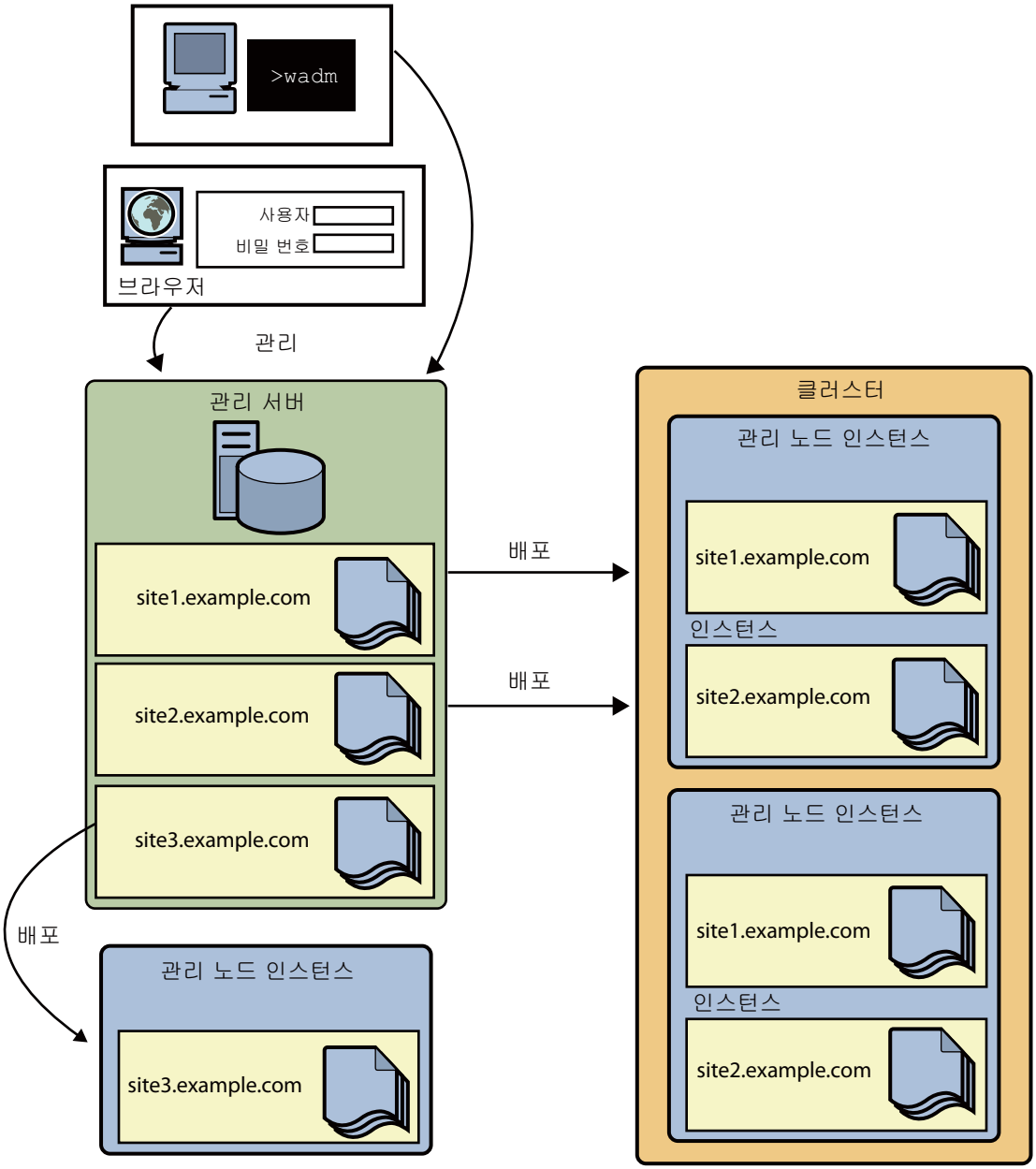
```
<install_dir>/admin-server/config-store/<config_name>/web-app/<virtual_servername>/
```

구성을 배포하면 `config-store` 아래에 있는 전체 웹 응용 프로그램 디렉토리 및 구성 디렉토리가 압축되어 서버 인스턴스 디렉토리에 복사됩니다. 이 파일은 다음 위치에 있는 `current.zip` 파일입니다.

```
<install_dir>/admin-server/config-store/<config_name>
```

따라서 웹 응용 프로그램의 크기에 따라 선택한 구성의 배포를 완료하는 데 시간이 걸릴 수 있습니다.

다음 그림은 구성을 관리 노드에 배포하는 방식을 개략적으로 나타낸 그림입니다.



노드(서버 또는 호스트 등의 네트워크 자원)에 구성을 배포하면 해당 구성의 **인스턴스**가 만들어집니다. 인스턴스에는 로그 파일을 비롯해 잠금 데이터베이스, 캐시 및 인스턴스에 필요한 임시 파일 등의 다른 런타임 파일이 포함되어 있습니다. CLI 또는 웹 기반 관리 인터페이스를 통해 이러한 인스턴스를 관리할 수 있습니다.

인스턴스가 하나 이상의 노드에 걸쳐 **클러스터**를 형성할 수 있습니다. 클러스터의 경우 클러스터를 형성하는 모든 노드는 구성이 동일해야 합니다. 클러스터에 있는 모든 노드는 동일한 유형이어야 합니다. 이러한 인스턴스는 운영 체제가 동일해야 하고 동일하게 구성되며 동일한 서비스를 제공해야 합니다.

서버 팜에서 하나의 노드에는 관리 응용 프로그램이 배포된 서버가 실행되고 있습니다. 특별하게 구성된 이러한 서버를 **Administration Server**라고 하며, 여기에 배포되는 관리 응용 프로그램은 웹 기반 **관리 콘솔**이라고 합니다. 관리 콘솔을 사용하면 서버 인스턴스의 라이프사이클을 제어할 수 있습니다.

Administration Server는 그 노드의 다른 서버(**관리 노드**)에서 이루어지는 작업을 제어합니다. 관리 노드는 GUI 인터페이스를 제공하지 않습니다. 서버 팜의 한 노드에는 Administration Server가 설치됩니다. 서버 팜에 있는 다른 모든 노드에는 관리 노드가 설치됩니다. 관리 노드는 설치 즉시 Administration Server에 등록됩니다. 이 작업으로 Administration Server가 관리 노드를 인식하게 됩니다.

Administration Server와 관리 노드는 항상 SSL을 통해 통신합니다. Administration Server와 관리 노드는 Administration Server에서 관리 노드의 서버 인증서를 신뢰하고 관리 노드에서 Administration Server가 제시하는 클라이언트 인증서를 신뢰하는 방식으로 서로를 인증합니다. 관리 노드를 등록하는 동안 Administration Server는 해당 관리 노드의 서버 인증서를 생성합니다. 생성된 인증서는 다운로드되어 관리 노드에 설치됩니다. 서버 인증서 발급자 역시 관리 노드에 설치됩니다.

◆ ◆ ◆ 2 장

2 구성, 인스턴스 및 노드

앞의 장에서는 Web Server 7.0에 도입된 몇 가지 새로운 개념에 대해 소개했습니다. 관리자의 기본적인 작업은 서버의 런타임 서비스를 구성하고 관리하는 것입니다. 이 장에서는 구성을 관리하는 여러 가지 방법과 구성을 배포하여 노드에서 인스턴스를 시작하는 방법에 대해 설명합니다.

개요

인스턴스는 지정된 노드에서 웹 서버 데몬의 환경을 나타내며, 해당 구성 및 로그 파일을 비롯하여 잠금 데이터베이스, 캐시, 임시 파일 등과 같은 다른 런타임 아티팩트를 포함합니다.

노드는 서버 또는 호스트와 같은 네트워크 자원입니다. 일반적인 데이터 센터에서 노드 네트워크는 **서버 팜**이라고 합니다. 이 절에서는 관리 콘솔 GUI를 사용하여 노드를 구성하는 방법에 대해 설명합니다.

노드에 하나 이상의 인스턴스를 배포할 수 있습니다. 같은 인스턴스를 여러 노드에 배포하여 서로 다른 클러스터의 부분을 형성할 수도 있습니다.

관리 용도로 인스턴스를 시작, 중지하거나 동적으로 다시 구성할 수 있습니다.

구성 관리

구성 만들기

웹 서버를 사용하기 시작하려면 구성을 만들어야 합니다.

새 구성을 만들려면 다음 작업을 수행하십시오.

1. **구성 탭**을 누릅니다.
2. **새로 만들기 버튼**을 누릅니다.

마법사가 나타나며 구성 만들기에 사용할 수 있는 설정을 안내합니다. 다음 절에서는 마법사 페이지에서 사용할 수 있는 필드에 대해 설명합니다.

단계 1 – 구성 정보 설정

이 마법사 페이지에서는 새 구성에 대한 일반 정보를 설정할 수 있습니다.

마법사 페이지에서 다음 매개 변수를 설정합니다.

- **구성 이름** — 구성에 고유한 새 이름을 추가합니다.
- **서버 이름** — 새 구성에 서버 이름을 추가합니다. 이 이름은 구성 이름과 동일할 수 있습니다.
- **문서 루트** — 배포된 모든 웹 응용 프로그램이 해당 디렉토리를 유지 관리하는 유효한 문서 루트를 입력합니다. 기본값은 ..../docs입니다. 서버의 유효한 모든 디렉토리 경로를 입력할 수 있습니다.
- **64비트** — 웹 서버의 64비트 지원을 활성화/비활성화 합니다. 기본값은 **비활성화**입니다.
- **서버 사용자** — UNIX 기반 시스템에서 서버가 실행 중인 경우 서버 프로세스에 대해 유효한 사용자 이름을 입력합니다. 예: root

단계 2 — 구성에 대한 수신기 만들기

이 마법사 페이지에서는 새 구성에 대해 HTTP Listener 등록 정보를 설정할 수 있습니다.

마법사 페이지에서 다음 매개 변수를 설정합니다.

- **포트** — 구성을 바인드하고 요청을 수신하는 포트 번호를 입력합니다.
- **IP 주소** — 호스트 시스템의 IP 주소입니다. 사용 가능한 모든 IP 주소를 설정하려면 *를 입력합니다.

단계 3 — Java, CGI 및 SHTML 구성

이 마법사 페이지에서는 Java/CGI 및 SHTML과 관련된 등록 정보를 구성할 수 있습니다.

마법사 페이지에서 다음 매개 변수를 설정합니다.

- **Java — 활성화**. 기본적으로 Java는 사용 가능으로 설정됩니다.**경고:**이 구성을 사용하여 Java 기반 웹 응용 프로그램을 배포하려면 Java를 비활성화하지 마십시오. Java SE 디렉토리의 홈을 설정합니다. 기본값은 번들된 Java SE 디렉토리를 가리키는 디렉토리입니다. 기본 Java SE 디렉토리를 선택하거나 새 경로를 지정할 수 있습니다.
- **CGI** — 없음(CGI 지원 비활성화), 파일 유형으로 활성화(CGI 지원 활성화) 및 디렉토리(CGI 문서가 저장되는 URI 및 경로 지정).
- **SHTML** — 기본적으로 SHTML은 비활성화되어 있습니다.

단계 4 — 인스턴스 만들기

이 마법사 페이지에서는 새 구성에 대한 인스턴스를 만들 수 있습니다.

마법사 페이지에서 다음 매개 변수를 설정합니다.

- 구성 — 새 구성의 이름입니다.
- 노드 선택 — 새 구성의 인스턴스를 만들기 위해 노드를 선택합니다. 사용 가능한 목록에서 노드를 선택하고 **추가** 또는 **모두 추가** 버튼을 눌러 노드를 추가합니다.

주 – CLI 사용

CLI를 통해 구성을 만들려면 다음 명령을 실행합니다.

```
wadm> create-config --user=admin --password-file=admin.pwd
--host=serverhost --port=8989 --http-port=8800 --server-user=user
--server-name=servername config1
```

config1은 새 구성의 이름입니다.

CLI 참조 `create-config`(1)를 참조하십시오.

서버 구성 복제

서버 구성을 복사하여 새 구성을 만들 수 있습니다. 새로 복사된 구성은 기존 구성과 동일합니다. 하지만 원본 구성에 인스턴스가 있었던 경우에도 새 구성에는 인스턴스가 없습니다.

구성을 복제하려면 다음 작업을 수행하십시오.

1. **구성 탭**을 누릅니다.

2. 목록에서 구성을 선택합니다.
3. **복사 버튼**을 누릅니다.
4. 팝업 창에서 새 구성 이름을 입력하고 확인을 누릅니다.

주 – CLI 사용

CLI를 통해 작업을 수행하려면 다음 명령을 실행합니다.

```
wadm> copy-config --user=admin --password-file=admin.pwd --host=serverhost
--port=8989 --config=config1 copyconfig1
```

copyconfig1은 새 구성의 이름입니다.

CLI 참조 `copy-config`(1)를 참조하십시오.

서버 구성 배포

노드에 구성을 배포하려면 먼저 구성을 만들어야 합니다.

기존 구성을 배포하려면 다음 작업을 수행하십시오.

1. **구성 탭**을 누릅니다.
2. 구성 확인란을 선택하여 구성을 확인합니다.
3. 구성 작업 목록에서 **구성 배포**를 선택합니다.

서버 구성 삭제

주 – 구성 인스턴스가 노드에 배포되어 있으면 구성을 삭제할 수 없습니다. 인스턴스가 배포되어 있으면 실행 중이 아니라도 서버 구성을 삭제할 수 없습니다. 구성을 삭제하려면 실행 중인 인스턴스를 중지하고 배포를 해제합니다.

구성을 삭제하려면 다음 작업을 수행하십시오.

1. **구성 탭**을 누릅니다.
2. 구성 확인란을 선택하여 구성을 확인합니다.
3. 구성 작업 목록에서 **구성 삭제**를 선택합니다.

서버 인스턴스 관리

서버 인스턴스 만들기

새 서버 인스턴스를 만들기 전에 다음 확인 작업을 수행하십시오.

1. 구성을 만들었는지 확인합니다. 새 서버 인스턴스를 만들려면 기존 인스턴스 구성을 지정해야 합니다.
2. 서버 팜에서 사용 가능한 모든 노드에 필요한 구성 인스턴스가 있는지 확인합니다. 중복 인스턴스는 만들 수 없습니다.

다음 작업을 수행하여 새 서버 인스턴스를 만듭니다.

1. **구성 탭**을 누르고 작업 목록에서 **새로 만들기**를 선택합니다.
2. 새 인스턴스 마법사 페이지에서 인스턴스를 만들 구성을 선택하고 **다음 버튼**을 누릅니다.
3. 선택한 구성의 인스턴스 [**단계** *2*]가 존재해야 할 노드를 선택합니다. **다음 버튼**을 누릅니다.
4. 선택 항목의 요약을 봅니다. **다음 버튼**을 눌러 작업 결과를 봅니다.

주 – CLI 사용

서버 인스턴스를 만들려면 다음 명령을 실행합니다.

```
wadm> create-instance --user=admin --password-file=admin.pwd
--host=serverhost --port=8989 --config=config1 serverhost
```

CLI 참조 `create-instance`(1)를 참조하십시오.

서버 인스턴스 시작

1. **노드 탭**을 눌러 서버에 구성된 노드 목록을 봅니다.
2. 노드 이름 확인란을 선택하여 노드를 선택합니다.
3. **인스턴스 시작 버튼**을 눌러 페이지 창을 열고 해당 노드에서 제어하는 모든 인스턴스를 나열합니다.

4. 인스턴스를 선택하고 **인스턴스 시작 버튼**을 눌러 인스턴스를 시작합니다.
5. 인스턴스 상태가 실행 중인지 확인하고 창을 닫습니다.

주 – CLI 사용

CLI를 통해 서버 인스턴스를 시작하려면 다음 명령을 실행합니다.

```
wadm> start-instance --user=admin --password-file=admin.pwd --host=serverhost
--port=8989 --config=config1 nodehost1
```

CLI 참조 `start-instance`(1)를 참조하십시오.

서버 인스턴스 중지

1. **노드 탭**을 눌러 서버에 구성된 노드 목록을 봅니다.
2. 노드 이름 확인란을 선택하여 노드를 선택합니다.
3. **인스턴스 중지 버튼**을 눌러 페이지 창을 열고 해당 노드에서 제어하는 모든 인스턴스를 나열합니다.
4. 인스턴스를 선택하고 **인스턴스 중지 버튼**을 눌러 인스턴스를 중지합니다.
5. 인스턴스의 상태가 정지됨인지 확인하고 창을 닫습니다.

주 – CLI 사용

CLI를 통해 서버 인스턴스를 중지하려면 다음 명령을 실행합니다.

```
wadm> stop-instance --user=admin --password-file=admin.pwd --host=serverhost
--port=8989 --config=config1 nodehost1
```

CLI 참조 `stop-instance`(1)를 참조하십시오.

서버 인스턴스 다시 시작

1. **노드 탭**을 눌러 서버에 구성된 노드 목록을 봅니다.
2. 노드 이름 확인란을 선택하여 노드를 선택합니다.
3. **인스턴스 다시 시작 버튼**을 눌러 페이지 창을 열고 해당 노드에서 제어하는 모든 인스턴스를 나열합니다.
4. 인스턴스를 선택하고 **인스턴스 다시 시작 버튼**을 눌러 인스턴스를 다시 시작합니다.
5. 인스턴스 상태가 실행중인지 확인하고 창을 닫습니다.

주 – CLI 사용

```
wadm> restart-instance --user=admin --password-file=admin.pwd
--host=serverhost --port=8989 --config=config1 nodehost1
```

CLI 참조 `restart-instance`(1)를 참조하십시오.

서버 인스턴스 다시 구성

구성을 변경할 때 인스턴스를 다시 시작할 필요가 없는 경우도 있습니다. Administration Server에서는 서버 인스턴스를 다시 구성하여 변경 사항을 구성 저장소로 가져오는 기능을 지원합니다. 이 구성에서는 서버를 다시 시작하지 않아도 변경 사항이 인스턴스에 반영됩니다. 구성에서 동적으로 다시 구성할 수 있는 변경 사항만 영향을 받습니다.

주 – user, temp-path, log, thread-pool, pkcs11, statistics, CGI, DNS, DNS-cache, file-cache, ACL-cache, SSL-session-cache, access-log-buffer 및 JVM(log-level 제외) 설정의 변경 사항은 다시 구성한 후에 적용되지 않습니다. 다시 시작해야 하는 이러한 변경은 다시 구성을 수행하면 기록됩니다. 파일 캐시를 다시 구성하려면 서버를 다시 시작해야 합니다.

1. **노드 탭**을 눌러 서버에 구성된 노드 목록을 봅니다.
2. 노드 이름 확인란을 선택하여 노드를 선택합니다.
3. **인스턴스 다시 구성 버튼**을 눌러 페이지 창을 열고 해당 노드에 배포된 모든 인스턴스를 나열합니다.
4. 인스턴스를 선택하고 **인스턴스 다시 구성 버튼**을 클릭하여 인스턴스를 다시 구성합니다.
5. 인스턴스 상태가 실행중인지 확인하고 창을 닫습니다.

주 – CLI 사용

CLI를 통해 서버 인스턴스를 다시 구성하려면 다음 명령을 실행합니다.

```
wadm> reconfig-instance --user=admin --password-file=admin.pwd
--host=serverhost --port=8989 --config=config1 serverhost
```

CLI 참조 `reconfig-instance`(1)를 참조하십시오.

서버 인스턴스 삭제

주 – 서버 인스턴스를 삭제하려면 실행 중이 아니어야 합니다.

1. **구성 탭**을 눌러 사용 가능한 구성 목록을 봅니다.
2. 구성 목록에서 구성을 선택합니다.
3. **인스턴스 하위 탭**을 누릅니다.
4. **노드 섹션** 아래에 있는 배포된 인스턴스 목록에서 인스턴스를 선택합니다.
5. 작업 드롭다운 목록에서 **인스턴스 삭제**를 선택하여 선택한 인스턴스를 삭제합니다.

주 – CLI 사용

CLI를 통해 서버 인스턴스를 삭제하려면 다음 명령을 실행합니다.

```
wadm> delete-instance --user=admin --password-file=admin.pwd --host=serverhost
--port=8989 --config=config1 serverhost
```

CLI 참조 `delete-instance`(1)를 참조하십시오.

인스턴스 자동 구성

예약된 이벤트를 기준으로 인스턴스를 다시 구성하거나 다시 시작할 수 있습니다. 자동 인스턴스 재구성 예약에 대해 특정 시간과 간격을 설정할 수 있습니다.

이벤트를 예약하려면 다음 작업을 수행하십시오.

1. **구성 탭**을 누르고 구성을 선택합니다.
2. **일반 하위 탭 > 예약된 이벤트 하위 탭**을 누릅니다.

▼ 예약된 이벤트 추가

1 구성을 선택합니다.

구성 탭을 누른 후에 표시되는 목록에서 구성을 선택합니다.

2 일반 > 예약된 이벤트 하위 탭을 누릅니다.

3 새로 만들기 버튼을 누릅니다.

4 다음 등록 정보를 구성합니다.

- **이벤트**

- **인스턴스 다시 시작** — 이 예약된 이벤트는 구성에 대해 배포되어 실행 중인 모든 인스턴스를 다시 시작합니다.
- **인스턴스 다시 구성** — 이 예약된 이벤트는 구성에 대해 배포되어 실행 중인 모든 인스턴스를 다시 구성합니다.
- **사용자 정의 명령줄** — 실행할 파일의 절대 경로를 제공합니다.

■ **일정**

이벤트를 시작하도록 구성된 시간입니다. 드롭다운 상자에서 시간 및 분 값을 선택합니다.

- **매일** — 지정된 이벤트를 매일 지정된 시간에 시작합니다.
- **특정 일** — 지정된 이벤트를 특정 날짜에 시작합니다.

  1. **요일** — 일요일부터 토요일까지의 요일을 지정합니다.

  2. **날짜** — 쉼표로 항목을 구분하여 1일부터 31일까지의 날짜를 지정합니다(예: 4,23,9).
- **특정 월** — 지정된 이벤트를 지정된 시간 및 월에 시작합니다. 1월부터 12월까지의 월을 지정합니다.
- **간격**

  지정된 이벤트를 이 기간 후에 시작합니다.

  1. *1***시간마다** — 드롭다운 상자에서 시간 단위를 선택합니다.

  2. *1***초마다** — 텍스트 필드에 초 값을 입력합니다.

주 – CLI 사용

CLI를 통해 이벤트를 예약하려면 다음 명령을 실행합니다.

```
wadm> create-event --user=admin --password-file=admin.pwd --host=serverhost
--port=8989 --config=config1 --time=10:10 --command=restart
```

CLI 참조 `create-event`(1)를 참조하십시오.

▼ 예약된 이벤트를 제거하는 방법

1 구성을 선택합니다.

구성 탭을 누른 후에 표시되는 목록에서 구성을 선택합니다.

2 일반 > 예약된 이벤트 하위 탭을 누릅니다.

3 예약된 이벤트를 선택하고 삭제 버튼을 누릅니다.

◆ ◆ ◆ 3 장

3 서버 팜 및 클러스터

이전 장에서는 구성을 소개하고 노드에 구성을 배포할 수 있는 방법에 대해 설명했습니다. 이 장에서는 간단한 서버 팜과 클러스터를 설정합니다.

Sun Java System Web Server에서 지원되는 클러스터

클러스터는 하나 이상의 노드에 걸쳐 모두 동일한 구성을 실행하고 동일한 런타임 서비스 집합을 제공하는 인스턴스 집합입니다. 각 클러스터에는 Administration Server로 지정된 서버가 하나 있어야 합니다. 클러스터가 두 개 이상인 경우 하나의 마스터 Administration Server에서 모든 클러스터를 관리할 수 있습니다. 마스터 Administration Server는 모든 클러스터의 정보를 검색하고 각 클러스터에 설치된 Sun Java System Web Server를 관리할 수 있는 인터페이스를 제공합니다.

주 – 클러스터에 있는 모든 인스턴스는 동일한 유형이어야 합니다. 예를 들면 동일한 운영 체제(와 패치) 및 서비스 팩을 실행하고, 동일한 웹 서버 구성을 실행하며, 동일한 서비스를 제공해야 합니다.

서버 팜 설정

클러스터를 설정하려면 먼저 한 개의 Administration Server와 하나 이상의 관리 노드를 설치해야 합니다. 관리 노드를 관리하려면 Administration Server에 개별적으로 등록해야 합니다. 이 작업은 노드를 설치하는 동안 수행할 수도 있고 설치 후에 `wadm` CLI를 통해 수행할 수도 있습니다.

▼ 서버 팜을 설정하는 방법

1 Administration Server 및 관리 노드 설치

Administration Server를 설치합니다. Sun Java System Web Server 설치 프로그램 GUI 또는 `wadm` CLI를 통해 Administration Server를 설치할 수 있습니다.

Administration Server를 포트 8989에 설치하는 **Express 설치** 옵션을 선택할 수 있습니다. 또는 **사용자 정의 설치** 옵션을 선택하여 기본 설정을 지정할 수도 있습니다. Administration Server를 설치하려면 설치 프로그램 설정 화면에서 **서버를 Administration Server로 설치** 옵션을 선택합니다. SSL 포트는 지정해야 하지만 비 SSL 포트는 지정할 수도 있고 지정하지 않을 수도 있습니다.

주 – 비 SSL 포트를 지정하는 경우 관리 노드가 Administration Server 노드에 만들어지며 이 노드는 Administration Server에 명시적으로 등록할 필요가 없습니다.

관리 노드를 설치하려면 **사용자 정의 설치**를 선택한 다음 **서버를 관리 노드로 설치**를 선택합니다. 설치에 사용할 포트를 지정합니다. Administration Server와 관리 노드 사이의 모든 통신이 보안 채널을 통해 이루어지기 때문에 비 SSL 포트는 선택할 수 없습니다. 설치 중에 Administration Server에 노드를 등록해야 하는지 묻는 메시지가 표시됩니다. 설치 중에 노드를 등록하지 않는 옵션을 선택하는 경우 `wadm` CLI를 사용하여 이 작업을 수행할 수 있습니다.

주 – Express 설치를 통해서는 관리 노드를 설치할 수 없습니다.

---

2 Administration Server에 관리 노드 등록

관리 노드를 클러스터 또는 서버 팜에 포함시키면 먼저 Administration Server에 등록해야 합니다. 관리 노드는 Administration Server에 등록하지 않으면 시작되지 않습니다. 관리 노드를 등록하려면 `wadm` CLI를 통해 다음 명령을 실행합니다.

```
wadm> register-node --user <admin-user> --port <SSL Port> --host <node name>
```

여기서 port는 Administration Server를 설치하는 동안 지정되는 포트입니다. host는 Administration Server가 설치된 노드의 호스트 이름입니다.

이 작업은 노드를 Administration Server에 등록합니다.

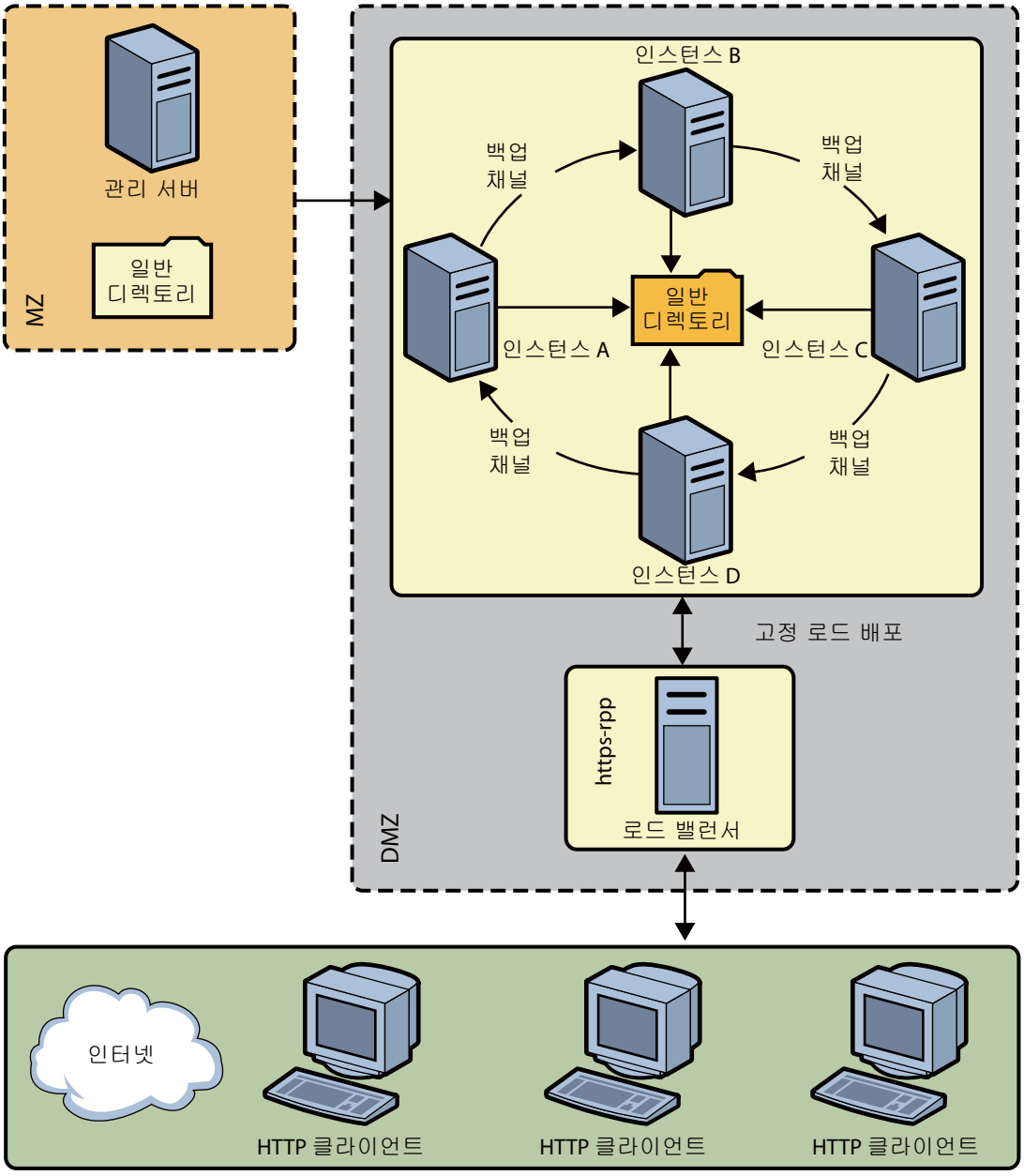
주 – 노드는 같은 노드에서만 등록할 수 있습니다. Administration Server의 CLI로 이동하여 아무 노드나 등록할 수는 없습니다. 또한 Administration Server에 노드를 등록하는 경우 SSL 모드로만 가능합니다.

작성된 클러스터에 세션 복제를 설정하려면 176 페이지 “세션 복제 구성”을 참조하십시오.

단순 클러스터 설정

이 예의 일부로 로드 밸런서 1개, Administration Server 1개, 세션 복제가 사용 가능한 웹 서버 인스턴스 4개가 있는 클러스터를 설정합니다. 세션 복제는 Java 웹 응용 프로그램 세션에 고가용성을 제공합니다. 이렇게 하려면 한 웹 서버 인스턴스의 메모리에 상주하는 세션을 다른 웹 서버 인스턴스에 복사합니다. 따라서 정상적인 작업 조건에서는 모든 세션에 적어도 2개의 복사본이 있으며, 각각 별도의 JVM에(최적 상황이라면 별도의 시스템에) 상주합니다.

다음 그림은 단순한 클러스터를 나타냅니다.


MZ
관리 서버
일반
디렉토리
DMZ
인스턴스 B
백업
채널
백업
채널
일반
디렉토리
인스턴스 A
인스턴스 C
백업
채널
백업
채널
인스턴스 D
고정 로드 배포
https-rpp
로드 밸런서
인터넷
HTTP 클라이언트
HTTP 클라이언트
HTTP 클라이언트

▼ 클러스터를 구성하는 방법

시작하기 전에 다음 시스템을 확인합니다.

- MachineA — 로드 밸런서와 Administration Server가 모두 있습니다.
- MachineB, MachineC, MachineD 및 MachineE — 관리 노드와 웹 서버 인스턴스가 실행되고 있습니다.

1 MachineA에 Administration Server를 설치합니다.

Administration Server 설치에 대한 자세한 내용은 46 페이지 "서버 팜을 설정하는 방법"을 참조하십시오. 일반적인 설치 프로세스에서는 웹 서버 인스턴스도 설치합니다. 이 시나리오에서는 해당 인스턴스를 사용하지 않습니다.

2 MachineB, MachineC, MachineD 및 MachineE에 관리 노드를 설치합니다.

4개의 시스템 모두에 관리 노드를 설치합니다. Administration Server에 관리 노드를 등록합니다.

3 웹 응용 프로그램을 구성합니다.

웹 응용 프로그램의 세션 복제를 활성화합니다. `WEb-INF/sun-web.xml` 파일을 다음과 같이 수정합니다.

```
<session-manager persistence-type="replicated"/>
```

4 인스턴스를 구성합니다.

- `wadm`을 시작합니다.

  ```
  wadm --host MachineA --port 8089
  ```

- 로드 밸런서에 대해 새 구성을 만듭니다.

  ```
  wadm> create-config --http-port=8080 --server-name=SampleCluster lb
  ```

- 역방향 프록시(로드 밸런서)를 설정합니다.

  ```
  wadm> create-reverse-proxy --config=lb --vs=lb
  -uri-prefix=/ --server="http://MachineB:8080,http://MachineC:8080,
  ttp://MachineD:8080,http://MachineE:8080"
  ```

- 인스턴스를 만듭니다.

  ```
  wadm> create-instance --config=lb MachineA
  ```

- 구성을 배포합니다.

  ```
  wadm> deploy-config lb
  wadm> start-instance --config=lb
  ```

5 클러스터를 만들고 시작합니다.

4개의 인스턴스에 새 구성을 만듭니다.

- 클러스터의 새 구성을 만듭니다.

```
wadm> create-config --http-port=8080 --server-name=SampleCluster clusterOf4
```

- 세션 복제를 활성화합니다.

```
wadm> set-session-replication-prop --config=clusterOf4 enabled=true
```

- 웹 응용 프로그램을 추가합니다.

```
wadm> add-webapp --config=clusterOf4 --uri=/simple webapps-simple.war
```

- 인스턴스를 만듭니다.

```
wadm> create-instance --config=clusterOf4 MachineB MachineC MachineD MachineE
```

- 클러스터를 시작합니다.

```
wadm> start-instance --config=clusterOf4
```

주 – start-instance 명령에 호스트 이름을 지정하지 않으면 이 작업은 구성이 배포된 모든 노드에서 인스턴스를 시작합니다.

◆ ◆ ◆ 4 장

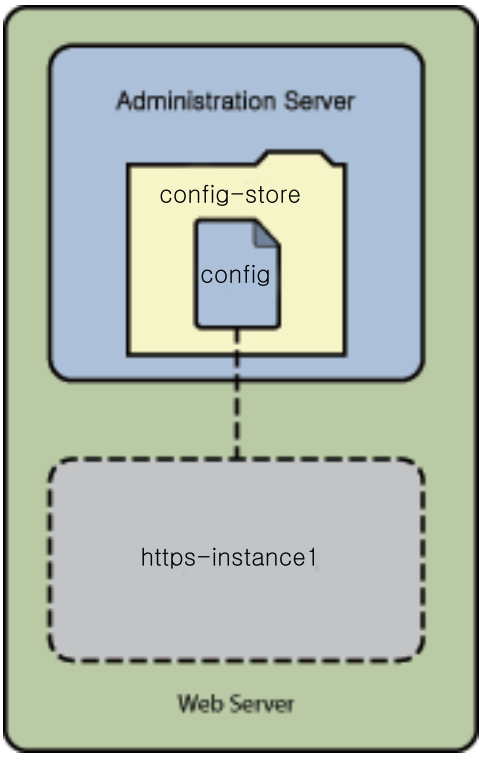
4 배포 시나리오

이 장에서는 Sun Java System Web Server 7.0을 단일 노드와 클러스터 환경에 배포하는 방법에 대해 설명합니다. 이 장에서는 다음 항목을 다룹니다.

배포 구조

이 절에서는 단일 노드 배포 구조에 대해 설명합니다.

다음 그림은 단일 노드 배포 설정의 Web Server를 나타냅니다.



그림에서 Web Server 배포 설정은 다음 구성 요소로 이루어집니다.

- *Administration Server*- Administration Server는 특별히 구성된 웹 서버 인스턴스입니다. Administration Server에 웹 응용 프로그램을 배포할 수 있습니다.
- **관리 노드**- 관리 노드는 서버 팜에 있는 노드 또는 서버/호스트에 배포되며 원격 Administration Server와 통신할 수 있습니다. Administration Server 내에서 사용할 수 있는 서버 구성은 이 노드에 배포할 수 있습니다. 서버 팜에 있는 모든 관리 노드는 동종이어야 합니다. 즉, 모든 노드는 동일한 운영 체제를 사용하며 하드웨어 구조가 동일해야 합니다.
- **구성**- 구성은 웹 응용 프로그램, 구성 파일, 검색 모음 색인과 같이 Web Server 인스턴스에서 구성 가능한 모든 요소의 집합을 나타냅니다. 구성은 작성, 수정 또는 삭제할 수 있습니다. Web Server는 구성에 대해 만들 수 있는 여러 구성 인스턴스를 관리할 수 있습니다. 수정된 구성을 배포하면 해당 구성의 인스턴스가 업데이트됩니다.
- *config-store*. 모든 구성이 저장되는 파일 시스템 기반의 저장소입니다.

주의 – `config-store` 디렉토리에 있는 파일을 편집하지 마십시오. 이 디렉토리에 있는 파일은 Sun Java System Web Server에서 내부용으로 만든 파일입니다.

`config-store` 디렉토리에 있는 구성 파일을 수동으로 편집해야 하는 경우에는 `wadm deploy-config` 명령을 사용하여 구성을 배포합니다.

이 명령의 사용에 대한 자세한 내용은 **Sun Java System Web Server 7.0 CLI Reference Manual**을 참조하십시오.

- **인스턴스** - 인스턴스는 지정된 노드에서 웹 서버 환경을 나타내며, 해당 구성 및 로그 파일을 비롯하여 잠금 데이터베이스, 캐시, 임시 파일 등과 같은 다른 런타임 아티팩트를 포함합니다. 관리 용도로 인스턴스를 시작, 중지, 다시 시작하거나 동적으로 다시 구성할 수 있습니다.

배포 개요

다음과 같은 목적이 있는 경우 단일 노드에 Web Server를 배포하는 것을 고려할 수 있습니다.

- 간단한 웹 또는 CGI 응용 프로그램 호스팅
- 웹 응용 프로그램 개발 및 테스트

다음 순서도는 Web Server를 노드에 배포하는 방법을 간략하게 표현한 것입니다.

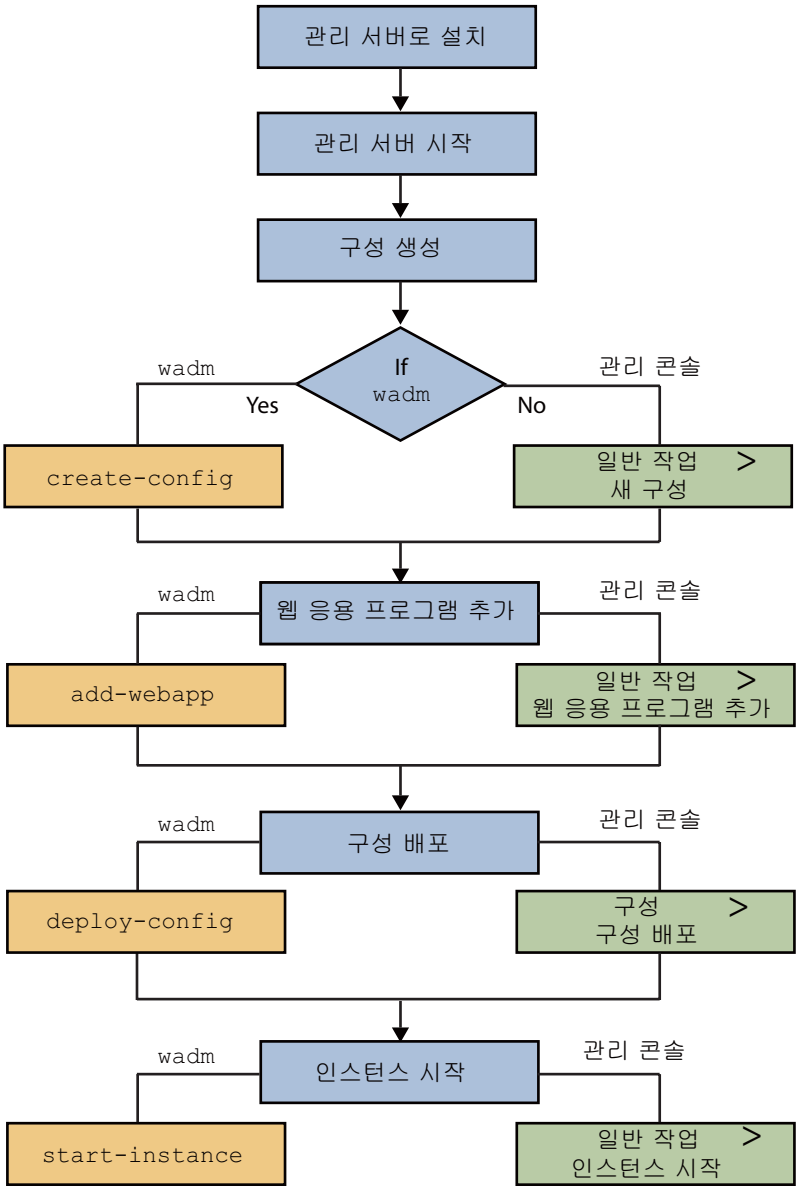


그림 4–1 단일 노드에서 웹 서버 배포를 나타내는 순서도

배포 프로세스에 대해서는 다음 절에서 설명합니다.

- 55 페이지 “배포 전 요구 사항”
- 55 페이지 “Web Server 배포”

배포 전 요구 사항

단일 노드에 Web Server를 배포하려면 다음 작업을 수행하여 시스템을 준비하십시오.

1. 노드에 Web Server를 설치합니다.

   Web Server를 설치하면서 Express 설치 옵션을 선택한 경우에는 다음 기본 항목이 작성됩니다.

   - Administration Server
   - HTTP Listener 하나와 가상 서버가 만들어진 기본 구성. 구성과 가상 서버의 이름은 호스트 이름과 같습니다.
   - 기본 구성의 인스턴스

   Web Server 설치에 대한 자세한 내용은 **Sun Java System Web Server 7.0 Installation and Migration Guide**의 2 장, “Installing the Web Server”를 참조하십시오.

   지원되는 플랫폼 및 시스템 요구 사항에 대한 자세한 내용은 **Sun Java System Web Server 7.0 릴리스 노트**의 “지원되는 플랫폼”을 참조하십시오.
2. Administration Server를 시작합니다.

   Administration Server가 지정된 SSL 포트에서 실행되기 시작합니다.

Web Server 배포

다음 절차를 수행하여 노드에 Web Server를 배포합니다.

1. 기본 구성을 사용할 수도 있고 새 구성을 만들 수도 있습니다.

   새 구성을 만드는 경우에는 구성에 고유한 이름을 지정합니다. 새 구성은 가상 서버와 기본 HTTP Listener를 만듭니다.

   주 – 관리 콘솔을 사용하여 구성을 만드는 경우 마법사에서 새 인스턴스를 만들라는 내용의 프롬프트를 표시합니다. CLI를 사용하는 경우에는 `create-instance` 명령을 사용하여 구성의 인스턴스를 명시적으로 만들어야 합니다.

   모든 구성은 <*install_dir*>`/admin-server/` 디렉토리 아래에 있는 `config-store` 디렉토리에 저장됩니다.

   주의 – `config-store` 디렉토리에 있는 파일을 편집하지 마십시오. 이 디렉토리에 있는 파일은 Sun Java System Web Server에서 내부용으로 만든 파일입니다.

   ---
2. 수정된 구성을 배포합니다.

클러스터 환경

클러스터는 두 개 이상의 노드에 걸쳐 있는 여러 서버 인스턴스의 그룹이며, 모두 동일한 구성으로 실행됩니다. 클러스터에 있는 모든 인스턴스는 함께 작동하며 고가용성, 안정성, 확장 가능성을 제공합니다.

클러스터는 로드 균형 조정을 사용하여 페일오버 및 세션 복제를 수행하며 인터럽트 없는 서비스 및 세션 데이터의 지속성을 제공합니다.

하드웨어 및 소프트웨어 요구 사항

이 절에서 설명하는 사용자 사례에서 Web Server 클러스터는 다음 항목으로 구성됩니다.

1) 4개의 인스턴스(동일한 4개의 노드에서 실행)
2) Administration Server
3) HTTP 요청의 로드 균형 조정을 위한 역방향 프록시

클러스터를 설정하려면 동일한 운영 체제 버전과 패치가 설치된 두 개 이상의 동일한 노드가 필요합니다. 예를 들어 Solaris® 9 SPARC® 운영 체제가 설치된 시스템을 선택한 경우 클러스터에 있는 다른 시스템에도 Solaris 9 SPARC가 설치되어 있어야 합니다.

지원되는 플랫폼 및 패치 요구 사항에 대한 자세한 내용은 **Sun Java System Web Server 7.0 릴리스 노트**를 참조하십시오.

다음 그림은 클러스터 환경을 나타냅니다.

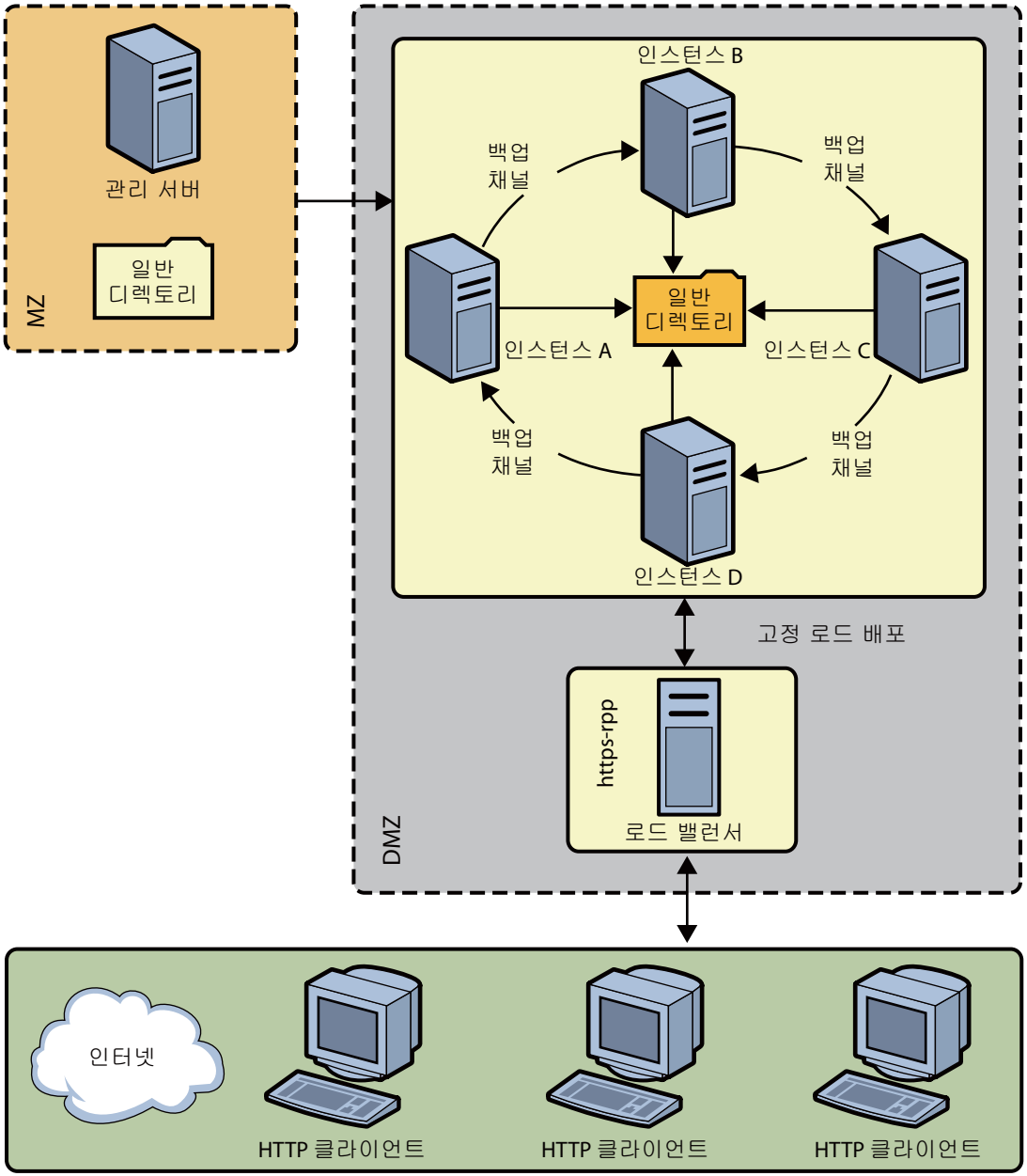


그림 4–2 클러스터 설정

그림에서 노드는 DMZ(De-Militarized Zone)에 구성되어 있습니다. Administration Server는 방화벽 뒤의 Militarized Zone에 구성되며 Administration Server에 대한 일반

액세스를 제한하고 보호합니다. 다른 노드가 역방향 프록시 서버로 구성됩니다. 역방향 프록시 서버는 보안을 강화하기 위해 DMZ 내부에 상주합니다.

주 – Solaris 영역 기능은 Solaris 10 운영 체제에서만 지원됩니다.

클러스터 설정

이 절에서는 클러스터를 설정하고 역방향 프록시를 사용 가능하게 설정하여 HTTP 요청의 로드 균형 조정을 지원하는 방법에 대해 설명합니다.

다음 순서도는 클러스터를 설정하는 절차를 나타냅니다.

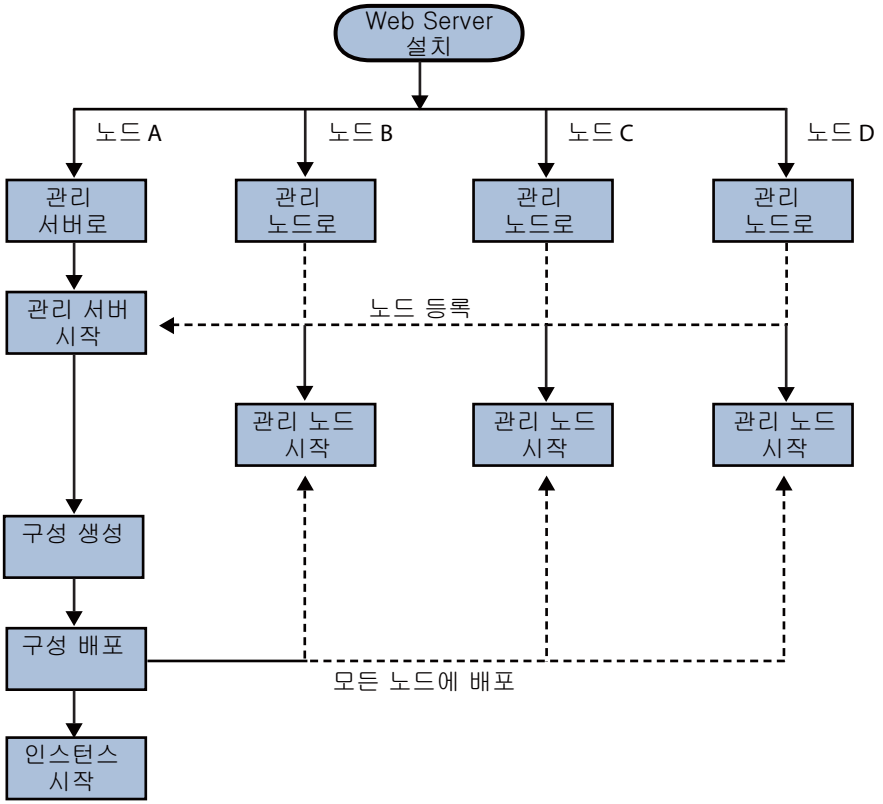


그림 4–3 클러스터 설정을 나타내는 순서도

1. 노드 중 하나에 클러스터의 Administration Server 역할을 하는 Web Server를 설치합니다.

2. 다른 세 개의 노드에 Web Server를 설치합니다. Web Server를 관리 노드로 설치하는 옵션을 선택합니다. 설치 중에 노드를 서버에 등록하는 옵션을 선택합니다.
3. 관리 노드는 보안 모드에서만 서버에 등록할 수 있기 때문에 Administration Server의 통신에는 SSL 포트만 사용해야 합니다.
4. Administration Server와 관리 노드가 설치되어 있는 모든 노드의 시스템 날짜와 시간은 같아야 합니다. 서버와 연결된 인증서는 Administration Server가 설치된 노드의 시스템 날짜와 시간을 기반으로 작성됩니다. 관리 노드의 시스템 날짜가 Administration Server보다 이전이면 Administration Server의 인증서가 유효하지 않기 때문에 등록이 실패합니다. 따라서 인증서가 만료된 경우에도 유효한 것으로 처리될 수 있습니다.
5. *install_dir*`/admin-server/bin/` 디렉토리에서 Administration Server를 시작합니다.

   install_dir`/admin-server/bin>./startserv`
6. 관리 노드에서 `wadm` 명령줄 도구를 시작합니다. `wadm` 명령줄 도구는 *install_dir*`/bin` 디렉토리에 있습니다.

   install_dir`/bin>./wadm`
7. 각 관리 노드를 Administration Server에 등록합니다. `register-node` 명령을 사용하여 각 노드를 서버에 등록합니다.

   예:

   ```
   ./wadm register-node -user=admin --host=abc.sfbay.sun.com --port=8989
   ```

   여기서

   `abc.sfbay.sun.com` 노드를 등록할 Administration Server의 호스트 이름입니다.

   `port` Administration Server의 SSL 포트 번호입니다.
8. 관리 비밀번호를 입력하는 프롬프트가 표시됩니다. Administration Server의 관리 비밀번호를 입력합니다.

   Administration Server와 관리 노드는 Administration Server에서 관리 노드의 서버 인증서를 신뢰하고 관리 노드에서 Administration Server가 제공하는 클라이언트 인증서를 신뢰하는 방식으로 서로를 인증합니다. 관리 노드를 등록하는 동안 Administration Server가 해당 관리 노드의 서버 인증서를 생성합니다. 그러면 이 인증서가 다운로드되고 관리 노드에 설치됩니다. 서버 인증서 발급자도 관리 노드에 설치됩니다.

주 – 등록은 SSL을 통해서만 수행될 수 있습니다.

노드 등록에 대한 자세한 내용은 **Sun Java System Web Server 7.0 Installation and Migration Guide**의 “Registering the Administration Node From the Command-Line”을 참조하십시오.

9. *install_dir*/admin-server/bin/ 디렉토리에서 startserv 명령을 사용하여 모든 관리 노드를 시작합니다.
10. 관리 콘솔 또는 CLI를 사용하여 Administration Server에 새 구성을 만듭니다.

    구성 이름, HTTP Listener 포트 및 서버 이름과 같은 구성 정보를 새 구성에 지정합니다.
11. 모든 노드에 구성 인스턴스를 만듭니다.
12. 모든 노드에서 인스턴스를 시작합니다.

주 – Web Server에는 클러스터를 확장 또는 축소할 수 있는 유연성이 있습니다. 언제든지 인스턴스를 클러스터에 추가하거나 클러스터에서 제거할 수 있습니다.

로드 균형 조정을 위한 역방향 프록시 구성

Web Server 7.0은 정교한 내장 로드 밸런서인 역방향 프록시를 제공합니다. 역방향 프록시는 서버 팜에 있는 Web Server의 게이트웨이입니다. 역방향 프록시를 구성하면 비슷하게 구성된 여러 개의 웹 서버로 요청이 전달됩니다.

다음 절차에 따라 Web Server 7.0에서 역방향 프록시를 사용 가능하게 설정합니다.

1. 역방향 프록시 구성에 사용할 노드에 Web Server를 설치합니다.
2. 구성을 만듭니다. 예를 들면 rp와 같습니다.
3. 관리 콘솔을 사용하여 구성 > 가상 서버 > 내용 처리 > 역방향 프록시 탭을 선택합니다. 새로 만들기 버튼을 누릅니다.
4. 역방향 프록시 URI와 클러스터에 있는 모든 시스템의 서버 URL을 쉼표로 구분하여 입력합니다.

   서버 URL은 *hostname*:*portnumber* 형식으로 입력합니다.
5. 변경 사항을 저장합니다.
6. 수정된 구성을 배포하여 변경 사항을 구성에 적용합니다.
7. 이렇게 수정된 구성의 모든 인스턴스를 시작합니다.

그러면 HTTP 요청의 로드 균형 조정을 위한 역방향 프록시 구성이 완료됩니다.

세션 복제

세션 복제는 세션에 저장된 데이터를 다른 인스턴스에 복제하는 데 사용되는 기법입니다. 하지만 복제된 인스턴스는 동일한 클러스터의 일부여야 합니다. 클러스터 환경에서 세션 복제가 사용 가능한 경우에는 전체 세션 데이터가 복제된 인스턴스에 복사됩니다. 하지만 세션 복제 작업은 세션에 있는 일련화할 수 없는 속성과 인스턴스별 데이터를 복사하지 않습니다.

세션 복제는 로드 균형 조정과 함께 웹 응용 프로그램에 페일오버 기능을 제공합니다.

세션 복제 및 페일오버 작업

이 절에서는 세션 복제 작업에 대해 자세히 설명합니다.

웹 요청이 끝날 때 Web Server는 서버 구성 파일 `server.xml`에 저장된 세션 복제 구성을 통해 세션 데이터를 복사해야 하는지 여부를 결정합니다.

4개의 인스턴스가 클러스터를 이루며 Administration Server에서 세션 복제가 사용 가능한 경우를 생각할 수 있습니다.

4개의 노드에서 실행되는 4개의 인스턴스(A, B, C 및 D)로 이루어진 Web Server 클러스터에서 세션 복제 프로세스는 다음과 같습니다.

- 인스턴스 A는 D의 백업이고, B는 A의 백업이고, C는 B의 백업이고, D는 C의 백업입니다. 이런 식으로 완전한 백업 링을 형성합니다.
- 클러스터의 각 인스턴스는 클러스터에 있는 모든 인스턴스의 정적 목록과 활성 백업 인스턴스를 추적합니다.
- 구성에 따라 각 요청이 끝날 때 세션 데이터가 백업 인스턴스에 동기적으로 전송됩니다.

Web Server 클러스터 환경에서 페일오버 프로세스는 다음과 같습니다.

- 인스턴스 A가 실패하면 로드 밸런서는 인스턴스 A를 향해 들어오는 모든 웹 요청을 클러스터에 있는 나머지 인스턴스로 리디렉션하고 백업 링을 다음과 같이 다시 구성합니다.
  - D에서는 백업 A가 중단된 것을 감지하고 순서 목록에서 A 다음에 있는 인스턴스를 새 백업 인스턴스로 선택합니다.
  - 이 경우에는 B가 선택되며, D는 B에 대한 새 백업 연결을 구성합니다. 이제 B에는 읽기 전용 백업 A와 활성 백업 D의 두 가지 백업이 있습니다.
  - 이제 B가 C에 백업되고, C가 D에 백업되고, D가 B에 백업되어 백업 링이 완료됩니다.
  - 실패한 인스턴스 A를 다시 사용할 수 있게 되면 지정된 백업 인스턴스 B에 다시 결합 메시지를 보내 백업 링과 다시 결합하고 B에 대한 백업 연결을 설정합니다.
  - D는 A로부터 성공적인 핑 반환이나 메시지를 받고 A가 온라인 상태임을 감지합니다.
  - 그런 다음 D는 A에 대한 백업 연결을 설정하고 B에 대한 백업 연결을 종료합니다.

Web Server 7.0은 세션 복제에서 다음 기능을 지원하지 않습니다.

- 두 개 이상의 인스턴스에서 동시에 장애 복구

- 두 장애 사이의 간격은 회복된 인스턴스를 완전히 복구하는 데 필요한 시간보다 커야 합니다.
- 두 개 이상의 인스턴스에 대한 세션 백업. 정상적으로 작동될 경우에는 모든 세션에 대해 기본 세션과 백업 세션이라는 두 개의 복사본만 있습니다.
- 세션 지속성: 세션은 페일오버를 위해 다른 인스턴스의 메모리에만 백업됩니다.
- Web Server는 Java 웹 응용 프로그램의 세션 복제만 지원합니다. CGI 또는 PHP와 같은 비 Java 응용 프로그램을 사용하는 경우에는 세션 데이터를 복제할 수 없습니다.

세션 복제 활성화

관리 콘솔이나 CLI를 사용하여 클러스터에서 세션 복제를 활성화할 수 있습니다. 세션 복제를 활성화하려면 브라우저에서 쿠기가 사용 가능해야 합니다.

`server.xml` 파일에는 세션 복제와 관련된 정보가 있습니다. 다음은 세션 복제가 활성화된 샘플 `server.xml` 파일입니다.

```
<cluster>
        <local-host>hostA</local-host>
            <instance>
         <host>hostB</host>
      </instance>
      <instance>
          <host>hostC</host>
      </instance>
       <instance>
          <host>hostD</host>
      </instance>
      <instance>
           <host>hostA</host>
    <session-replication/>
</cluster>
```

다음 요소의 기본값을 사용하지 않는 경우에는 `server.xml` 구성 파일에서 이러한 요소의 항목을 사용할 수 없습니다.

`Port number`(기본값은 1099)
`Protocol`(기본값은 `jrmp`)
`Encrypted`(기본값은 false)
`Getattribute Triggers Replication`(기본값은 true)
`Replica Discovery MaxHops`(기본값은 –1)
`Startup Discovery Timeout`(기본값은 ?)
`Cookie Name`(기본값은 `CLUSTERSESSIONLOCATOR`)

이러한 세션 복제 등록 정보에 대한 자세한 내용은 **Sun Java System Web Server 7.0 Administrator's Configuration File Reference**를 참조하십시오.

세션 복제를 위한 웹 응용 프로그램 구성

서버에서 세션을 복제할 수 있게 하려면 웹 응용 프로그램도 세션 복제용으로 활성화해야 합니다.

1. 웹 응용 프로그램의 세션 복제를 활성화하려면 <*web-application*>/WEB-INF 디렉토리에 있는 sun-web.xml 구성 파일을 수정합니다.

   sunweb.xml에서 수정해야 할 사항은 다음과 같습니다.

   <session-manager/> 요소를 <session-manager persistence-type="replicated">로 변경합니다.

   다음은 세션 복제가 활성화된 샘플 sun-web.xml 파일입니다.

   ```
   <sun-web-app>
           <session-config>
        <session-manager persistence-type="replicated">
        </session-manager>
     </session-config>
   </sun-web-app>
   ```

2. sunweb.xml 파일을 수정한 후 웹 응용 프로그램을 다시 작성하거나 응용 프로그램을 JAR 파일로 다시 만들어 웹 응용 프로그램 아카이브(war 파일)를 만듭니다.
3. 모든 인스턴스를 다시 시작하여 모든 인스턴스에서 웹 응용 프로그램을 사용할 수 있게 합니다.
4. 웹 응용 프로그램은 클러스터의 모든 노드에서 액세스할 수 있습니다. 웹 응용 프로그램에 액세스하려면 브라우저에서 다음을 입력합니다.

   http://*webserver-name*/*webapplication-name*/

주 – 모든 노드에 액세스할 수 있는 디렉토리를 사용하는 것이 배포할 응용 프로그램을 저장하는 가장 좋은 방법입니다. 하지만 이 디렉토리에서 Administration Server에 액세스할 수 있어야 하는 것은 아닙니다. 크기가 1MB를 넘는 웹 응용 프로그램의 경우에는 디렉토리 기반 배포를 사용하는 것이 좋습니다.

검색 모음을 만드는 경우 모든 노드에 액세스할 수 있는 일반 디렉토리에 검색 모음이 있어야 합니다.

클러스터 모니터링

Administration Server에서는 클러스터에 있는 모든 인스턴스를 모니터할 수 있습니다. Web Server의 모니터링 기능은 다음 작업에 사용할 수 있는 런타임 구성 요소와 프로세스의 상태에 대한 정보를 제공합니다.

- 성능 병목 현상 식별
- 최적의 성능을 얻을 수 있도록 시스템 조정
- 용량 계획 지원
- 장애 예상
- 장애가 발생한 경우 근본 원인 분석 수행

Solaris 영역

Solaris 영역은 Solaris 10의 응용 프로그램 및 자원 관리 기능입니다. 영역 환경은 일반적으로 프로세스 관리, 메모리, 네트워크 구성, 파일 시스템, 패키지 레지스트리, 사용자 계정, 공유 라이브러리 및 일부 경우에는 설치된 응용 프로그램 등의 자원으로 구성됩니다. 영역을 사용하면 Solaris 인스턴스 내에서 가상화된 운영 체제 환경을 만들 수 있으므로 하나 이상의 프로세스를 시스템의 다른 작동과 격리하여 실행할 수 있습니다. 여기에는 물리적 장치 경로 및 네트워크 인터페이스 이름, 네트워크 라우팅 테이블과 같이 응용 프로그램이 배포된 시스템의 물리적 속성에서 응용 프로그램을 분리하는 추상화 계층도 제공됩니다. 이러한 격리를 통해 사용자 아이디와 기타 자격 증명 정보에 관계없이 지정된 영역에서 실행되는 프로세스가 다른 영역에서 실행되는 프로세스를 모니터하거나 영향을 주는 것을 방지할 수 있습니다.

영역은 하나 이상의 응용 프로그램이 시스템의 나머지 부분에 영향을 주거나 상호 작용하지 않고 실행될 수 있는 **샌드 박스**입니다.

Solaris 영역에 대한 자세한 내용은 `http://docs.sun.com/app/docs/doc/817-1592`에 있는 *System Administration Guide — Solaris Containers-Resource Management and Solaris Zones*를 참조하십시오.

◆ ◆ ◆ 5 장

5 가상 서버 사용

가상 서버 개요

가상 서버를 사용하면 설치된 단일 서버에서 회사 또는 개별 도메인 이름, IP 주소 및 일부 서버 모니터링 기능을 제공할 수 있습니다. 하드웨어와 기본적인 웹 서버 유지 관리가 제공될 뿐이지만 사용자는 거의 자신의 전용 웹 서버를 가지고 있는 것과 같습니다.

모든 가상 서버에는 HTTP Listener가 지정되어 있습니다. 새 요청이 들어오면 서버는 구성된 HTTP Listener를 기준으로 요청을 보낼 가상 서버를 결정합니다.

사용 사례

Sun Java System Web Server의 서버 인스턴스에 지정할 수 있는 보안 및 비보안 HTTP Listener의 수에는 제한이 없습니다. IP 주소 기반 및 URL 호스트 기반 가상 서버를 모두 사용할 수 있습니다.

모든 가상 서버는 자체 ACL 목록, 자체 `mime.types` 파일, 자체 Java Web Applications 세트를 가질 수 있습니다(반드시 그렇지는 않음).

이러한 설계의 경우 다양한 응용 프로그램에 대해 서버를 구성할 수 있도록 유연성을 최대한으로 제공합니다. 다음 예에서는 Sun Java System Web Server에서 사용할 수 있는 몇 가지 구성에 대해 설명합니다.

기본 구성

Sun Java System Web Server를 새로 설치하면 서버 인스턴스가 하나 생깁니다. 이 서버 인스턴스에는 포트 80(또는 설치할 때 선택한 포트)에서 컴퓨터가 구성된 모든 IP 주소를 수신하는 HTTP Listener 이더넷이 하나만 있습니다.

로컬 네트워크에는 컴퓨터가 구성된 각 주소에 대해 이름과 주소 매핑을 설정하는 몇 가지 기법이 있습니다. 다음 예의 컴퓨터에는 주소 127.0.0.1의 루프백 인터페이스(네트워크 카드가 없어도 존재하는 인터페이스)와 주소 10.0.0.1의 이더넷 인터페이스가 있습니다.

example.com이라는 이름이 DNS를 통해 10.0.0.1에 매핑됩니다. 수신 소켓은 시스템이 구성된 모든 주소를 포트 80에서 수신하도록 구성됩니다("ANY:80" 또는 "0.0.0.0:80").

이 구성에서는 다음에 대한 연결이 서버에 도달하며 가상 서버 VS1에 의해 서비스됩니다.

- `http://127.0.0.1/`(`example.com`에서 시작)
- `http://localhost/`(`example.com`에서 시작)
- `http://example.com/`
- `http://10.0.0.1/`

기존 웹 서버를 사용하는 경우 이 구성을 사용합니다. 추가 가상 서버나 HTTP Listener를 추가할 필요는 없습니다.

보안 서버

83 페이지 “서버의 SSL 구성”을 참조하십시오.

인트라넷 호스팅

더 복잡한 Sun Java System Web Server 구성에서는 서버가 인트라넷 배포에 사용할 가상 서버 몇 개를 더 호스팅합니다. 예를 들어 직원들이 다른 사용자의 전화 번호를 조회하고, 구내 지도를 보고, 정보 협력 부서에 대한 요청 상태를 추적할 수 있는 3개의 내부 사이트가 있습니다. 이 예의 앞부분에서 이러한 사이트는 `phone.example.com`, `maps.example.com` 및 `is.example.com`이 매핑된 3개의 다른 컴퓨터에서 호스팅되었습니다.

하드웨어 및 관리 오버헤드를 최소화하기 위해 시스템 `example.com`에 있는 웹 서버 하나에 3개 사이트 모두를 통합할 수 있습니다. URL 호스트 기반 가상 서버 사용 또는 별도의 HTTP Listener를 사용하는 두 가지 방법을 사용하여 이러한 통합을 설정할 수 있습니다. 두 가지 방법에는 각각 장단점이 있습니다.

URL 호스트 기반 가상 서버를 사용한 인트라넷 호스팅

URL 호스트 기반 가상 서버는 설정하기 쉽지만 다음과 같은 단점이 있습니다.

- 이 구성에서 SSL을 지원하려면 와일드카드 인증서를 사용하는 비표준 설정이 필요합니다.
- URL 호스트 기반 가상 서버는 기존 HTTP 클라이언트에서 작동하지 않습니다.

주소별로 한 개의 HTTP Listener를 사용하여 IP 주소 기반 구성을 설정할 수도 있습니다.

별도의 HTTP Listener를 사용한 인트라넷 호스팅

IP 주소 기반 가상 서버를 사용하는 경우의 장점은 다음과 같습니다.

- HTTP/1.1 Host 헤더를 지원하지 않는 이전 클라이언트에서 작동합니다.
- SSL 지원 제공이 간단합니다.

다음과 같은 단점이 있습니다.

- 호스트 컴퓨터의 구성을 변경해야 합니다(실제 또는 가상 네트워크 인터페이스 구성).
- 수천 개의 가상 서버가 있는 구성으로 확장되지 않습니다.

두 가지 구성을 위해서는 3개의 이름에 대해 이름 대 주소 매핑을 설정해야 합니다. IP주소 기반 구성에서 각 이름은 서로 다른 주소에 매핑됩니다. 호스트 시스템이 이러한 모든 주소에서 연결을 수신하도록 설정해야 합니다. URL 호스트 기반 구성에서 모든 이름은 시스템이 원래 가졌던 동일한 주소에 매핑될 수 있습니다.

HTTP Listener가 여러 개인 구성에서는 서버가 요청이 들어온 주소를 찾을 필요가 없기 때문에 최소한의 성능 향상이 있을 수 있습니다. 그러나 여러 개의 HTTP Listener를 사용하면 추가 억셉터 스레드 때문에 추가 오버헤드(메모리 및 일정 계획)가 발생하기도 합니다.

대량 호스팅

대량 호스팅은 트래픽이 낮은 여러 가상 서버를 사용하는 구성입니다. 예를 들어 트래픽이 낮은 여러 개의 개인 홈 페이지를 호스팅하는 ISP가 이 범주에 속합니다.

가상 서버는 일반적으로 URL 호스트 기반입니다. 예를 들어 정적 내용만을 허용하는 구성 하나와 정적 내용 및 CGI를 허용하는 구성 하나가 있을 수 있습니다.

가상 서버 관리

가상 서버 추가

▼ 가상 서버 추가

시작하기 전에

- 가상 서버를 만들어야 하는 구성을 작성/확인해야 합니다.
- HTTP Listener를 작성/확인해야 합니다.
- 새 가상 서버의 호스트를 확인해야 합니다.

1 가상 서버를 추가해야 하는 구성을 선택합니다. 구성 탭에 표시된 구성 목록에서 구성을 선택할 수 있습니다.

2 가상 서버 탭 > 새로 만들기 버튼을 누릅니다.

3 그러면 가상 서버 구성 프로세스를 안내하는 팝업 마법사 페이지가 나타납니다. 마법사 페이지에서 다음 작업을 수행합니다.

- 새 가상 서버 정보를 입력합니다.
  a. 새 가상 서버를 확인하는 이름을 입력합니다. 이름은 영숫자일 수 있으며 점(.),대시(-) 및 밑줄(_) 문자를 포함할 수 있습니다.
  b. *(**선택 사항**)* 새 가상 서버에 추가할 호스트의 목록을 입력합니다.
  c. *(**선택 사항**)* 가상 서버에 대한 문서 루트를 입력합니다.
- 새로 구성된 가상 서버에 대한 HTTP Listener를 선택합니다. 기존 HTTP Listener를 선택하거나 새 HTTP Listener를 만들 수 있습니다.

4 마법사 요약 페이지가 표시됩니다. 구성을 변경하려면 이전을 눌러 이전 페이지로 돌아갑니다. 마침을 눌러 새 가상 서버 구성 프로세스를 완료합니다.

5 결과페이지가 표시됩니다. 오류가 발생하면 마법사에서 이전 페이지로 돌아가 가상 서버를 다시 구성합니다.

주 – CLI 사용

CLI를 통해 가상 서버를 추가하려면 다음 명령을 실행합니다.

```
wadm> create-virtual-server --user=admin --password-file=admin.pwd
--host=serverhost --port=8989 --config=config1 --document-root=../docs config1_vs_1
```

CLI 참조 `create-virtual-server`(1)를 참조하십시오.

가상 서버 구성

- 69 페이지 "가상 서버를 구성하는 방법"

가상 서버의 일반 설정을 구성하려면 다음 작업을 수행합니다.

▼ 가상 서버를 구성하는 방법

1. **구성을 선택합니다.**
   구성 목록에서 구성을 선택합니다. 구성 탭을 눌러 사용 가능한 구성 목록을 가져옵니다.
2. **가상 서버를 선택합니다.**
   가상 서버 목록에서 가상 서버를 선택합니다. 가상 서버 탭을 눌러 선택한 구성에 사용할 수 있는 가상 서버를 가져옵니다.
3. **일반 탭을 누릅니다. 다음 설정을 구성합니다.**
   - **사용 가능**— 런타임에 가상 서버를 사용할 수 있는지 여부를 지정합니다.
   - **문서 루트**— 가상 서버의 문서 루트 경로이며 가상 서버의 데이터가 저장됩니다. 여기에는 탐색된 웹 응용 프로그램 디렉토리와 로그 파일이 포함됩니다.
   - **호스트** — 둘 이상의 URL 호스트를 쉼표로 구분하여 입력할 수 있습니다.

가상 서버 복제

가상 서버를 복제하려면 다음 작업을 수행합니다.

▼ 가상 서버를 복제하는 방법

1. **구성을 선택합니다.**
   구성 목록에서 구성을 선택합니다. 구성 탭을 눌러 사용 가능한 구성 목록을 가져옵니다.

2 **가상 서버를 선택합니다.**
가상 서버 목록에서 가상 서버를 선택합니다. 가상 서버 탭을 눌러 선택한 구성에 사용할 수 있는 가상 서버를 가져옵니다.

3 **복사 버튼을 누릅니다.**
새 가상 서버에 이름을 지정합니다.

주 – CLI 사용

CLI를 통해 가상 서버를 복제하려면 다음 명령을 실행합니다.

```
wadm> copy-virtual-server --user=admin --password-file=admin.pwd --host=serverhost
--port=8989 --config=config1 --vs=config1_vs_1 copiedVs
```

copiedVS는 새 가상 서버의 이름입니다.

CLI 참조 `copy-virtual-server`(1)를 참조하십시오.

HTTP Listener 구성

서버는 구성된 가상 서버로 요청을 전달하기 전에 HTTP Listener를 통해 HTTP 요청을 수락합니다. 이 페이지에서는 HTTP Listener를 추가하고 구성할 수 있습니다.

HTTP Listener는 포트 번호 및 IP 주소의 고유한 조합이어야 합니다. IPV4 또는 IPV6 주소를 사용할 수 있습니다. IP 주소를 "*"로 설정하면 해당 포트에서 모든 IP 주소를 수신하는 HTTP Listener를 만듭니다.

HTTP Listener 만들기

다음 단계를 실행하면 가상 서버의 새 HTTP Listener를 만들어 들어오는 HTTP 요청을 처리할 수 있습니다.

1. 구성 탭 아래에 있는 **가상 서버 탭**을 누릅니다.
2. **HTTP Listener 하위 탭**을 눌러 구성된 HTTP Listener 목록을 봅니다.
3. **새로 만들기 버튼**을 눌러 새 HTTP Listener를 만들 마법사 페이지를 엽니다.

마법사 페이지에서 다음 정보를 입력합니다.

- **이름** — 새 HTTP Listener의 이름입니다.
- **포트** — HTTP Listener가 들어오는 HTTP 요청을 바인드하고 수신하는 포트입니다.

- **IP 주소** — 유효한 IPv4 또는 IPv6 주소입니다. "*"는 HTTP Listener가 구성된 포트에 대해 지정된 모든 IP 주소를 수신하는 것을 의미합니다.
- **서버 이름** — 서버 이름을 입력합니다(예: *sales.mycomp.com*).
- **기본 가상 서버** — 드롭다운 목록에서 가상 서버를 선택합니다. 이 작업은 선택한 가상 서버에 새 HTTP Listener를 연결합니다.
- **설명(선택 사항)** — HTTP Listener에 대한 짧은 설명을 입력합니다.

주 – CLI 사용

CLI를 통해 HTTP Listener를 만들려면 다음 명령을 실행합니다.

```
wadm> create-http-listener --user=admin --password-file=admin.pwd --host=serverhost
--port=8989 --listener-port=18003 --config=config1 --server-name=config1.com
--default-virtual-server-name=config1_vs_1 config1_ls_1
```

CLI 참조 `create-http-listener`(1)를 참조하십시오.

HTTP Listener 구성

다음 작업을 수행하여 기존 HTTP Listener 설정을 편집할 수 있습니다.

1. **가상 서버 탭**을 눌러 기존 HTTP Listener 설정을 편집니다.
2. **HTTP Listener 하위 탭**을 눌러 구성된 HTTP Listener의 목록을 봅니다.
3. **Listener 이름 테이블 열**에서 설정을 편집할 HTTP Listener를 누릅니다.

HTTP Listener의 일반 설정 및 보안 관련 설정을 모두 편집할 수 있습니다.

HTTP Listener 매개 변수 수정

일반 탭을 눌러 기본 및 고급 HTTP Listener 설정을 편집합니다. 다음 옵션을 구성합니다.

- **이름** — 새 HTTP Listener의 이름입니다.
- **포트** — HTTP Listener가 들어오는 HTTP 요청을 바인드하고 수신하는 포트입니다.
- **IP 주소** — 유효한 IPv4 또는 IPv6 주소입니다. "*"는 HTTP Listener가 구성된 포트에 대해 지정된 모든 IP 주소를 수신하는 것을 의미합니다.
- **서버 이름** — 서버 이름을 입력합니다(예: *sales.mycomp.com*).

HTTP Listener 고급 설정을 편집하려면 **고급 섹션** 아래에서 **고급 설정 구성 옵션**을 선택합니다. 다음 옵션을 구성합니다.

- **억셉터 스레드** — 이 수신기에서 받은 연결을 전담하여 수락하는 스레드의 수입니다. 사용 가능한 값은 1 - 128입니다.
- **프로토콜 패밀리** — 수신기에서 사용되는 프로토콜입니다. 이 값을 수정하지 마십시오. 기본값은 HTTP입니다.

- **수신 대기열 크기** — 운영 체제 수신 대기열 백로그의 최대 크기입니다.
- **수신 버퍼 크기** — 운영 체제 소켓 수신 버퍼의 크기(바이트)입니다.
- **전송 버퍼 크기** — 운영 체제 소켓 전송 버퍼의 크기(바이트)입니다.
- **I/O 차단** — HTTP 수신기 소켓이 차단 모드인지 확인합니다. 기본적으로 사용 안 함으로 설정됩니다.

◆ ◆ ◆ 6 장

6 인증서 및 키

이 장에서는 인증서 및 키 인증을 사용하여 Sun Java System Web Server의 보안을 강화하는 방법에 대해 설명합니다. 여기서는 다양한 보안 기능을 활성화하여 데이터를 보호하고 침입자의 액세스를 방지하며 원하는 사용자의 액세스만 허용하는 방법에 대해 설명합니다.

이 장을 읽기 전에 공용 키 암호화에 대한 기본 개념을 알고 있어야 합니다. 이 개념에는 암호화 및 복호화, 공용 및 개인 키, 디지털 인증서 및 암호화 프로토콜 등이 포함됩니다.

인증용 인증서 사용

인증은 신분을 확인하는 과정입니다. 네트워크 상호 작용이라는 맥락에서 인증은 한 쪽이 다른 쪽의 신분을 명확히 확인하는 것입니다. 인증서는 인증을 지원하는 방법 중 한 가지입니다.

인증서는 개인, 회사 또는 기타 엔티티의 이름을 지정하는 디지털 데이터로 구성되며 인증서에 포함된 공용 키는 해당 엔티티에 속한다는 것을 인증합니다. 클라이언트와 서버 모두 인증서를 가질 수 있습니다.

인증서는 인증 기관 또는 CA에서 발행하고 디지털로 서명됩니다. CA는 인터넷에서 인증서를 판매하는 회사일 수도 있고 회사의 인트라넷 또는 익스트라넷용으로 인증서를 발행하는 부서일 수도 있습니다. 다른 사람의 신분을 확인하는 데 충분히 신뢰할 수 있는 CA를 선택합니다.

인증서에 의해 확인되는 공용 키와 엔티티의 이름 외에도 인증서에는 만기일, 인증서를 발행한 CA 이름 및 발행 CA의 "디지털 서명"이 포함됩니다.

주 – 암호화를 활성화하려면 서버 인증서가 설치되어 있어야 합니다.

서버 인증

서버 인증은 클라이언트가 서버의 신분을 확인하는 것을 말합니다. 즉, 특정 네트워크 주소에서 서버에 대한 책임이 있는 조직의 신분을 확인하는 것입니다.

클라이언트 인증

클라이언트 인증은 서버에서 클라이언트의 신분을 확인하는 것을 말합니다. 즉, 클라이언트 소프트웨어를 사용하는 개인의 신분을 확인하는 것입니다. 개인이 여러 개의 신분증을 가질 수 있는 것처럼 클라이언트에는 여러 개의 인증서가 있을 수 있습니다.

인증서 키 유형

RSA 키 외에도, Sun Java System Web Server 7.0에서는 ECC(Elliptic Curve Cryptography)를 지원합니다.

ECC는 새롭게 각광 받고 있는 공용 키 암호화 시스템으로, RSA와 같은 기존의 암호화 시스템에 비해 크기가 작은 키로도 동등한 수준의 보안을 제공하기 때문에 계산 속도가 빠르고 전력 소모가 적을 뿐 아니라 메모리와 대역폭도 절약됩니다. ECC(Elliptic Curve Cryptography)는 최근에 미국 정부의 승인을 받았습니다.

이제 인증서 요청을 생성할 것인지 아니면 RSA 키 또는 ECC 키를 사용하여 자체 서명된 인증서를 생성할 것인지 선택할 수 있습니다.

RSA 키의 경우 다양한 키 크기가 제공됩니다. 키 크기가 커지면 암호화 성능이 좋아집니다. 기본 키 크기는 1024입니다. ECC 키에서는 키 쌍을 생성할 곡선을 선택해야 합니다. 다양한 조직(ANSI X9.62, NIST, SECG)에서 수많은 곡선을 명명하였으며 Sun Java System Web Server 7.0에서는 현재 지정된 모든 곡선을 지원합니다.

자체 서명된 인증서를 사용하는 대신 CA에 인증서를 요청하려는 경우에는 먼저 원하는 CA에 연락하여 ECC 사용에 관한 최신 정보를 받으십시오. 사용자 사례에 적합한 특정

ECC 곡선이 있는지 문의하면 됩니다. CA에서 곡선 선택에 관한 안내를 제공하지 않고 조직 내부 정책에서도 정보를 얻을 수 없는 경우에는 다음 사항을 고려하십시오. ECC는 발전 중인 기술이기 때문에 특정 사례에 대해 권장되는 곡선이 이 문서를 작성한 이후로 달라졌을 수도 있습니다.

다음은 일부 지원되는 ECC 곡선의 목록입니다.

prime256v1
secp256r1
nistp256
secp256k1
secp384r1
nistp384
secp521r1
nistp521
sect163k1
nistk163
sect163r1
sect163r2
nistb163
sect193r1
sect193r2
sect233k1
nistk233k1
nistk233
sect233r1
nistb233
sect239k1
sect283k1
nistk283
sect283r1
nistb283
sect409k1
nistk409
sect571k1
nistk571
sect571r1
nistb571
secp160k1
secp160r1
secp160r2
secp192k1
secp192r1
nistp192

secp224k1
secp224r1
nistp224
prime192v1

자체 서명된 인증서 만들기

CA에서 서명한 인증서가 필요하지 않거나 CA가 인증서에 서명하는 동안 새 SSL 구현을 테스트하려는 경우에는 자체 서명된 인증서를 생성할 수 있습니다. 이러한 임시 인증서로 인해 클라이언트 브라우저에서 서명한 인증 기관을 알 수 없으며 신뢰할 수 없다는 내용의 오류가 발생합니다.

CLI를 통해 자체 서명된 인증서를 만들려면 다음 명령을 실행합니다.

```
wadm> create-selfsigned-cert --user=admin --port=8989 --password-file=admin.pwd
--config=config1 --token=internal --org-unit=org1 --locality=XYZ --state=DEF
--validity=10 --org=sun --country=ABC --server-name=serverhost --nickname=cert1
```

CLI 참조 `create-selfsigned-cert`(1)를 참조하십시오.

인증서 관리

인증서 요청

인증서는 개인, 회사 또는 기타 엔티티의 이름을 지정하는 디지털 데이터로 구성되며 인증서에 포함된 공용 키가 개인에 속한다는 것을 인증합니다. SSL 사용 서버에는 인증서가 있어야 하며 클라이언트는 선택적으로 인증서를 가질 수 있습니다.

인증서는 인증 기관 또는 CA에서 발행하고 디지털로 서명됩니다. CA는 인터넷에서 인증서를 판매하는 회사일 수도 있고 회사의 인트라넷 또는 익스트라넷용으로 인증서를 발행하는 부서일 수도 있습니다. 다른 사람의 신분을 확인하는 데 충분히 신뢰할 수 있는 CA를 선택합니다.

인증서를 요청하여 인증 기관(CA)에 제출할 수 있습니다. 회사에 내부 CA가 있는 경우에는 해당 CA로부터 인증서를 요청합니다. 상용 CA로부터 인증서를 구매할 계획인 경우에는 CA를 선택하고 필요한 정보 형식이 있는지 문의합니다. 서버용으로 자체 서명된 인증서를 만들 수도 있습니다. 자체 서명된 인증서는 인터넷 배포에는 적합하지 않지만 CA의 개입 없이도 테스트 서버를 설정할 수 있기 때문에 개발과 테스트에는 매우 유용합니다.

위에 언급한 것과 같이 인증서에는 엔티티(이 경우 웹 서버)의 공용 키가 포함됩니다. 공용 키는 특정 알고리즘을 기반으로 생성됩니다(알고리즘 유형도 인증서에 암호화). 다음 절에서는 Web Server에서 키에 지원하는 알고리즘 유형에 대한 배경을 설명합니다.

▼ 인증서를 요청하는 방법

1 서버 인증서 탭 > 요청 버튼을 누릅니다.

2 구성 선택

구성 목록에서 인증서를 설치하려는 구성을 선택합니다.

3 토큰 선택

키가 들어있는 토큰(암호화 장치)을 선택합니다. Sun Java System Web Server 7.0에서 유지 관리하는 로컬 키 데이터베이스에 키가 저장되어 있으면 내부를 선택합니다. 스마트 카드 또는 기타 외부 장치나 엔진에 키가 저장되어 있으면 드롭다운 목록 상자에서 외부 토큰의 이름을 선택합니다. 선택한 토큰의 비밀번호를 입력합니다.

4 세부 정보 입력

요청 프로세스를 시작하기 전에 CA에서 필요한 정보를 확인하십시오. 상용 CA 또는 내부 CA 중 어느 곳에 서버 인증서를 요청할 것인가에 상관 없이 다음 정보를 제공해야 합니다.

- **서버 이름**은 DNS 조회에 사용되는 정규화된 호스트 이름이어야 합니다(예: *www.sun.com*). 이는 브라우저가 사이트에 연결할 때 사용하는 URL 내의 호스트 이름입니다. 이 두 이름이 일치하지 않으면 클라이언트에게 인증서 이름이 사이트 이름과 일치하지 않는다는 알림을 보내고 인증서의 진위 여부를 의심하게 됩니다.

  또한 내부 CA에 인증서를 요청하는 경우에는 이 필드에 와일드카드와 정규 표현식을 입력할 수 있습니다. 대부분의 공급업체는 공통 이름에 와일드카드나 정규 표현식이 있을 경우 인증서 요청을 승인하지 않습니다.
- **조직**은 회사, 교육 기관, 협력 관계 등의 공식적, 법적 이름을 지정합니다. 대부분의 CA는 이 정보에 대해 법적 문서(사업자 등록 등)로 확인할 것을 요구합니다.
- **조직 단위**는 회사 내의 조직에 대한 설명을 입력하는 선택 필드입니다. 또한 *Inc.*, *Corp.* 등이 없는 비공식적인 회사 이름을 나타내는 데 사용할 수 있습니다.
- **구/군/시**는 선택 필드로, 조직의 소재 시/도 또는 국가를 나타냅니다.
- **시/도**는 선택 필드입니다.
- **국가**는 필수 필드로 국가 이름의 두 자리 약자를 지정합니다(ISO 형식). 미국의 국가 코드는 US입니다.

이 모든 정보는 DN(고유 이름)이라고 하는 일련의 속성 값 쌍으로 조합되어 인증서의 개체를 구성합니다.

5 인증서 옵션 선택

키 정보를 입력해야 합니다. 키 유형으로 RSA 또는 ECC를 선택할 수 있습니다. 키 유형이 RSA인 경우 키 크기는 1024, 2048 또는 4098이 될 수 있습니다. 키 유형이 ECC인 경우에는 곡선도 선택해야 합니다. 새로운 키 쌍을 생성하려면 시간이 걸립니다. 키 길이가 길수록 마법사에서 키를 생성하는 시간이 더 오래 걸립니다.

주의 – 나중에 서명을 위해 요청을 제출할 CA에서 지원할 수 있는 키 유형을 선택해야 합니다.

6 인증서 유형 선택

인증서의 인증서 서명 기관(CSA)을 선택합니다(자체 서명 또는 CA 서명). 자체 서명된 인증서를 선택하는 경우에는 인증서의 HTTP Listener를 연결할 수도 있습니다. 이 작업을 나중에 수행할 수도 있습니다.

7 요청 생성

CA 서명이 있는 인증서의 경우 생성된 인증서 요청은 ASCII 형식으로 사용할 수 있습니다. 자체 서명된 인증서의 경우에는 인증서가 바로 설치됩니다. 자체 서명 유형인 경우에는 별명, 유효 기간(개월) 및 보안 요청을 처리할 HTTP Listener 이름의 값을 입력합니다.

8 결과 보기

이 페이지에서는 선택한 옵션에 대한 요약을 제공합니다. 요청 생성을 완료하려면 마침을 누릅니다.

주 – CLI 사용

CLI를 통해 인증서를 요청하려면 다음 명령을 실행합니다.

```
wadm> create-cert-request --user=admin --password-file=admin.pwd
--host=serverhost --port=8989 --config=config1 --server-name=servername.org
--org=sun --country=ABC --state=DEF --locality=XYZ --token=internal
```

CLI 참조 `create-cert-request`(1)를 참조하십시오.

인증서 설치

CA에서 인증서를 받은 후 관리 콘솔을 사용하여 특정 구성에 대해 인증서를 설치할 수 있습니다.

▼ 인증서 설치

1 서버 인증서 탭 > 설치 버튼을 누릅니다.

2 구성 선택

구성 목록에서 인증서를 설치하려는 구성을 선택합니다.

3 토큰 선택

키가 들어있는 토큰(암호화 장치)을 선택합니다. Sun Java System Web Server 7.0에서 유지 관리하는 로컬 키 데이터베이스에 키가 저장되어 있으면 내부를 선택합니다. 스마트 카드 또는 기타 외부 장치나 엔진에 키가 저장되어 있으면 드롭다운 목록 상자에서 외부 토큰의 이름을 선택합니다. 선택한 토큰의 비밀번호를 입력합니다.

4 인증서 데이터 입력

제공된 텍스트 영역에 인증서 텍스트를 붙여넣습니다. 텍스트를 복사하여 붙여넣는 경우, 반드시 "Begin Certificate" 및 "End Certificate" 헤더를 시작 및 끝 하이픈과 함께 포함해야 합니다. 찾아보기 버튼을 누르고 DER 파일을 수동으로 선택할 수도 있습니다.

5 인증서 세부 정보 제공

인증서에 사용할 별칭을 제공합니다. 보안 요청을 처리할 HTTP Listener를 사용 가능한 목록에서 선택합니다.

6 결과 보기

이 페이지에서는 선택한 옵션에 대한 요약을 제공합니다. 설치 프로세스를 완료하려면 마침을 누릅니다.

주 – CLI 사용

CLI를 통해 인증서를 설치하려면 다음 명령을 실행합니다.

```
wadm> install-cert --user=admin --port=8989  --password-file=admin.pwd
--config=config1 --token=internal --cert-type=server --nickname=cert1 cert.req
```

여기서 cert.req에는 인증서 데이터가 포함됩니다.

CLI 참조 `install-cert`(1)를 참조하십시오.

인증서 갱신

다음 단계를 통해 기존 인증서를 갱신할 수 있습니다.

▼ 인증서 갱신

1. **서버 인증서 탭 > 인증서 이름 > 갱신 버튼을 누릅니다.**

2. **토큰 정보 입력**
   필요한 경우 토큰의 비밀번호를 입력합니다. 그렇지 않으면 다음을 눌러 계속합니다.

3. **인증서 세부 정보 업데이트**
   인증서 세부 정보를 검토하고 유효 기간을 개월 수로 입력합니다.

4. **키 정보 업데이트**
   키 유형으로 RSA 또는 ECC를 선택할 수 있습니다. 키 유형이 RSA인 경우 키 크기는 1024, 2048 또는 4098이 될 수 있습니다. 키 유형이 ECC인 경우에는 곡선도 선택해야 합니다. 새로운 키 쌍을 생성하려면 시간이 걸립니다.

5. **요약 보기**
   이 페이지에서는 선택한 옵션에 대한 요약을 제공합니다. 갱신 프로세스를 완료하려면 마침을 누릅니다.

인증서 삭제

인증서를 삭제하려면 다음 작업을 수행하십시오.

▼ 인증서 삭제

1. **서버 인증서 탭을 누릅니다.**

2. **인증서 선택**
   인증서 목록에서 인증서 이름을 선택합니다.

3. **인증서 삭제**
   삭제 버튼을 눌러 선택한 인증서를 삭제합니다.

주 – CLI 사용

CLI를 통해 인증서를 삭제하려면 다음 명령을 실행합니다.

```
wadm> delete-cert --user=admin --port=8989  --password-file=admin.pwd
--token=internal --config=config1 cert1
```

CLI 참조 `delete-cert`(1)를 참조하십시오.

Administration Server 인증서 갱신

Administration Server 인증서를 갱신하려면 `renew-admin-certs` CLI 명령을 실행합니다. 이 명령을 사용하면 별명이 `Admin-CA-Cert`, `Admin-Server-Cert` 및 `Admin-Client-Cert`인 관리 인증서를 갱신할 수 있습니다. 이 명령은 현재 실행 중이며 갱신된 인증서로 액세스할 수 있는 노드도 업데이트합니다.

이 명령을 실행한 후 새 인증서를 적용하려면 Administration Server와 노드를 다시 시작하는 것이 좋습니다. 인증서를 갱신하는 동안 노드가 오프라인 상태였던 경우(실행 중이 아니거나 네트워크 문제로 인해 액세스할 수 없었던 경우)에는 노드를 다시 등록해야 합니다. Administration Server 인증서를 갱신하려면 다음 명령을 실행합니다.

```
wadm> renew-admin-certs --user=admin --password-file=admin.pwd --host=serverhost
--port=8989 --validity=120
```

CLI 참조 `renew-admin-certs`(1)를 참조하십시오.

인증서 철회 목록(CRL) 관리

CRL(Certificate Revocation Llist)을 사용하면 클라이언트나 서버 사용자가 더 이상 신뢰하지 않는 인증서 및 키를 알릴 수 있습니다. 예를 들어, 사용자가 사무실을 변경하거나 인증서가 만료되기 전에 조직을 나가는 등 인증서의 데이터가 변경된 경우에는 인증서가 철회되고 해당 데이터가 CRL에 표시됩니다. CRL은 CA에서 생성되며 정기적으로 업데이트됩니다.

▼ CRL 설치

CA에서 CRL을 얻으려면 다음 단계를 수행합니다.

1. **CA에서 CRL을 파일로 받습니다.**
2. **관리 콘솔에서 구성 페이지로 이동합니다.**

3. **인증서 > 인증 기관 탭을 누릅니다.**
4. **CRL 설치 버튼을 누릅니다.**
5. **해당 파일의 전체 경로 이름을 입력합니다.**
6. **OK를 누릅니다.**

주 – 데이터베이스에 이미 CRL이 있는 경우에는 인증서 철회 목록 대체 페이지가 표시됩니다.

---

7. **변경 사항을 적용하려면 배포를 눌러야 할 수도 있습니다.**

주 – CLI 사용

CLI를 통해 CRL을 설치하려면 다음 명령을 실행합니다.

```
wadm> install-crl --user=admin --password-file=admin.pwd
--host=serverhost --port=8989 --config=config1 data/install-crl/ServerSign.crl
```

CLI 참조 `install-crl`(1)을 참조하십시오.

▼ CRL 삭제

1. **관리 콘솔에서 구성 페이지로 이동합니다.**
2. **인증서 > 인증 기관 탭을 누릅니다.**
3. **CRL 항목을 선택하고 삭제를 누릅니다.**
4. **변경 사항을 적용하려면 배포를 눌러야 할 수도 있습니다.**

주 – CLI 사용

CLI를 통해 CRL을 삭제하려면 다음 명령을 실행합니다.

```
wadm> delete-crl --user=admin --password-file=admin.pwd
--host=serverhost --port=8989 --config=config1 issuer
```

CLI 참조 `delete-crl`(1)을 참조하십시오.

내부 토큰에 비밀번호 설정

내부 PKCS11 토큰에 비밀번호를 설정하려면 다음 작업을 수행합니다.

▼ 토큰 비밀번호 설정

1. **관리 콘솔에서 구성 페이지로 이동합니다.**
2. **인증서 > PKCS11 토큰 탭을 누릅니다.**
3. **PKCS11 토큰 이름을 누릅니다(기본값은 내부).**
4. **토큰 상태 확인란을 선택합니다.**
5. **비밀번호 정보를 입력합니다.**
6. **인스턴스를 시작할 때 비밀번호 프롬프트를 표시하지 않으려면 인스턴스 시작 시 새 비밀번호 확인 메시지 표시 안 함 확인란을 선택합니다. OK를 누릅니다.**
7. **비밀번호는 구성에 저장됩니다. 비밀번호를 제거하려면 위의 단계를 수행하고 비밀번호 설정 해제 옵션을 선택합니다.**

주 – CLI 사용

CLI를 통해 내부 PKCS11 토큰의 비밀번호를 설정하려면 다음 명령을 실행합니다.

```
wadm> set-token-pin --user=admin --password-file=admin.pwd --host=serverhost
--port=8989 --config=config1 --token=internal
```

CLI 참조 `set-token-pin`(1)을 참조하십시오.

서버의 SSL 구성

`create-cert-request` 명령을 사용하여 요청을 생성하고 요청을 CA로 보냅니다. 나중에 CA에서 인증서를 받으면 `install-cert` 명령을 사용해서 설치해야 합니다.
마이그레이션할 Java 키 저장소에 키와 인증서가 있는 경우에는 `migrate-jks-keycert` 명령을 사용합니다. 개발/테스트 서버인 경우에는 `create-selfsigned-cert` 명령을 사용하여 자체 서명된 인증서를 생성하는 것이 가장 쉬운 방법입니다.

```
wadm> create-selfsigned-cert --server-name=hostname --nickname=MyServerCert
--token=internal
```

추가 옵션과 예는 설명서 페이지를 참조하십시오.

인증서가 설치된 후에는 SSL을 사용할 몇 개의 포트에 수신기가 필요합니다.

```
wadm> create-http-listener --user=admin --password-file=admin.pwd --host=serverhost
--port=8989 --listener-port=18003 --config=config1 --server-name=config1.com
--default-virtual-server-name=config1_vs_1 config1_ls_1
```

이제 수신기의 SSL을 활성화하고 수신기를 인증서 별명에 연결합니다.

```
wadm> set-ssl-prop --http-listener=http-listener-ssl enabled=true
wadm> set-ssl-prop --http-listener=http-listener-ssl server-cert-nickname=MyServerCert
```

이 설정을 수행하고 나면 구성을 배포하고 인스턴스를 시작합니다.

```
wadm> deploy-config config_name
wadm> start-instance --config config_name hostname
```

구성에 대해 SSL 암호 활성화

구성에 대해 SSL 암호를 활성화하려면 다음 명령을 실행합니다.

```
wadm> enable-ciphers --user=admin --password-file=admin.pwd --host=serverhost
--port=8989 --config=config1 --http-listener=http-listener-1
--cipher-type=ssl2 SSL_RC4_128_WITH_MD5
```

CLI 참조 `enable-ciphers`(1)를 참조하십시오.

HTTP Listener의 보안 활성화

주 – 사용 가능한 인증서가 설치된 경우에만 HTTP Listener에 보안을 사용할 수 있습니다.

인증서가 있으면 인증서를 HTTP Listener에 연결하여 서버 보안을 강화할 수 있습니다.

암호화는 의도된 수신자 외에는 아무 의미가 없도록 정보를 변환하는 프로세스입니다. 복호화는 암호화된 정보를 변환하여 다시 의미 있는 정보가 되도록 하는 프로세스입니다. Sun Java System Web Server는 SSL 및 TLS 프로토콜을 지원합니다.

암호는 암호화 또는 복호화에 사용되는 암호화 알고리즘(수학적 함수)입니다. SSL 및 TLS 프로토콜에는 다양한 암호 제품군이 포함됩니다. 보안의 안전성과 강도는 암호마다 다릅니다. 일반적으로 암호에서 사용하는 비트 수가 많을수록 데이터를 해독하는 것이 어렵습니다.

모든 양방향 암호화 프로세스에서 양쪽에 반드시 동일한 암호가 있어야 합니다. 다양한 암호를 사용할 수 있으므로 서버를 가장 공통적으로 사용되는 암호용으로 설정해야 합니다.

보안 연결에서 클라이언트와 서버는 양쪽이 통신에 사용할 수 있는 가장 강력한 암호화를 사용하도록 동의합니다. 암호는 SSL2, SSL3 및 TLS 프로토콜 중에서 선택할 수 있습니다.

주 – SSL 버전 2.0 이후 보안과 성능이 향상되었으므로 SSL3을 사용할 수 있는 클라이언트인 경우 SSL2를 사용하면 안 됩니다. 클라이언트 인증서가 SSL 2 암호와 작동되도록 보장되지 않습니다.

암호화 프로세스 그 자체는 서버의 비밀 정보를 보안하는 데 충분하지 않습니다. 실제의 암호화 결과를 얻거나 이전에 암호화된 정보를 해독하려면 키와 함께 암호화 암호를 사용해야 합니다. 이를 위해서 암호화 프로세스에는 두 가지 키(공용 키와 개인 키)가 사용됩니다. 공용 키로 암호화된 정보는 오직 연결된 개인 키로만 해독할 수 있습니다. 공용 키는 인증서의 일부로 만들어지며 오직 연결된 개인 키만 보호됩니다.

Sun Java System Web Server는 암호화 통신에 SSL(Secure Sockets Layer) 및 TLS(Transport Layer Security) 프로토콜을 사용합니다. SSL과 TLS는 응용 프로그램과 상관 없는 고수준 프로토콜로 응용 프로그램에 투명하게 배치될 수 있습니다.

SSL과 TLS 프로토콜은 서버와 클라이언트가 서로를 인증하고, 인증서를 전송하며 세션 키를 설정하는 등의 작업에 사용되는 다양한 암호를 지원합니다. 클라이언트와 서버에서 지원하는 프로토콜, 암호화 정도에 대한 회사 정책, 암호화된 소프트웨어의 수출에 대한 정부 규제 등, 다양한 요인에 따라 지원되는 암호 제품군이 달라집니다. 다른 기능 중, SSL 및 TLS 핸드셰이크 프로토콜에 따라 서버와 클라이언트가 통신에 사용할 암호 제품군을 선택하는 방식이 결정됩니다.

구성 > HTTP Listener > 보안 탭을 눌러 HTTP Listener 보안 설정을 편집합니다. 다음 표에는 이 페이지에서 구성할 수 있는 등록 정보가 나열되어 있습니다.

표 6–1 HTTP Listener 보안 등록 정보

등록 정보	설명
이름	HTTP Listener의 이름입니다.
보안	선택한 HTTP Listener의 보안을 활성화/비활성화합니다.
인증서	사용 가능한 인증서 중에서 서버 인증서를 선택합니다. 이 작업을 수행하려면 RSA 또는 ECC 인증서가 설치되어 있어야 합니다.
클라이언트 인증	클라이언트 인증서가 필수인지 또는 선택인지 여부를 지정합니다. 클라이언트 인증을 비활성화하려면 False 옵션을 선택합니다.
인증 시간 초과	클라이언트 인증 핸드셰이크가 실패하게 될 시간 초과입니다. [0.001–3600]. 기본값은 60초입니다.

표 6–1 HTTP Listener 보안 등록 정보 *(계속)*

등록 정보	설명
최대 인증 데이터	버퍼링할 인증 데이터의 최대 크기입니다. [0–2147.0483647.0]. 기본값은 104857.06입니다.
SSL 버전 2/버전 3	SSL 버전 2/ SSL 버전 3을 활성화/비활성화합니다.
TLS	TLS를 활성화/비활성화합니다. 버전 롤백 검색은 기본적으로 활성화되어 있습니다. 이렇게 구성하면 서버가 중간개입자(man-in-the-middle) 버전 롤백 공격 시도를 감지할 수 있습니다. TLS 사양을 잘못 구현한 일부 클라이언트와의 상호 운용성을 위해 이 옵션을 사용 불가로 설정해야 할 수도 있습니다.
SSL3/SSL2/TLS 암호	웹 서버의 안전을 보호하려면 SSL을 활성화해야 합니다. SSL 2.0, SSL 3.0 및 TLS 암호화 프로토콜을 사용하고 다양한 암호화 제품군을 선택할 수 있습니다. SSL과 TLS는 Administration Server용 청취 소켓에서 사용하도록 설정할 수 있습니다. Server Manager용 청취 소켓에서 SSL 및 TLS를 사용하면 해당 청취 소켓과 연결된 모든 가상 서버용 보안 기본 설정이 설정됩니다. 기본 설정의 경우 가장 많이 사용되는 암호를 허용합니다. 특정 암호화 제품군을 사용할 수 없는 특별한 이유가 있는 경우가 아니라면 모두 허용해야 합니다.

◆ ◆ ◆ 7 장

7 서버 액세스 제어

인증, 권한 및 액세스 제어 등 다양한 보안 서비스 및 기법을 통해 웹 서버에 있는 자원을 보호할 수 있습니다. 이 장에서는 Sun Java System Web Server 7.0에 대한 액세스를 제어하기 위해 지원되는 기법 중 몇 가지에 대해 설명합니다.

액세스 제어란?

인증은 신분을 확인하는 과정입니다. 권한은 특정 사용자가 제한된 자원에 액세스할 수 있도록 허용하는 것을 의미하며 액세스 제어 기법을 통해 이러한 제한을 실행합니다. 인증 및 권한은 여러 보안 모델(웹 응용 프로그램 보안, htaccess, 인증 영역 등)과 서비스를 통해 실행될수 있습니다.

액세스 제어를 사용하면 다음을 결정할 수 있습니다.

- Administration Server에 액세스할 수 있는 사용자
- 액세스할 수 있는 응용 프로그램
- 웹 사이트의 파일 또는 디렉토리에 액세스할 수 있는 사용자

서버의 전체 또는 일부, 또는 웹 사이트의 파일 또는 디렉토리에 대한 액세스를 제어할 수 있습니다. ACE(Access Control Entries)라는 규칙 계층을 만들어 액세스를 허용 또는 거부합니다. 만드는 ACE의 모음을 액세스 제어 목록(ACL)이라고 합니다.

기본으로 서버에는 하나의 ACL 파일이 있으며 여기에는 여러 개의 ACL이 있습니다. 들어오는 요청에 사용할 가상 서버를 결정한 다음 Sun Java System Web Server는 해당

가상 서버에 ACL이 구성되어 있는지 확인합니다. 현재 요청에 적용되는 ACL이 있는 경우 서버는 ACE를 평가하여 액세스를 허용할 것인지 또는 거부할 것인지 결정합니다.

다음을 기준으로 액세스를 허용 또는 거부합니다.

- 요청하는 사용자(사용자 그룹)
- 요청의 출처(호스트-IP)
- 요청이 발생한 시간(예: 하루 중 시간)
- 사용되는 연결의 종류(SSL)

액세스 제어의 작동 방식

서버에 페이지에 대한 요청이 수신되면 서버는 ACL 파일에 있는 규칙을 사용하여 액세스 허용 여부를 결정합니다. 규칙은 요청을 보내는 컴퓨터의 호스트 이름 또는 IP 주소를 참조할 수 있습니다. 또한 LDAP 디렉토리에 저장된 사용자 및 그룹을 참조할 수 있습니다.

주 – 일치하는 ACL이 두 개 이상인 경우 서버는 일치되는 마지막 ACL 문을 사용합니다. 일치하는 마지막 문이 `uri` ACL이기 때문에 `default` ACL은 무시됩니다.

위 그림은 Web Server 7.0에서 액세스 제어가 작동하는 방식을 나타냅니다. 사용자

에이전트(클라이언트)가 Web Server에 액세스합니다. Web Server는 `obj.conf` 파일에 있는 `PathCheck` 지시문을 실행합니다. Web Server는 클라이언트에 HTTP 401(인증되지 않음)을 반환합니다. 클라이언트는 사용자의 인증 확인을 요구하는 프롬프트를 표시합니다. 클라이언트가 브라우저인 경우에는 로그인 대화 상자가 열립니다. 사용자가 로그인 정보를 입력합니다. Web Server는 내부 `check-acl` 함수를 실행합니다. Web Server는 사용자 자격 증명을 검증하고 요청을 처리합니다.

사용자 그룹용 액세스 제어 설정

웹 서버에 대한 액세스를 특정 사용자 또는 그룹으로 제한할 수 있습니다. 사용자 그룹 액세스 제어를 사용하려면 사용자가 해당 서버에 액세스하기 전에 사용자 이름과 비밀번호를 입력해야 합니다. 서버는 클라이언트 인증서에 있는 정보를 디렉토리 서버 항목과 비교합니다.

Administration Server는 오직 기본 인증만 사용합니다. Administration Server에 클라이언트 인증이 필요하도록 하려면 ACL 파일을 직접 편집하여 방법을 SSL로 변경해야 합니다.

사용자 그룹 인증은 Web Server에서 사용자 그룹 데이터베이스에 있는 항목을 읽어서 수행합니다. 디렉토리 서비스가 액세스 제어를 구현하는 데 사용하는 정보는 다음 중 한 가지 소스에서 구합니다.

- 내부 보통 파일 유형 데이터베이스

- 외부 LDAP 데이터베이스

서버가 외부 LDAP 기반 디렉토리 서비스를 사용하는 경우 서버 인스턴스용으로 다음 유형의 사용자 그룹 인증 방법을 지원합니다.

- Default
- Basic
- SSL
- Digest
- 기타

서버가 내부 파일 기반 디렉토리 서비스를 사용하는 경우 서버 인스턴스용으로 다음 유형의 사용자 그룹 인증 방법을 지원합니다.

- Default
- Basic
- Digest

사용자 그룹 인증을 수행하려면 사용자가 서버 또는 웹 사이트의 파일 및 디렉토리에 액세스하기 전에 자신을 인증해야 합니다. 인증 시 사용자는 클라이언트 인증서를 사용하여 사용자 이름과 비밀번호를 입력하는 방식으로 신분을 확인합니다. 클라이언트 인증서는 SSL 통신에만 필요합니다.

Default 인증

Default 인증은 가장 많이 사용되는 방법입니다. Default 설정에서는 `server.xml` 파일에 있는 Default 방법을 사용하고 `server.xml`에 설정이 없으면 "Basic"을 사용합니다. Default를 선택하면 ACL 규칙에서 ACL 파일에 방법을 지정하지 않습니다. Default를 선택한 경우 `obj.conf` 파일에서 한 줄만 편집하면 모든 ACL의 방법을 쉽게 변경할 수 있습니다.

Basic 인증

Basic 인증을 사용하려면 사용자가 웹 서버나 웹 사이트에 액세스하기 위해 사용자 이름과 비밀번호를 입력해야 합니다. 이 설정이 기본값입니다. Sun Java System Directory Server와 같은 LDAP 데이터베이스나 파일에 사용자 및 그룹 목록을 만들어 저장해야 합니다. 웹 서버가 아닌 다른 루트 디렉토리에 설치된 디렉토리 서버 또는 원격 컴퓨터에 설치된 디렉토리 서버를 사용해야 합니다.

Administration Server 또는 웹 사이트에서 사용자 그룹 인증이 있는 자원에 액세스하려는 경우 웹 브라우저에 사용자 이름과 비밀번호를 입력하라는 대화 상자가 표시됩니다. 서버에서 암호화 기능이 사용되는지의 여부에 따라 이 정보는 암호화 또는 암호화되지 않은 형태로 서버에 입력됩니다.

주 – SSL 암호화가 없는 Basic 인증을 사용하는 경우 사용자 이름과 비밀번호가 암호화되지 않은 텍스트로 네트워크에 전송됩니다. 네트워크 패킷은 가로챌 수 있으며 사용자 이름과 비밀번호가 도용될 수 있습니다. Basic 인증은 SSL 암호화, 호스트-IP 인증 또는 두 가지 인증을 모두 사용하는 경우에 가장 효과적입니다. Digest 인증을 사용하면 이러한 문제를 방지할 수 있습니다.

SSL 인증

서버는 다음 두 가지 방법을 사용하여 보안 인증서가 있는 사용자의 신분을 확인합니다.

- 클라이언트 인증서의 정보를 신분 증명으로 사용
- LDAP 디렉토리에 게시된 클라이언트 인증서 확인(추가)

서버가 클라이언트 인증용으로 인증서 정보를 사용하도록 설정하면 서버는 다음 작업을 수행합니다.

- 우선 인증서가 신뢰할 수 있는 인증 기관에서 발급된 것인지 확인합니다. 그렇지 않은 경우 인증이 실패하며 트랜잭션이 종료됩니다.
- 인증서가 신뢰할 수 있는 인증 기관(CA)에서 발급된 경우 `certmap.conf` 파일을 사용하여 인증서를 사용자 항목에 매핑합니다.
- 인증서가 올바로 매핑된 경우 해당 사용자에 대해 지정된 ACL 규칙을 확인합니다. 인증서가 올바로 매핑된 경우라도 ACL 규칙에 따라 사용자 액세스를 거부할 수 있습니다.

특정 자원에 대한 액세스 제어를 위해 클라이언트 인증을 요구하는 것은 서버에 대한 모든 연결에 대해 클라이언트 인증을 요구하는 것과 다릅니다. 모든 연결에 대해 서버가 클라이언트 인증을 요구하도록 설정한 경우 클라이언트는 신뢰할 수 있는 인증 기관에서 발급한 유효한 인증서만 제시하면 됩니다. 서버의 액세스 제어가 사용자 및 그룹 인증을 위해 SSL 방법을 사용하도록 설정하는 경우 클라이언트는 다음 작업을 수행합니다.

- 신뢰할 수 있는 인증 기관에서 발급한 유효한 인증서를 제시합니다.
- 인증서는 LDAP에 있는 유효한 사용자에 매핑되어야 합니다.
- 액세스 제어 목록에서 적절히 평가해야 합니다.

액세스 제어와 함께 클라이언트 인증을 요구하는 경우 웹 서버용 SSL 암호를 사용하도록 설정해야 합니다.

SSL 인증이 요구되는 자원에 성공적으로 액세스하려면 웹 서버가 신뢰하는 인증 기관으로부터 클라이언트 인증서가 발급되어야 합니다. 웹 서버의 `certmap.conf` 파일이 브라우저에 있는 클라이언트 인증서와 디렉토리 서버에 있는 클라이언트 인증서를 비교하도록 구성된 경우에는 클라이언트 인증서가 디렉토리 서버에 게시되어야 합니다. 그러나 `certmap.conf` 파일은 인증서의 선택된 정보만 디렉토리 서버 항목과 비교하도록 구성할 수 있습니다. 예를 들어 브라우저 인증서의 사용자 아이디와 전자 메일 주소만 디렉토리 서버 항목과 비교하도록 `certmap.conf` 파일을 구성할 수 있습니다.

주 – 인증서는 LDAP 디렉토리와 비교해 확인되기 때문에 SSL 인증 방법을 사용하려면 `certmap.conf` 파일을 수정해야 합니다. 서버로의 모든 연결에 대해 클라이언트 인증이 요구되는 경우에는 이 파일을 변경할 필요가 없습니다. 클라이언트 인증서를 사용하도록 선택한 경우 `magnus.conf`의 `AcceptTimeout` 지시문 값을 올려야 합니다.

Digest 인증

LDAP 기반 또는 파일 기반 디렉토리 서비스를 사용하여 Digest 인증을 수행하도록 서버를 구성할 수 있습니다.

Digest 인증을 사용하면 사용자가 사용자 이름과 비밀번호를 일반 텍스트로 전송하지 않고 사용자 이름과 비밀번호를 기준으로 인증할 수 있습니다. 브라우저는 MD5 알고리즘을 사용하여 Web Server가 제공하는 사용자 비밀번호 및 일부 정보를 사용하는 다이제스트 값을 만듭니다.

서버에서 LDAP 기반 디렉토리 서비스를 사용하여 Digest 인증을 수행하는 경우 이 다이제스트 값은 서버측에서 Digest 인증 플러그인을 통해서도 계산되며 클라이언트에서 제공하는 다이제스트 값과 비교됩니다. 다이제스트 값이 일치하면 사용자가 인증됩니다. 이렇게 하려면 디렉토리 서버가 일반 텍스트로 사용자의 비밀번호에 액세스해야 합니다. Sun Java System Directory Server에는 역변환 가능한 비밀번호 플러그인이 있으며, 이는 데이터를 암호화된 형태로 저장하여 나중에 원래 형태로 해독할 수 있는 대칭 암호화 알고리즘을 사용합니다. 오직 Directory Server만이 데이터의 키를 보유합니다.

LDAP 기반 Digest 인증의 경우 서버에 포함된 역전환 가능 비밀번호 플러그인과 digestauth 특정 플러그인을 사용하도록 설정해야 합니다. 웹 서버가 Digest 인증을 처리하도록 구성하려면 `dbswitch.conf`에 있는 데이터베이스 정의의 `digestauth` 등록 정보를 설정해야 합니다.

ACL 방법을 지정하지 않는 경우, 서버는 인증이 필요하면 Digest 또는 Basic 인증을 사용하고 인증이 필요하지 않으면 Basic 인증을 사용합니다. 이것이 가장 많이 사용되는 방법입니다.

표 7–1 Digest 인증 질문 생성

ACL 방법	인증 데이터베이스가 지원하는 Digest 인증	인증 데이터베이스가 지원하지 않는 Digest 인증
"default" 지정된 사항 없음	digest 및 basic	basic
"basic"	basic	basic

표 7–1 Digest 인증 질문 생성 *(계속)*

ACL 방법	인증 데이터베이스가 지원하는 Digest 인증	인증 데이터베이스가 지원하지 않는 Digest 인증
"digest"	digest	ERROR

`method = digest`로 설정된 ACL을 처리할 경우 서버는 다음을 수행하여 인증을 시도합니다.

- 인증 요청 헤더 확인. 없는 경우 Digest 시도를 포함하는 401 응답이 생성되며 프로세스는 중지됩니다.
- 인증 유형 확인. 인증 유형이 Digest인 경우:
  - nonce를 확인합니다. 유효하지 않은 경우 이 서버가 새 nonce를 생성하고 401 응답이 생성되며 프로세스가 중지됩니다. 오래된 경우 `stale=true`로 설정된 401 응답이 생성되며 프로세스가 중지됩니다.

    `server_root/https-server_name/config/`에 있는 `magnus.conf` 파일에서 `DigestStaleTimeout` 매개 변수 값을 변경하여 nonce가 새로운 상태를 유지하는 시간을 구성할 수 있습니다. 이 값을 설정하려면 `magnus.conf`에 다음 줄을 추가합니다.

    ```
    DigestStaleTimeout seconds
    ```

    여기서 *seconds*는 nonce가 새로운 상태를 유지하는 시간(초)을 나타냅니다. 지정된 시간(초)이 경과하면 nonce가 만료되며 사용자에게 새로운 인증이 요구됩니다.
  - 영역 확인. 일치하지 않는 경우 401 응답이 생성되며 프로세스가 중지됩니다.
  - 인증 디렉토리가 LDAP 기반인 경우 LDAP 디렉토리에 사용자가 있는지 확인하며 인증 디렉토리가 파일 기반인 경우 파일 데이터베이스에 사용자가 있는지 확인합니다. 찾을 수 없는 경우 401 응답이 생성되며 프로세스가 중지됩니다.
  - 디렉토리 서버 또는 파일 데이터베이스에서 요청 다이제스트 값을 가져오고 클라이언트의 요청 다이제스트와 일치하는지 확인합니다. 일치하지 않는 경우 401 응답이 생성되며 프로세스가 중지됩니다.
  - Authorization-Info 헤더를 만들고 이를 서버 헤더에 삽입합니다.

호스트-IP용 액세스 제어 설정

Administration Server 또는 웹 사이트의 파일 및 디렉토리를 특정 컴퓨터를 이용하는 사용자만 사용할 수 있도록 설정하여 액세스를 제한할 수 있습니다. 허용 또는 거부할 컴퓨터의 호스트 이름이나 IP 주소를 지정합니다. 여러 대의 컴퓨터 또는 전체 네트워크를 지정하려면 와일드카드 패턴을 사용합니다. 호스트-IP 인증을 사용하는 파일 또는 디렉토리 액세스는 사용자가 알 수 없게 진행됩니다. 사용자는 사용자 이름이나 비밀번호를 입력하지 않고 파일과 디렉토리에 액세스할 수 있습니다.

특정 컴퓨터를 여러 사람이 사용할 수 있기 때문에 호스트-IP 인증을 사용자 그룹 인증과 함께 사용하면 더욱 효과적일 수 있습니다. 두 가지 인증 방법이 모두 사용되는 경우 액세스할 때 사용자 이름과 비밀번호가 필요합니다.

호스트-IP 인증의 경우 서버에 DNS를 구성할 필요가 없습니다. 호스트-IP 인증을 사용하도록 선택한 경우 DNS가 네트워크에서 실행되어야 하며 서버가 DNS를 사용하도록 구성되어야 합니다. Server Manager의 Preferences 탭에 있는 Performance Tuning 페이지에서 서버의 DNS를 사용하도록 설정할 수 있습니다.

DNS를 사용하도록 설정하면 서버가 DNS 조회를 수행해야 하므로 서버의 성능이 저하됩니다. DNS 조회가 서버 성능에 미치는 영향을 줄이려면 모든 요청의 IP 주소를 확인하는 대신 액세스 제어 및 CGI에 대해서만 IP 주소를 확인합니다. 이렇게 하려면 `obj.conf` 파일에서 `iponly=1`을 `AddLog fn="flex-log" name="access"`에 추가합니다.

```
AddLog fn="flex-log" name="access" iponly=1
```

ACL 사용자 캐시 구성

기본적으로 서버는 ACL 사용자 캐시에 사용자 및 그룹 인증 결과를 캐시합니다. `magnus.conf` 파일의 `ACLCacheLifetime` 지시문을 사용하여 ACL 사용자 캐시가 유효한 시간을 제어할 수 있습니다. 캐시에 있는 항목을 참조할 때마다 시간이 계산되고 `ACLCacheLifetime`과 비교하여 확인됩니다. 항목의 시간이 `ACLCacheLifetime` 이상이면 해당 항목은 사용되지 않습니다. 기본값은 120초입니다. 값을 0으로 설정하면 캐시가 해제됩니다. 이 값에 큰 수를 사용하면 LDAP 항목을 변경할 때마다 서버를 다시 시작해야 합니다. 예를 들어, 이 값을 120초로 설정하면 서버가 최대 2분까지 LDAP 디렉토리와 동기화되지 않을 수 있습니다. LDAP 디렉토리가 자주 변경되지 않는 경우에만 큰 값을 사용하십시오.

`ACLUserCacheSize`의 `magnus.conf` 매개 변수를 사용하면 캐시에 보유할 수 있는 최대 항목 수를 구성할 수 있습니다. 이 매개 변수의 기본값은 200입니다. 새 항목은 목록의 앞에 추가되며 목록의 끝에 있는 항목은 캐시가 최대 크기에 도달하면 재활용되어 새로운 항목이 됩니다.

또한 `magnus.conf` 매개 변수 `ACLGroupCacheSize`를 사용하여 각 사용자 항목마다 캐시될 수 있는 그룹 구성원의 최대 수를 설정할 수 있습니다. 이 매개 변수의 기본값은 4입니다. 그룹에 있는 사용자가 구성원이 아닌 경우 캐시되지 않으며 이로 인해 모든 요청에 대해 LDAP 디렉토리 액세스가 여러 번 발생하게 됩니다.

ACL 파일 지시문에 대한 자세한 내용은 *NSAPI Developer's Guide*를 참조하십시오.

ACL 캐시 등록 정보 설정

CLI를 통해 ACL 캐시 등록 정보를 설정하려면 다음 명령을 실행합니다.

```
wadm> set-acl-cache-prop --user=admin --password-file=admin.pwd --host=serverhost
--port=8989 --config=config1 property=value
```

CLI 참조 `set-acl-cache-prop`(1)를 참조하십시오.

설정할 수 있는 유효한 등록 정보는 다음과 같습니다.

- `enabled` — 서버에서 파일 내용과 메타 정보를 캐시하는지 여부를 나타냅니다. 기본값은 true입니다.
- `max-age` — 파일 내용과 메타 정보를 캐시하는 최대 시간(초)입니다. 값의 범위는 0.001 - 3600입니다.
- `max-groups-per-user` — 서버에서 구성원 정보를 캐시하는 사용자별 최대 그룹 수입니다. 값의 범위는 1 - 1024입니다.
- `max-age` — 인증 정보를 캐시하는 최대 시간(초)입니다. 값의 범위는 0.001 - 3600입니다.

액세스 제어 구성

서버는 로컬에 저장된 액세스 제어 목록(ACL)을 사용하여 인증 및 권한을 지원합니다. 이 ACL에는 자원에 대해 사용자가 가지는 액세스 권한이 설명되어 있습니다. 예를 들어, ACL에 있는 항목에서 John이라는 이름의 사용자에게 `misc`라는 특정 폴더에 대해 `read` 권한을 부여할 수 있습니다.

이 절에서는 웹 사이트의 파일 및 디렉토리에 대한 액세스를 제한하는 과정에 대해 설명합니다. 모든 서버에 대한 전역 액세스 제어 규칙을 설정할 수도 있고 특정 서버에 대한 개별 규칙을 설정할 수도 있습니다. 예를 들어, 인력 관리 부서에서는 모든 인증된 사용자가 자신의 연봉 데이터를 볼 수 있으나 오직 인력 관리 부서의 연봉을 담당하는 직원만 데이터를 업데이트할 수 있도록 제한하는 ACL을 만들 수 있습니다.

서버가 지원하는 핵심 ACL에는 기본, SSL 및 다이제스트와 같은세 가지 유형의 인증이 있습니다.

액세스 제어 설정을 편집하려면 다음 작업을 수행하십시오.

1. **구성 탭**을 누르고 구성을 선택합니다.
2. **보안 하위 탭** > **액세스 제어 하위 탭**을 누릅니다.
3. **ACL 추가 버튼**을 눌러 새 ACL을 추가하거나 기존 ACL을 눌러 설정을 편집합니다.

액세스 제어 목록(ACL) 추가

다음 절에서는 구성에 새 ACL을 추가하는 과정에 대해 설명합니다.

1. **구성 탭**을 누르고 구성을 선택합니다.
2. **액세스 제어 하위 탭** > **액세스 제어 목록 하위 탭**을 누릅니다.

3. **새로 만들기 버튼**을 눌러 새 ACL을 추가합니다.

다음 매개 변수를 구성합니다.

표 7–2 ACL 매개 변수

매개 변수	설명
자원	이름 지정/URI/경로. 액세스 제한을 설정해야 하는 자원의 유형을 선택하고 값을 지정합니다. URI 자원 예 — "/sales". 경로 자원 예 — "/usr/sun/server4/docs/cgi-bin/*".
인증 DB	**인증 데이터베이스**를 사용하여 서버가 사용자 인증에 사용할 데이터베이스를 선택할 수 있습니다. 기본값은 **keyfile**입니다.
인증 방법	1. **Basic** — HTTP Basic 방법을 사용하여 클라이언트에서 인증 정보를 가져옵니다. 서버에 대해 SSL을 사용하는 경우 사용자 이름 및 암호는 네트워크를 통해서만 암호화됩니다. 2. **SSL** — 클라이언트 인증서를 사용하여 사용자를 인증합니다. 이 방법을 사용하려면 서버에서 SSL을 사용해야 합니다. 암호화를 사용하는 경우 Basic 방법과 SSL 방법을 함께 사용할 수 있습니다. 3. **Digest** — 사용자 이름과 비밀번호를 일반 텍스트로 보내지 않고 브라우저가 사용자 이름과 비밀번호를 기준으로 인증하는 방법을 제공하는 인증 기법을 사용합니다. 브라우저는 *MD5* 알고리즘을 사용하여 Web Server에서 제공한 사용자 비밀번호 및 일부 정보를 사용하는 다이제스트 값을 만듭니다. Digest를 사용하려면 배후의 auth-db에서도 Digest를 지원해야 합니다. 즉, digestfile을 사용하는 파일 auth-db 또는 Digest 인증 플러그인이 설치된 경우에 한해 LDAP auth-db를 의미합니다. 4. **기타** — 액세스 제어 API를 사용하여 만든 사용자 정의 방법을 사용합니다.
인증 확인 프롬프트	**인증 확인 프롬프트** 옵션을 사용하여 인증 대화 상자에 표시되는 메시지 텍스트를 입력할 수 있습니다. 이 텍스트를 사용하여 사용자가 입력해야 하는 내용을 설명할 수 있습니다. 브라우저에 따라 사용자는 프롬프트의 처음 40자 정도만 볼 수 있습니다. 웹 브라우저는 일반적으로 사용자 이름과 비밀번호를 캐시하고 프롬프트 텍스트에 연결합니다. 사용자가 동일한 프롬프트를 가진 서버의 파일 및 디렉토리에 액세스하는 경우에는 사용자 이름과 비밀번호를 다시 입력하지 않아도 됩니다. 특정 파일 및 디렉토리에 대해 사용자 인증을 원하는 경우 간단히 해당 자원에 대한 ACL용 프롬프트를 변경하면 됩니다.

표 7–2 ACL 매개 변수 *(계속)*

매개 변수	설명
거부된 액세스 응답	자원에 대한 액세스가 거부되었을 때의 응답 작업을 지정합니다. 1. 기본 메시지로 응답 — 서버의 표준 액세스 거부 메시지를 표시하는 경우 이 옵션을 선택합니다. 2. URL로 응답 — 요청을 다른 외부 URL 또는 오류 페이지로 전달하는 경우 이 옵션을 선택합니다.

주 – CLI 사용

CLI를 통해 ACL을 추가하려면 다음 명령을 실행합니다.

```
wadm> set-acl --user=admin --password-file=admin.pwd
--host=serverhost --port=8989 --vs=config1_vs_1 --config=config1
--aclfile=aclfile1
```

CLI 참조 `set-acl(1)`을 참조하십시오.

액세스 제어 항목(ACE) 추가

이 절에서는 선택한 구성에 대해 새 액세스 제어 항목(ACE)을 추가하는 과정에 대해 설명합니다.

1. **구성 탭**을 누르고 구성을 선택합니다.
2. **액세스 제어 하위 탭** > **액세스 제어 목록 하위 탭**을 누릅니다.
3. **새로 만들기 버튼**을 누릅니다.
4. 액세스 제어 항목(ACE) 아래에 있는 **새로 만들기 버튼**을 누릅니다.

다음 ACE 매개 변수를 구성합니다.

표 7–3 ACE 매개 변수

매개 변수	설명
액세스	■ **허용**은 사용자 또는 시스템이 요청된 자원에 액세스할 수 있음을 나타냅니다. ■ **거부**는 사용자 또는 시스템이 자원에 액세스할 수 없음을 나타냅니다. 서버는 ACE(Access Control Expressions) 목록 전체를 확인하여 액세스 권한을 판단합니다.

표 7–3 ACE 매개 변수 *(계속)*

매개 변수	설명
사용자	1.**모든 사용자** — 인증이 없습니다. 모든 사용자에게 액세스를 허용합니다. 2. **인증 DB의 모든 사용자** — 인증 데이터베이스에 지정된 모든 사용자에게 액세스를 허용합니다. 3. **인증 DB의 다음 사용자만** — 인증 DB에서 선택된 사용자에 대해서만 액세스를 허용합니다. 이름, 성 및 전자 메일 주소와 같은 공통 속성을 기반으로 인증 DB를 쿼리할 수 있습니다.
그룹	그룹 인증을 사용하면 사용자가 액세스 제어 규칙에 지정된 자원에 액세스하기 전에 사용자 이름 및 비밀번호를 입력하라는 프롬프트가 표시됩니다. 이 옵션을 사용하여 특정 그룹에 대한 액세스를 제한합니다.

표 7–3 ACE 매개 변수 *(계속)*

매개 변수	설명
시작 호스트	요청을 보내는 컴퓨터를 기준으로 Administration Server 또는 웹 사이트에 대한 액세스를 제한할 수 있습니다. 요청을 보내는 컴퓨터를 기준으로 Administration Server 또는 웹 사이트에 대한 액세스를 제한할 수 있습니다. ■ 모든 위치는 모든 사용자 및 시스템에 대한 액세스를 허용합니다. ■ 다음 위치에서만은 특정 호스트 이름 또는 IP 주소에 대한 액세스를 제한할 수 있습니다. 다음 위치에서만 옵션을 선택하는 경우에는 호스트 이름 또는 IP 주소 필드에 와일드카드 패턴 또는 쉼표로 분리된 목록을 입력합니다. 호스트 이름을 기준으로 제한하는 것이 IP 주소를 기준으로 제한하는 것보다 유연성이 큽니다. 사용자의 IP 주소가 변경되더라도 목록을 업데이트할 필요가 없습니다. 그러나 IP 주소를 기준으로 제한하는 것이 더욱 안전합니다. 연결된 클라이언트에 대한 DNS 조회가 실패하면 호스트 이름 제한은 사용할 수 없습니다. 컴퓨터의 호스트 이름 또는 IP 주소를 검색하는 와일드카드 패턴에는 * 와일드카드만 사용할 수 있습니다. 예를 들어, 특정 도메인에 있는 모든 컴퓨터를 허용하거나 거부하려면 해당 도메인에 있는 모든 호스트와 일치하는 와일드카드 패턴(예: `*.sun.com`)을 입력합니다. Administration Server에 액세스하는 수퍼유저를 위해 다른 호스트 이름 및 IP 주소를 설정할 수 있습니다. 호스트 이름의 경우 *는 반드시 이름의 전체 구성 요소를 대체해야 합니다. 즉, `*.sun.com`은 사용할 수 있지만 `*users.sun.com`은 사용할 수 없습니다. 호스트 이름에 *를 사용하는 경우 가장 왼쪽에 표시해야 합니다. 예를 들어, `*.sun.com`은 사용할 수 있지만 `users.*.com`은 사용할 수 없습니다. IP 주소의 경우 *는 반드시 주소의 전체 바이트를 대체해야 합니다. 예를 들어 `198.95.251.*`는 사용할 수 있지만 `198.95.251.3*`는 사용할 수 없습니다. IP 주소에 *를 사용하는 경우 가장 오른쪽에 표시해야 합니다. 예를 들어 `198.*`는 사용할 수 있지만 `198.*.251.30`은 사용할 수 없습니다.

표 7–3 ACE 매개 변수 *(계속)*

매개 변수	설명
권한	액세스 권한은 웹 사이트의 파일 및 디렉토리에 대한 액세스를 제한합니다. 모든 액세스 권한을 허용 또는 거부하는 규칙 외에 부분적인 액세스 권한을 허용 또는 거부하는 규칙을 지정할 수 있습니다. 예를 들어, 사용자에게 파일에 대한 읽기 전용 액세스 권한을 허용하면 사용자가 정보를 볼 수 있지만 파일을 변경할 수는 없습니다. ■ 모든 액세스 권한은 기본값이며 모든 권한을 허용하거나 거부합니다. ■ 다음 권한만에서는 허용 또는 거부할 권한을 조합하여 선택할 수 있습니다. ■ **읽기** 권한은 사용자가 HTTP 메소드인 GET, HEAD, POST 및 INDEX를 비롯한 파일을 볼 수 있도록 허용합니다. ■ **쓰기**는 사용자가 HTTP 메소드 PUT,DELETE,MKDIR,RMDIR 및MOVE를 포함하여 파일을 변경 및 삭제할 수 있게 합니다. 파일을 삭제하려면 사용자에게 반드시 쓰기 및 삭제 권한이 있어야 합니다. ■ **실행**은 사용자가 서버측 응용 프로그램(예: CGI 프로그램, Java 애플릿 및 에이전트)을 실행할 수 있게 합니다. ■ **삭제**는 쓰기 권한을 가진 사용자가 파일 또는 디렉토리를 삭제할 수 있게 합니다. ■ **목록**은 사용자가 `index.html` 파일을 포함하지 않는 디렉토리의 파일 목록에 액세스할 수 있도록 허용합니다. ■ **정보**는 사용자가 URI에 대한 정보(예: `http_head`)를 받을 수 있게 합니다.
계속	서버는 ACE(Access Control Expression) 목록 전체를 확인하여 액세스 권한을 판단합니다. 예를 들어, 첫 번째 ACE는 보통 모든 사용자를 거부합니다. 첫 번째 ACE가 "계속"으로 설정된 경우 서버는 목록에 있는 두 번째 ACE를 확인하며, 일치하는 경우 다음 ACE를 사용합니다. 계속이 선택되지 **않은** 경우 자원에 대한 모든 사용자의 액세스가 거부됩니다. 서버는 일치하지 않는 ACE를 발견하거나 일치하지만 계속으로 설정되지 않은 ACE를 발견할 때까지 계속해서 목록을 검색합니다. 마지막으로 일치되는 ACE에 따라 액세스의 허용 또는 거부가 결정됩니다.

.htaccess 파일 사용

서버는 `.htaccess` 동적 구성 파일을 지원합니다. 사용자 인터페이스를 통해 또는 구성 파일을 직접 변경하여 `.htaccess` 파일을 사용 가능으로 설정할 수 있습니다.

`.htaccess` 파일을 서버의 표준 액세스 제어와 함께 사용할 수 있습니다. `PathCheck` 지시문의 순서에 관계없이 표준 액세스 제어는 모든 `.htaccess` 액세스 제어에 우선하여

적용됩니다. 사용자 그룹 인증이 "Basic"인 경우에는 사용자 인증에 표준 및 .htaccess 액세스 제어를 모두 요구하면 안 됩니다. 표준 서버 액세스 제어를 통해 SSL 클라이언트 인증을 사용하는 동시에 .htaccess 파일을 통해 HTTP "Basic" 인증을 요구할 수는 있습니다.

.htaccess 파일을 사용하도록 설정하면 서버가 자원을 서비스하기 전에 .htaccess 파일을 확인합니다. 서버는 우선 자원과 동일한 디렉토리에서 시작하여 그 상위 디렉토리, 다시 문서 루트까지 .htaccess 파일을 찾습니다. 예를 들어, Primary Document Directory가 /sun/server/docs로 설정되어 있고 클라이언트가 /sun/server/docs/reports/index.html을 요청하는 경우 서버는 /sun/server/docs/reports/.htaccess와 /sun/server/docs/.htaccess에서 .htaccess 파일을 확인하게 됩니다.

참고로 관리자는 서버의 추가 문서 디렉토리와 CGI 디렉토리 기능을 사용하여 대체 문서 루트를 정의할 수 있습니다. 대체 문서 루트가 있으면 .htaccess 파일 처리가 달라집니다. 예를 들어, 서버의 기본 문서 디렉토리는 /sun/server/docs로 설정되고 CGI 프로그램은 /sun/server/docs/cgi-bin/program.cgi에 있는 경우를 가정합니다. CGI를 파일 유형으로 사용 설정하면 클라이언트가 CGI 프로그램을 요청할 때 서버는 /sun/server/docs/.htaccess와 /sun/server/docs/cgi-bin/.htaccess의 내용을 모두 확인합니다. 그러나 대신 CGI 디렉토리를 /sun/server/docs/cgi-bin으로 구성하면 서버가 /sun/server/docs/cgi-bin/.htaccess는 확인하지만 /sun/server/docs/.htaccess는 확인하지 않습니다. 이는 /sun/server/docs/cgi-bin을 CGI 디렉토리로 지정하면 이 디렉토리가 대체 문서 루트가 되기 때문입니다.

서버에 대한 서비스 거부 공격 방지

서비스 거부(DoS) 공격은 악의적인 서버 사용자가 합법적인 사용자의 서비스 사용을 방해하려고 시도하는 것입니다. 이러한 공격은 다음과 같은 방법으로 시작될 수 있습니다.

- 특정 웹 자원에 대한 요청을 서버로 계속 보냅니다.

요청 빈도가 매우 높은 경우 Sun Java System Web Server는 자주 액세스되는 URI를 모니터하여 DoS 공격을 감지한 다음 요청을 거부할 수 있습니다.

다음 절에서는 가상 서버 수준에서 DoS 공격을 방지할 수 있는 방법에 대해 설명합니다.

서버에 대한 요청 제한

요청 제한을 구성하고 가상 서버당 최대 연결 수를 모니터하여 서버에서 서비스 거부 공격을 방지하도록 조정할 수 있습니다. 이런 값을 몇 개 구성하면 서버 성능에 영향을 줄 수 있습니다.

서버의 요청 제한을 구성하려면 **구성** > **가상 서버** > **서버 설정** > **요청 제한**을 누릅니다. 다음 표에 나열된 매개 변수를 구성합니다.

표 7–4 요청 제한 구성

매개 변수	설명
요청 제한	이 가상 서버에 대해 제한을 활성화/비활성화합니다. 요청 제한 옵션은 기본적으로 비활성화됩니다.
최대 연결 수	이 가상 서버에 허용되는 최대 동시 연결 수입니다.
최대 RPS	클라이언트에서 초당 허용되는 최대 요청 수입니다.
RPS 계산 간격	초당 평균 요청(RPS)을 계산하는 시간 간격입니다. 기본값은 30초입니다.
계속 조건	차단된 요청 유형을 다시 처리하기 위해 충족해야 하는 조건을 결정합니다. **무응답** — 서비스를 다시 시작하려면 거부된 요청이 후속 간격에서 0이 되어야 합니다. **임계값** — 서비스를 다시 시작하려면 거부된 요청 비율이 RPG 임계값보다 작아야 합니다. 기본값은 임계값입니다.
오류 코드	차단된 요청에 대해 사용할 HTTP 상태 코드입니다. 기본 코드는 HTTP 503 — 사용할 수 없는 서비스입니다.
속성 모니터	모니터할 선택적 요청 속성

주 – CLI 사용

CLI를 통해 서버 요청을 제한하려면 다음 명령을 실행합니다.

```
wadm> enable-request-limits --user=admin --password-file=admin.pwd
--host=serverhost --port=8989 --config=config1 --vs=config1_vs_1
```

CLI 참조 `enable-request-limits(1)`를 참조하십시오.

▼ 최대 연결 수를 제한하는 방법

최대 동시 연결 수를 제한할 수 있습니다. 최소한의 지정된 요청 수를 처리하는 동안 일치하는 요청이 수신되면 요청이 거부됩니다. 요청 거부는 특정 시간에 한하여 이루어집니다. 동시 요청 수가 이 제한보다 작아지면 바로 새 요청을 처리할 수 있게 됩니다.

1 구성 탭을 누릅니다.

2 목록에서 구성을 선택합니다.

3 가상 서버 탭 아래에서 가상 서버를 선택합니다.

4 서버 설정 > 요청 제한을 누릅니다.

5 최대 연결 섹션에 값을 입력합니다.

◆ ◆ ◆ 8 장

8 사용자 및 그룹 관리

이 장에서는 Sun Java System Web Server에 액세스할 수 있는 사용자 및 그룹을 추가, 삭제 및 편집하는 방법에 대해 설명합니다.

사용자 및 그룹에 대한 정보 액세스

Administration Server에서는 사용자 계정, 그룹 목록, 액세스 권한(ACL), 조직 단위 및 기타 사용자별 및 그룹별 정보에 대한 응용 프로그램 데이터에 액세스할 수 있습니다.

사용자 및 그룹 정보는 일반 파일에 텍스트 형식으로 저장되거나 LDAP(Lightweight Directory Access Protocol)를 지원하는 Sun Java System Directory Server와 같은 디렉토리 서버에 저장됩니다. LDAP는 개방형 디렉토리 액세스 프로토콜로 TCP/IP를 통해 실행되며 전역 규모의 수백만 항목을 수용하도록 확장될 수 있습니다.

디렉토리 서비스 정보

Sun Java System Directory Server와 같은 디렉토리 서버를 사용하면 단일 응용 프로그램에서 모든 사용자 정보를 관리할 수 있습니다. 또한 사용자가 쉽게 액세스할 수 있는 여러 네트워크 위치에서 디렉토리 정보를 검색할 수 있도록 디렉토리 서버를 구성할 수 있습니다.

Web Server 7.0에서는 서로 다른 세 가지 유형의 디렉토리 서비스를 구성하여 사용자 및 그룹을 인증하고 권한을 부여할 수 있습니다. 다른 디렉토리 서비스가 구성되어 있지 않은 경우 새로 만드는 디렉토리 서비스의 값은 유형에 관계없이 `default`로 설정됩니다.

디렉토리 서비스를 만들면 디렉토리 서비스 세부 정보로 `server.xml` 파일이 업데이트됩니다.

디렉토리 서비스 유형

Web Server 7.0이 지원하는 여러 디렉토리 서비스 유형은 다음과 같습니다.

- **LDAP** — 사용자 및 그룹 정보를 LDAP 기반 디렉토리 서버에 저장합니다.
- **키 파일** — 키 파일은 해시 형식의 사용자 비밀번호와 사용자가 속한 그룹 목록을 포함하는 텍스트 파일입니다. 키 파일에 저장된 사용자 및 그룹은 **파일** 영역에 의한 인증 및 권한 부여에만 사용되며 시스템 사용자 및 그룹과는 관계가 없습니다.

  키 파일 형식은 HTTP Basic 인증을 사용하려는 경우에만 사용할 수 있습니다.
- **다이제스트 파일** — 암호화된 사용자 이름 및 비밀번호를 기반으로 사용자 및 그룹 정보를 저장합니다.

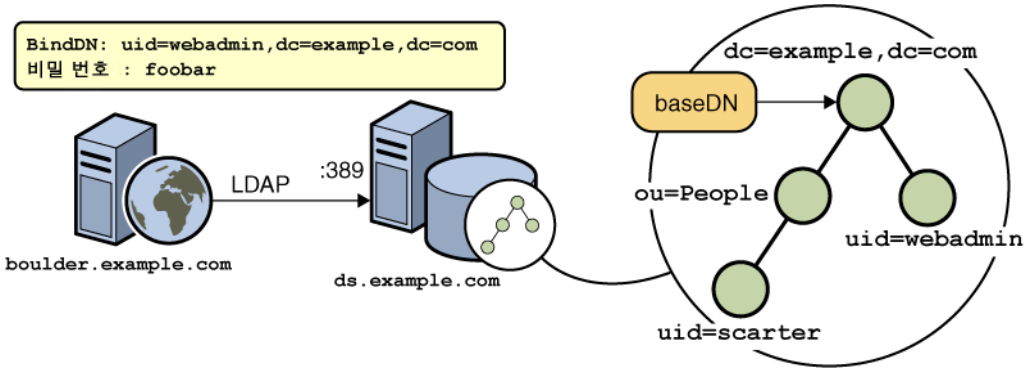
  다이제스트 파일 형식은 기본적으로 HTTP Digest 인증 사용을 지원합니다. 하지만 Basic 인증도 지원하기 때문에 두 가지 인증 방법 모두에 대해 사용할 수 있습니다.

주 – 분산 관리를 설정하려는 경우 기본 디렉토리 서비스는 LDAP 기반 디렉토리 서비스여야 합니다.

DN(Distinguished Name) 이 해

사용자는 회사 직원과 같이 LDAP 데이터베이스에 있는 개인입니다. 그룹은 공통 속성을 공유하는 둘 이상의 사용자입니다. 조직 단위는 회사 내의 하위 부서입니다.

기업의 각 사용자와 그룹은 고유 이름(DN) 속성으로 나타냅니다. DN 속성은 연결된 사용자, 그룹 또는 객체에 대한 식별 정보가 있는 문자열입니다. 사용자 또는 그룹 디렉토리 항목을 변경하는 경우 항상 DN을 사용합니다. 예를 들어 디렉토리 항목 작성 또는 수정, 액세스 제어 설정, 메일이나 게시 등의 응용 프로그램에 대한 사용자 계정 설정과 같은 작업을 할 때 항상 DN 정보를 지정해야 합니다.



위 그림은 샘플 DN 표현을 나타냅니다. 다음 예는 전형적인 Sun Microsystems 직원의 DN을 나타냅니다.

```
uid=doe,e=doe@sun.com,cn=John Doe,o=Sun Microsystems Inc.,c=US
```

이 예에서 각 등호(=) 앞의 약자는 다음을 의미합니다.

- uid: 사용자 아이디
- e: email address
- cn: 사용자의 공통 이름
- o: 조직
- c: 국가

DN에는 다양한 이름 값 쌍이 있을 수 있습니다. 이는 LDAP를 지원하는 디렉토리의 인증서 개체 및 항목을 확인하는 데 사용됩니다.

LDIF 사용

현재 디렉토리가 없는 경우나 기존 디렉토리에 새 하위 트리를 추가하려는 경우에는 Directory Server의 Administration Server LDIF 가져오기 기능을 사용할 수 있습니다. 이 기능은 LDIF가 포함된 파일을 받아서 LDIF 항목에서 디렉토리를 구축하거나 새 하위 트리를 만듭니다. Directory Server의 LDIF 내보내기 기능을 사용하여 현재 디렉토리를 LDIF로 내보낼 수도 있습니다. 이 기능은 디렉토리를 나타내는 LDIF 형식의 파일을 만듭니다. `ldapmodify` 명령과 적절한 LDIF 업데이트문을 함께 사용하여 항목을 추가 또는 편집합니다.

LDIF를 사용하여 데이터베이스에 항목을 추가하려면 먼저 `LDIF` 파일에서 항목을 정의한 다음 Directory Server에서 `LDIF` 파일을 가져옵니다.

인증 데이터베이스 작업

auth-db라고도 하는 **인증 데이터베이스**는 알려진 사용자의 데이터베이스와 해당 데이터베이스에 대해 클라이언트 요청을 인증하기 위해 사용되는 기법을 나타냅니다. 서버에는 동시에 여러 개의 auth-db 항목이 구성될 수 있으며, 구성된 항목들의 유형이 같을 수도 있습니다. auth-db 사용자 데이터베이스는 ACL 처리 모듈에서 사용됩니다.

서버에서는 다음 인증 데이터베이스를 사용할 수 있습니다.

1. **LDAP** — 사용자 데이터는 Sun Java System Directory Server와 같은 LDAP 디렉토리 서버에 저장됩니다.
2. **파일** — 사용자 데이터는 디스크 파일에 저장됩니다. 이 auth-db는 중앙 집중식 사용자 관리를 사용할 수 없는(또는 바람직하지 않은) 개발 또는 소규모 배포의 경우 특히 편리합니다. 파일 auth-db에서는 다양한 파일 형식을 지원합니다.
   a. keyfile — keyfile 형식은 사용자 목록(및 각 사용자의 선택 그룹 구성원)을 저장합니다. 비밀번호는 단방향(복구할 수 없음) 해시로 저장됩니다. 이 형식이 기본 형식입니다.
   b. digestfile — digestfile은 keyfile과 매우 비슷하며 HTTP Digest 인증 방법도 지원합니다.
   c. htaccess — 기존 형식일 뿐이며, 새로운 설치나 새 사용자 추가에는 사용하지 말아야 합니다.
3. **PAM** — PAM은 Sun Java System Web Server 7.0에서 지원되는 새로운 auth-db입니다. PAM auth-db는 인증을 Solaris PAM 스택에 위임하여 웹 서버 시스템에 있는 기존 Solaris 사용자도 웹 서버에 인증될 수 있도록 합니다.

주 – PAM auth-db는 Solaris 9 및 10 이상에서만 지원되며 웹 서버 인스턴스가 루트로 실행 중이어야 합니다.

인증 데이터베이스 만들기

관리 콘솔을 통해 인증 데이터베이스를 만들려면 **구성 > 구성 이름 > 액세스 제어 > 인증 데이터베이스 > 새로 만들기** 버튼을 클릭합니다. 필드 설명은 관리 콘솔 인라인 도움말을 참조하십시오. 선택한 인증 데이터베이스에 따라 필드가 달라집니다. 예를 들어 PAM 기반 인증 DB의 경우에는 인증 DB 이름만 필요합니다.

인증 데이터베이스를 만들기 위해 필요한 옵션은 다음과 같습니다.

LDAP	■ 인증 데이터베이스 이름 ■ 호스트 이름 ■ 포트 ■ 기본 DN
키 파일	■ 인증 데이터베이스 이름 ■ 파일 경로
다이제스트 파일	■ 인증 데이터베이스 이름 ■ 파일 경로
PAM	■ 인증 데이터베이스 이름

CLI를 통해 인증 데이터베이스를 만들려면 다음 명령을 실행합니다.

```
wadm> create-authdb --user=admin --password-file=admin.pwd
--host=serverhost --port=8989 --config=config1
--url=ldap://ldapserver.com:20002/dc=xxx,dc=sun,dc=com LDAP1
```

CLI 참조 `create-authdb`(1)를 참조하십시오.

위의 예에서 인증 데이터베이스에는 URL이 지정되었습니다. 인증 데이터베이스의 유형은 이 URL 체계로 지정됩니다. 예를 들어 `ldap://ds.example.come/dc=example,dc=com`은 LDAP 디렉토리 서버를 인증 데이터베이스로 구성합니다.

사용자 및 그룹 설정

Administration Server에서 LDAP 및 파일 auth-db 유형 모두에 대해 사용자 계정, 그룹 목록, 액세스 권한, 조직 단위, 기타 사용자별 및 그룹별 정보를 편집할 수 있습니다.

▼ 사용자 추가

1 구성을 선택합니다.

구성 목록에서 구성을 선택합니다. 구성 탭을 눌러 목록을 가져옵니다.

2 액세스 제어 > 사용자 탭을 누릅니다.

3 새로 만들기 버튼을 누릅니다.

4 사용자 정보를 추가합니다.

사용자 아이디 및 비밀번호를 입력합니다. 사용자가 속한 그룹을 입력합니다(선택 사항). 사용자 아이디는 반드시 고유해야 합니다. LDAP 기반 인증 DB 경우 Administration

Server는 사용자 아이디가 사용 중인지 확인하기 위해 검색 기준(**기본** DN) 아래의 전체 디렉토리를 검색하여 사용자 아이디가 고유함을 확인합니다. 하지만 디렉토리 서버 `ldapmodify` 명령줄 유틸리티(사용 가능한 경우)를 사용하여 사용자를 만들면 사용자 아이디가 고유한지 확인되지 않습니다.

주 – CLI 사용

CLI를 통해 사용자를 만들려면 다음 명령을 실행합니다.

```
wadm> create-user --user=admin --password-file=admin.pwd --host=serverhost
--port=8989 --config=config1 --authdb=KEYFILE1 --full-name=keyfile-config1-u1
keyfile-config1-u1
```

CLI 참조 `create-user`(1)를 참조하십시오.

▼ 그룹 추가

1. **구성을 선택합니다.**
   구성 목록에서 구성을 선택합니다. 구성 탭을 눌러 목록을 가져옵니다.
2. **액세스 제어 > 그룹 탭을 누릅니다.**
3. **새로 만들기 버튼을 누릅니다.**
4. **그룹 이름을 입력합니다.**
5. **그룹에 사용자 추가 섹션에서 기존 사용자를 검색하여 그룹에 추가합니다.**

주 – keyfile 또는 digestfile과 같은 인증 데이터베이스에서 그룹을 만들려면 사용자를 한 명 이상 지정해야 합니다.

주 – CLI 사용

CLI를 통해 그룹을 만들려면 다음 명령을 실행합니다.

```
wadm> create-group --user=admin --password-file=admin.pwd --host=serverhost
--port=8989 --config=config1 --authdb=LDAP1 group1
```

CLI 참조 `create-group`(1)을 참조하십시오.

▼ 사용자 삭제

1 **구성을 선택합니다.**
구성 목록에서 구성을 선택합니다. 구성 탭을 눌러 목록을 가져옵니다.

2 **액세스 제어 > 사용자 탭을 누릅니다.**

3 **사용자를 삭제할 인증 데이터베이스를 선택합니다.**

4 **사용자 검색 입력란에 사용자 아이디를 입력하고 검색 버튼을 누릅니다.**

5 **사용자 아이디 열에서 사용자를 선택하고 삭제 버튼을 누릅니다.**

주의 – 사용자를 삭제한 후 그룹에 구성원이 남지 않게 되는 경우 keyfile/digestfile 인증 데이터베이스에서 사용자를 삭제하면 연결된 그룹도 삭제됩니다. 구성원이 없는 그룹은 keyfile/digestfile 인증 데이터베이스에서 사용할 수 없기 때문입니다.

주 – CLI 사용

CLI를 통해 사용자를 삭제하려면 다음 명령을 실행합니다.

```
wadm> delete-user --user=admin --password-file=admin.pwd --host=serverhost
--port=8989 --config config1 --authdb KEYFILE1 user1
```

CLI 참조 `delete-user`(1)를 참조하십시오.

▼ 그룹 삭제

1 **구성을 선택합니다.**
구성 목록에서 구성을 선택합니다. 구성 탭을 눌러 목록을 가져옵니다.

2 **액세스 제어 > 그룹 탭을 누릅니다.**

3 **그룹을 삭제할 인증 데이터베이스를 선택합니다.**

4 **사용자 검색 입력란에 사용자 아이디를 입력하고 검색 버튼을 누릅니다.**

5 **사용자 아이디 열에서 사용자를 선택하고 삭제 버튼을 누릅니다.**

주 – 그룹을 삭제해도 그룹에 속한 사용자는 삭제되지 않습니다. 사용자를 직접 삭제하거나 그룹을 다시 지정해야 합니다.

주 – CLI 사용

CLI를 통해 그룹을 삭제하려면 다음 명령을 실행합니다.

```
wadm> delete-group --user=admin --password-file=admin.pwd --host=serverhost
--port=8989 --config config1 --authdb LDAP1 group1
```

CLI 참조 `delete-group(1)`을 참조하십시오.

정적 및 동적 그룹

그룹은 LDAP 데이터베이스에 있는 일련의 객체를 기술하는 객체입니다. Web Server 7.0 그룹은 공통 속성을 공유하는 사용자로 구성됩니다. 예를 들어 일련의 객체는 회사의 마케팅 부서에서 일하는 다수의 직원일 수 있습니다. 이러한 직원은 Marketing이라는 이름의 그룹에 속할 수 있습니다.

LDAP 서비스의 경우 정적 및 동적의 두 가지 방법으로 그룹의 구성원을 정의할 수 있습니다. 정적 그룹은 구성원 객체를 명시적으로 열거합니다. 정적 그룹은 CN이며 `uniqueMembers` 및/또는 `memberURLs` 및/또는 `memberCertDescriptions`를 포함합니다. 정적 그룹의 경우 구성원은 `CN=<Groupname>` 속성을 제외한 공통 속성을 공유하지 않습니다.

동적 그룹을 사용하면 LDAP URL을 사용하여 그룹 구성원에만 적용되는 일련의 규칙을 정의할 수 있습니다. 동적 그룹에서 구성원은 공통 속성 또는 `memberURL` 필터에 정의된 일련의 속성을 공유합니다. 예를 들어 Sales의 모든 직원이 포함된 그룹이 필요하며 이 직원들이 이미 LDAP 데이터베이스의

"`ou=Sales,o=Airius.com`"에 있는 경우 다음 `memberurl`을 사용하여 동적 그룹을 정의합니다.

```
ldap:///ou=Sales,o=sun??sub?(uid=*)
```

결과적으로 이 그룹에는 "`ou=Sales,o=sun`" 지점 아래의 트리에 `uid` 속성이 있는 모든 객체 즉, 모든 Sales 구성원이 포함됩니다.

정적 및 동적 그룹의 경우 `memberCertDescription`을 사용하면 구성원이 인증서에 있는 공통 속성을 공유할 수 있습니다. 참고로 이는 ACL이 SSL 메소드를 사용하는 경우에만 작동합니다.

새 그룹을 만들었으면 그룹에 사용자 또는 구성원을 추가할 수 있습니다.

정적 그룹

LDAP 서비스의 경우 Administration Server를 사용하면 사용자 수에 상관없이 DN에서 동일한 그룹 속성을 지정하여 정적 그룹을 만들 수 있습니다. 정적 그룹은 사용자를 추가하거나 삭제하지 않는 한 변경되지 않습니다.

정적 그룹 생성을 위한 지침

Administration Server 형식을 사용하여 새 정적 그룹을 만드는 경우 다음 지침을 고려하십시오.

- 정적 그룹에는 다른 정적 또는 동적 그룹이 포함될 수 있습니다.
- 또한 선택적으로 새 그룹에 대한 설명을 추가할 수 있습니다.
- 디렉토리에 조직 단위가 정의된 경우 새 그룹을 추가할 위치 목록을 사용하여 새 그룹을 배치할 위치를 지정할 수 있습니다. 기본 위치는 디렉토리의 루트 지점 또는 최상위 항목입니다.

동적 그룹

동적 그룹에는 `groupOfURLs`의 `objectclass`가 있으며 0개 이상의 `memberURL` 속성이 있습니다. 각 속성은 일련의 객체를 기술하는 LDAP URL입니다.

LDAP 서비스의 경우, Web Server에서 임의의 속성을 기반으로 사용자를 자동으로 그룹화하려는 경우 또는 일치하는 DN이 포함된 특정 그룹에 ACL을 적용하려는 경우 동적 그룸을 만들 수 있습니다. 예를 들어 `department=marketing` 속성이 있는 DN이 자동으로 포함되도록 그룹을 만들 수 있습니다. department=marketing 검색 필터를 적용하면 `department=marketing` 속성이 있는 모든 DN을 포함하는 그룹이 반환됩니다. 그러면 이 필터에 기반한 검색 결과에서 동적 그룹을 정의할 수 있습니다. 따라서 결과의 동적 그룹에 대한 ACL을 정의할 수 있습니다.

Web Server의 동적 그룹 구현 방식

Web Server는 LDAP 서버 스키마의 동적 그룹을 `objectclass = groupOfURLs`로 구현합니다. `groupOfURLS` 클래스에는 여러 `memberURL` 속성이 있을 수 있으며 각 속성은 디렉토리에 있는 객체 세트를 열거하는 LDAP URL로 구성됩니다. 그룹의 구성원은 이러한 세트의 조합이 됩니다. 예를 들어 다음 그룹은 하나의 구성원 URL만 포함합니다.

```
ldap:///o=mcom.com??sub?(department=marketing)
```

이 예는 부서가 "marketing"인 "`o=mcom.com`" 아래의 모든 객체로 구성된 세트를 나타냅니다. LDAP URL는 검색 기본 DN, 범위 및 필터를 포함할 수 있지만 호스트 이름과 포트는 포함할 수 없습니다. 따라서 동일한 LDAP 서버에 있는 객체만 참조할 수 있습니다. 범위는 모두 지원됩니다.

DN은 자동으로 포함되므로 직접 개인을 그룹에 추가할 필요가 없습니다. ACL 확인을 위해 그룹 조회가 필요할 때마다 Sun ONE Web Server에서 LDAP 서버 검색을 수행하므로 그룹은 동적으로 변경됩니다. ACL 파일에서 사용된 사용자 및 그룹 이름은 LDAP 데이터베이스에 있는 객체의 cn 속성에 대응됩니다.

주 – Web Server는 ACL용 그룹 이름으로 `cn(commonName)` 속성을 사용합니다.

ACL에서 LDAP 데이터베이스로의 매핑은 `dbswitch.conf` 구성 파일(ACL 데이터베이스 이름을 실제 LDAP 데이터베이스 URL과 연결) 및 ACL 파일(ACL용으로 사용할 데이터베이스 정의) 모두에 정의됩니다. 예를 들어 "staff"라는 그룹의 구성원에게 기본 액세스 권한을 부여하려는 경우 ACL 코드는 객체 클래스가 `groupOf<anything>`이고 CN이 "`staff`"로 설정된 객체를 조회합니다.객체는 구성원 DN을 명시적으로 열거하거나(정적 그룹의 `groupOfUniqueNames`와 동일) LDAP URL을 지정(예: `groupOfURLs`)하여 그룹의 구성원을 정의합니다.

정적 및 동적 그룹 가능

그룹 객체에는 `objectclass = groupOfUniqueMembers` 및 `objectclass = groupOfURL`이 모두 있을 수 있으며, 따라서 "`uniqueMember`" 및 "`memberURL`" 속성이 모두 유효합니다. 그룹의 구성원은 정적 및 동적 구성원의 조합입니다.

서버 성능에 미치는 동적 그룹의 영향

동적 그룹을 사용하는 경우 서버 성능에 영향을 미칠 수 있습니다. 그룹 구성원을 테스트해서 DN이 정적 그룹의 구성원이 아닌 경우 Web Server는 데이터베이스의 baseDN에 있는 모든 동적 그룹을 확인합니다. Web Server는 이러한 작업을 위해 해당 baseDN과 사용자의 DN에 대한 범위를 확인하여 각 `memberURL`이 일치하는지 확인한 다음 사용자 DN을 baseDN으로 사용하고 `memberURL`의 필터를 사용하여 기본 검색을 수행합니다. 이 절차로 인하여 많은 수의 개별 검색이 누적될 수 있습니다.

동적 그룹 생성을 위한 지침

Administration Server를 사용하여 새 동적 그룹을 만드는 경우에는 다음 지침을 고려하십시오.

- 동적 그룹에는 다른 그룹이 포함될 수 없습니다.
- 다음 형식으로 그룹의 LDAP URL을 입력합니다(호스트 및 포트 매개 변수는 무시되므로 생략).

  ```
  ldap:///<basedn>?<attributes>?<scope>?<(filter)>
  ```

  필요한 매개 변수는 다음 표에 설명한 것과 같습니다.

표 8–1 동적 그룹: 필요한 매개 변수

매개 변수 이름	설명
<base_dn>	검색 기반의 고유 이름(DN) 또는 LDAP 디렉토리에서 모든 검색이 수행되는 지점. 이 매개 변수가 "o=mcom.com" 등의 디렉토리 접미사 또는 루트로 설정되는 경우도 종종 있습니다.
<attributes>	검색이 반환할 수 있는 속성 목록. 속성을 두 개 이상 지정하려면 속성 사이를 쉼표로 분리(예: "cn,mail,telephoneNumber")합니다. 속성이 지정되어 있지 않으면 모든 속성이 반환됩니다. 참고로 동적 그룹 구성원 확인의 경우 이 매개 변수는 무시됩니다.
<scope>	검색의 범위로 다음 중 한 가지 값을 가집니다. ■ base는 해당 URL에 지정된 고유 이름(<base_dn>)에 대한 정보를 검색합니다. ■ one은 해당 URL에 지정된 고유 이름(<base_dn>)보다 한 수준 아래의 항목에 대한 정보를 검색합니다. 기본 항목은 이 범위에 포함되지 않습니다. ■ sub는 해당 URL에 지정된 고유 이름(<base_dn>)보다 아래에 있는 모든 수준의 항목에 대한 정보를 검색합니다. 기본 항목은 이 범위에 포함되지 않습니다. 이 매개 변수는 필수입니다.
<(filter)>	검색의 지정된 범위 내에 있는 항목에 적용되는 검색 필터. Administration Server 형식을 사용하는 경우에는 이 속성을 지정해야 합니다. 괄호는 반드시 필수입니다. 이 매개 변수는 필수입니다.

<attributes>, <scope> 및 <(filter)> 매개 변수는 URL에서의 위치에 따라 구분됩니다. 속성을 지정하지 않으려는 경우에도 해당 필드에 물음표를 넣어 구분해야 합니다.

- 또한 선택적으로 새 그룹에 대한 설명을 추가할 수 있습니다.
- 디렉토리에 조직 단위가 정의된 경우 새 그룹을 추가할 위치 목록을 사용하여 새 그룹을 배치할 위치를 지정할 수 있습니다. 기본 위치는 디렉토리의 루트 지점 또는 최상위 항목입니다.

◆ ◆ ◆ 9 장

9 서버 내용 관리

이 장에서는 가상 서버의 내용을 구성하고 관리하는 방법에 대해 설명합니다.

문서 디렉토리 구성

문서 루트라고도 하는 기본 문서 디렉토리는 원격 클라이언트에서 사용할 수 있도록 할 모든 파일을 저장하는 중앙 디렉토리입니다.

기본 문서 디렉토리 외에 문서 디렉토리를 만들 수 있습니다. 이렇게 하면 기본 문서 루트에 대한 액세스를 부여하지 않고도 다른 사용자가 문서 그룹을 관리하도록 할 수 있습니다.

▼ 문서 디렉토리를 만드는 방법

1 **구성을 선택합니다.**
구성 목록에서 구성을 선택합니다. 구성 탭을 눌러 사용 가능한 구성을 가져옵니다.

2 **가상 서버를 선택합니다.**
새 문서 디렉토리를 추가해야 하는 가상 서버를 선택합니다. 가상 서버 탭을 눌러 선택한 구성에 대해 구성된 가상 서버 목록을 가져옵니다.

3 **내용 처리 > 문서 디렉토리 탭을 누릅니다.**

4 **새로 만들기 버튼을 누릅니다. 다음 매개 변수를 구성합니다.**

- **URL 접두어** — 디렉토리에 매핑해야 하는 URI 접두어입니다.
- **디렉토리 경로** — 절대 서버 경로와 문서를 저장할 유효 디렉토리입니다.

주 – CLI 사용

CLI를 통해 문서 디렉토리를 만들려면 다음 명령을 실행합니다.

```
wadm> create-document-dir --user=admin --password-file=admin.pwd
--host=serverhost --port=8989 --config=config1 --vs=config1_vs_1
--uri-prefix=/config1_uri --directory=../docs1
```

CLI 참조 `create-document-dir`(1)을 참조하십시오.

기본 MIME 유형 변경

문서가 클라이언트로 전송될 때 서버는 문서의 유형을 식별하는 부분을 포함시켜 클라이언트가 문서를 제대로 표시할 수 있도록 합니다. 하지만 문서의 확장자가 서버에 정의되어 있지 않기 때문에 서버에서 문서의 적절한 유형을 확인하기 어려운 경우도 있습니다. 이런 경우에는 기본값이 전송됩니다.

기본값은 보통 text/plain이지만 서버에 가장 일반적으로 저장되는 유형의 파일을 설정해야 합니다. 일반적인 MIME 유형은 다음과 같습니다.

■ `text/plain`	■ `text/html`
■ `text/richtext`	■ `image/tiff`
■ `image/jpeg`	■ `image/gif`

■ application/x-tar	■ application/postscript
■ application/x-gzip	■ audio/basic

▼ 기본 MIME 유형을 변경하는 방법

1. **구성을 선택합니다.**
   구성 목록에서 구성을 선택합니다. 구성 탭을 눌러 사용 가능한 구성을 가져옵니다.
2. **가상 서버를 선택합니다.**
   가상 서버 탭을 눌러 선택한 구성에 대해 구성된 가상 서버 목록을 가져옵니다.
3. **내용 처리 > 일반 탭을 누릅니다.**
4. **기타 섹션 아래에서 기본 MIME 유형 값을 변경합니다.**

주 – CLI 사용

CLI를 통해 MIME 유형을 만들려면 다음 명령을 실행합니다.

```
wadm> create-mime-type --user=admin --password-file=admin.pwd --host=serverhost
--port=8989 --config=config1 --extensions=sxc application/sxc
```

CLI 참조 `create-mime-type`(1)을 참조하십시오.

각 가상 서버용으로 별도의 MIME 유형을 만들 필요는 없습니다. 대신, 필요한 만큼의 MIME 유형 파일을 만들고 이를 가상 서버에 연결할 수 있습니다. 기본적으로 서버에는 한 개의 MIME 유형 파일(mime.types)이 있고 이 파일은 삭제할 수 없습니다.

사용자 공용 정보 디렉토리 사용자 정의(UNIX/Linux)

사용자가 자신의 웹 페이지를 유지 관리하려는 경우도 있습니다. 서버에 있는 모든 사용자가 관리자의 개입 없이 홈 페이지 및 기타 문서를 만들 수 있게 해 주는 공용 정보 디렉토리를 구성할 수 있습니다.

이 시스템에서 클라이언트는 서버에서 공용 정보 디렉토리로 인식하는 특정 URL을 사용하여 서버에 액세스할 수 있습니다. 예를 들어 접두어 ~ 및 디렉토리 `public_html`을 선택하는 경우를 가정합니다. `http://www.sun.com/~jdoe/aboutjane.html`에 대한 요청이 들어오면 서버는 `~jdoe`가 사용자의 공용 정보 디렉토리를 가리킨다는 사실을 인식합니다. 서버는 시스템의 사용자 데이터베이스에서 `jdoe`를 조회하고 Jane의 홈 디렉토리를 찾습니다. 그런 다음 `~/jdoe/public_html/aboutjane.html`을 찾습니다.

공용 디렉토리를 사용하도록 서버를 구성하려면 다음 단계를 수행합니다.

▼ 문서 디렉토리 구성

1 **가상 서버 페이지에서 내용 처리 탭을 누릅니다.**

2 **문서 디렉토리를 누릅니다.**

3 **사용자 문서 디렉토리에서 사용자 URL 접두어를 선택합니다.**
일반적으로 사용되는 접두어는 ~입니다. 이는 ~ 문자가 사용자의 홈 디렉토리 액세스에 사용되는 표준 UNIX/Linux 접두어이기 때문입니다.

4 **사용자의 홈 디렉토리에서 서버가 HTML 파일을 찾는 하위 디렉토리를 선택합니다.**
일반적인 디렉토리는 `public_html`입니다.

5 **비밀번호 파일을 지정합니다.**
서버는 시스템에 있는 사용자를 나열하는 파일을 어디에서 찾을지 알아야 합니다. 서버는 이 파일을 사용하여 유효한 사용자 이름과 해당 사용자의 홈 디렉토리를 확인합니다. 이러한 용도로 시스템 비밀번호 파일을 사용하면 서버에서 표준 라이브러리 호출을 사용하여 사용자를 조회합니다. 또는 사용자를 조회할 다른 사용자 파일을 만들 수 있습니다. 이 사용자 파일을 절대 경로로 지정할 수 있습니다.

파일에 있는 각 줄의 구조는 다음과 같아야 합니다(필요 없는 `/etc/passwd`파일 요소는 *로 표시).

```
username:*:*:groupid:*:homedir:*
```

6 **시작할 때 비밀번호 데이터베이스를 로드할지 여부를 선택합니다.**

7 **저장을 누릅니다.**
자세한 내용은 User Document Directories 페이지에 대한 온라인 도움말을 참조하십시오.

사용자에게 별도의 디렉토리를 부여하는 또 다른 방법은 모든 사용자가 수정할 수 있는 중앙 디렉토리에 매핑되는 URL을 작성하는 것입니다.

내용 게시 제한

시스템 관리자가 사용자 문서 디렉토리를 통해 내용을 게시할 수 있는 사용자 계정을 제한하려는 경우도 있을 수 있습 있습니다. 사용자의 게시를 제한하려면 `/etc/passwd file`의 사용자 홈 디렉토리 경로 끝에 슬래시를 추가합니다.

```
jdoe::1234:1234:John Doe:/home/jdoe:/bin/sh
```

becomes:

```
jdoe::1234:1234:John Doe:/home/jdoe/:/bin/sh
```

이렇게 수정하면 Sun Java System Web Server는 이 사용자의 디렉토리에서 페이지를 서비스하지 않습니다. 해당 URI를 요청하는 브라우저에 "404 File Not Found" 오류가 수신되고 웹 서버 액세스 로그에 404 오류가 기록됩니다. 오류 로그에는 오류가 기록되지 않습니다.

나중에 이 사용자가 내용을 게시할 수 있게 허용하려면 `/etc/passwd` 항목에서 끝에 오는 슬래시를 제거한 다음 웹 서버를 다시 시작합니다.

시작시 전체 비밀번호 파일 로드

시작할 때 전체 비밀번호 파일을 로드하는 옵션도 있습니다. 이 옵션을 선택한 경우 서버는 시작할 때 메모리에 비밀번호 파일을 로드하여 사용자 조회 속도를 더 빠르게 합니다. 하지만 비밀번호 파일의 용량이 매우 큰 경우 이 옵션이 훨씬 더 많은 메모리를 사용할 수 있습니다.

URL 리디렉션 설정

URL 리디렉션을 사용하여 한 HTTP URL에 대한 문서 요청을 다른 HTTP URL로 리디렉션할 수 있습니다. URL 전달 또는 리디렉션은 서버가 사용자에게, 예를 들어 사용자가 파일을 다른 디렉토리 또는 서버로 옮겼으므로 URL이 변경되었음을 알리는 방법입니다. 또한 리디렉션을 사용하여 한 서버의 문서 요청을 아무런 문제 없이 다른 서버의 문서로 보낼 수도 있습니다.

예를 들어 `http://www.sun.com/info/movies`를 접두어 `film.sun.com`으로 전달하면 URL `http://www.sun.com/info/movies`는 `http://film.sun.com/info/movies`로 리디렉션됩니다.

한 하위 디렉토리의 모든 문서에 대한 요청을 특정 URL로 리디렉션할 경우가 있습니다. 예를 들어 어떤 디렉토리에 너무 많은 트래픽이 발생하거나 해당 문서가 어떤 이유로 인해 더 이상 서비스되지 않아 이 디렉토리를 제거해야 하는 경우, 문서를 더 이상 사용할 수 없음을 표시하는 페이지로 해당 요청을 보낼 수 있습니다. 예를 들어 `/info/movies`의 접두어는 `http://www.sun.com/explain.html`로 리디렉션될 수 있습니다.

URL 리디렉션을 가상 서버 수준에서 설정할 수 있습니다.

URL 리디렉션을 구성하려면 다음 단계를 수행하십시오.

1. **구성 탭**을 누르고 구성 목록에서 구성을 선택합니다.
2. **가상 서버 하위 탭**을 누르고 가상 서버 목록에서 가상 서버를 선택합니다.
3. **내용 처리 하위 탭**과 **URL 리디렉션 하위 탭**을 누릅니다.
4. **새로 만들기 버튼**을 눌러 새 URL 리디렉션 규칙을 추가합니다.
5. 설명된 필드에 필요한 값을 지정합니다. **확인 버튼**을 누릅니다. 필요한 경우 구성의 **배포** 버튼을 눌러야 할 수도 있습니다.

다음 표에서는 새 URL 리디렉션 규칙을 추가하는 동안 필요한 매개 변수에 대해 설명합니다.

표 9–1 URL 리디렉션 매개 변수

매개 변수	설명
시작 URL	요청 리디렉션을 시작해야 하는 URL입니다. 이 URL로의 모든 HTTP 요청은 대상 URL에 지정된 URL로 리디렉션됩니다.
대상 URL	요청을 리디렉션할 URL입니다. 시작 URL에 지정된 URL의 모든 HTTP 요청은 이 URL로 리디렉션됩니다.
URL 유형	고정. **사용 가능/사용 안 함**. 고정 URL은 HTML 페이지에 대한 링크와 같은 정적 URL입니다. 비고정 URL은 요청 매개 변수가 있는 동적 URL이거나 접두어만 있는 URL입니다.

주 – CLI 사용

CLI를 통해 새 URL 리디렉션 규칙을 추가하려면 다음 명령을 실행합니다.

```
wadm> create-url-redirect --user=admin --password-file=admin.pwd --host=serverhost
--port=8989 --no-ssl --config=config1 --vs=config1_vs_1 --uri-prefix=/redirect
--target-url=http://www.cnet.com
```

CLI 참조 `create-url-redirect`(1)를 참조하십시오.

정규 표현식을 사용하여 URL 리디렉션

Sun Java System Web Server 7.0는 정규 표현식(패턴이라고도 함)을 지원하고 구성 파일의 시간 매개 변수 보간법을 요청하도록 성능이 향상되었습니다. 또한 와일드카드 패턴 일치 지원이 server.xml로 확장되었습니다. URL 리디렉션은 SAF로 구현됩니다. 리디렉션 SAF를 사용하면 특정 접두어와 일치하는 URI를 리디렉션할 수 있습니다. `from` 매개 변수를 사용하여 접두어를 지정하고 `url` 또는 `url-prefix` 매개 변수를 사용하여 리디렉션할 URL을 지정할 수 있습니다. Sun Java System Web Server 7.0에서 `from` 매개 변수는 선택 사항입니다. `from`을 생략하면 모든 URI가 리디렉션됩니다.

obj.conf 파일의 새 `<If>`, `<Elseif>` 및 `<Else>` 태그에서는 SAF 매개 변수가 지원됩니다. 부록 - obj.conf - 구문 및 사용을 참조하십시오. 이 태그에는 지시문이 포함되어 있습니다. 이 태그를 사용하면 지시문이 실행되는 조건을 정의할 수 있습니다. 이 태그는 SAF 매개 변수를 동적으로 생성하는 데도 사용할 수 있습니다.

Sun Java System Web Server 7.0은 강력한 Apache HTTP 서버의 `mod_rewrite` 모듈인 URL 다시 쓰기 기능을 제공합니다. Apache의 `mod_rewrite` 기능과 달리 `<If>` 태그는 다음 기능을 제공합니다.

- URI, 경로, 헤더 필드 및 응답 본문을 조작할 수 있습니다.
- 요청 처리의 모든 단계에서 작동합니다.
- 타사 플러그인을 포함한 모든 SAF에서 작동합니다.

다음과 같은 지시문을 생각할 수 있습니다.

```
NameTrans fn="redirect"
                    from="/site1"
                url="http://site1.mycompany.com"
```

위의 지시문은 정규 표현식을 사용하여 다음과 같이 다시 쓸 수 있습니다.

```
<If $uri =~ '^/site1'>
           NameTrans fn="redirect"
           url="http://site1.mycompany.com"
</If>
```

위의 코드 부분에서는 `from` 매개 변수 대신 정규 표현식이 사용되었습니다. `/site1/*`의 모든 요청을 `http://site1.mycompany.com/*/index.html`로 리디렉션해야 하는 경우에는 다음 기법을 사용할 수 있습니다.

```
<If $uri =~ '^/site1/(.*)'>
           NameTrans fn="redirect"
           url="http://site1.mycompany.com/$1/index.html"
</If>
```

여기서 <If> 태그는 (.*)와 일치하는 모든 값을 변수 $1에 지정합니다. `url` 매개 변수에 있는 $1은 원래 요청 값으로 동적으로 교체됩니다. 즉, 위의 obj.conf 코드를 사용하면 `/site1/download`에 대한 요청이 `http://site1.mycompany.com.com/download/index.html`로 리디렉션됩니다.

`<If>`와 `redirect`를 함께 사용하면 `mod_rewrite`의 유연성을 약간은 제공할 수 있습니다. 하지만 `mod_rewrite`와 달리 `<If>`는 리디렉션 및 URL 다시 쓰기 이외의 작업에 사용할 수 있습니다. 또한 `<If>`는 타사 플러그인을 포함한 모든 SAF와 함께 사용할 수 있습니다.

위 방법에서는 `302 Moved Temporarily` 리디렉션을 구성합니다. 또한 Sun Java System Web Server 7.0에서는 `status="301"` 매개 변수를 추가하여 `301 Moved Permanently` 리디렉션이 필요한 것을 대신 나타낼 수도 있습니다.

```
NameTrans fn="redirect" from="/path" url="http://server.example.com" status="301"
```

CGI의 개요

CGI(Common Gateway Interface) 프로그램은 어떤 프로그래밍 언어로도 정의할 수 있습니다. UNIX/Linux 시스템에서는 Bourne 쉘 또는 Perl 스크립트로 작성된 CGI 프로그램을 찾을 수 있습니다.

주 – UNIX/Linux에서는 추가 `CGIStub` 프로세스가 실행되고 서버는 이것을 CGI 실행을 돕는 데 사용합니다. 이러한 프로세스는 CGI에 처음 액세스하는 동안에만 이루어집니다. 프로세스의 수는 서버의 CGI 로드에 따라 다릅니다. 이러한 CGIStub 프로세스를 종료하지 마십시오. 서버가 중지되면 프로세스가 사라집니다.

자세한 내용은 온라인 Sun Java System Web Server **Performance Tuning and Sizing Guide**에 있는 `MinCGIStub`, `MaxCGIStub` 및 `CGIStubIdleTimeout`을 참조하십시오.

Windows 컴퓨터에서 C++ 또는 일괄 처리 파일로 작성된 CGI 프로그램을 찾을 수 있습니다. Windows에서 Visual Basic과 같이 Windows 기반 프로그래밍 언어로 작성된 CGI 프로그램은 다른 기법을 사용하여 서버와 작동합니다. 이러한 프로그램을 Windows CGI 프로그램이라고 합니다.

주 – 명령줄 유틸리티를 실행하려면 *server_root*`/bin/https/bin`이 포함되도록 `Path` 변수를 수동으로 설정해야 합니다.

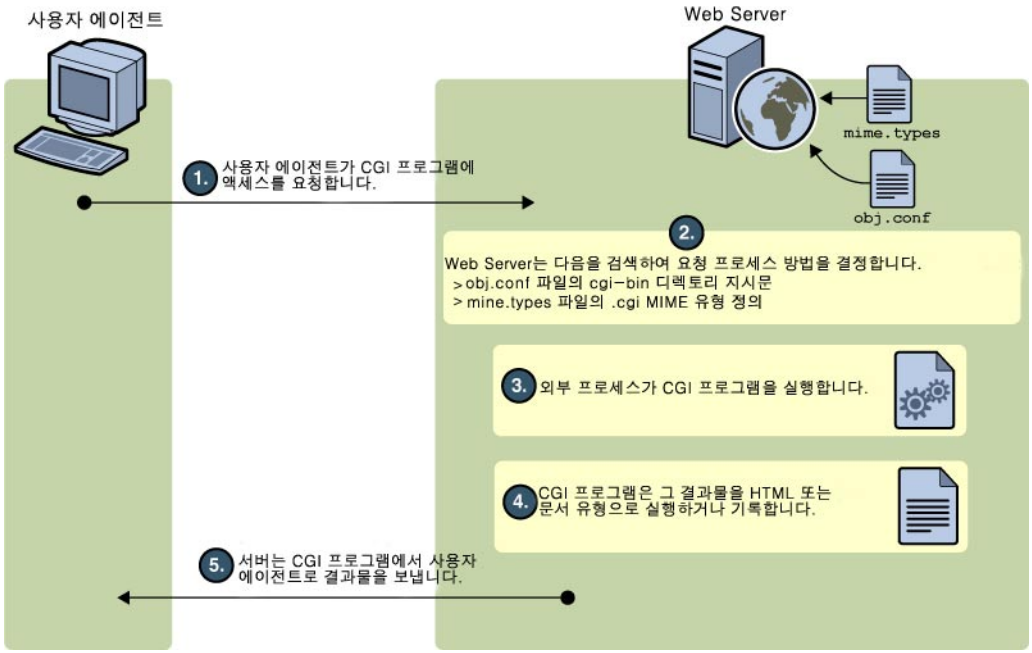
프로그래밍 언어에 관계없이 모든 CGI 프로그램은 동일한 방식으로 데이터를 받고 반환합니다. CGI 프로그램 작성에 대한 자세한 내용은 다음을 참조하십시오.

- Sun Java System Web Server *Developer's Guide*
- 다음 위치에 있는 *The Common Gateway Interface*:

  `http://hoohoo.ncsa.uiuc.edu/cgi/overview.html`
- 다음 온라인 설명서 웹 사이트에서 볼 수 있는 CGI 관련 기사:

  `http://docs.sun.com`

다음 그림은 Web Server 7.0에서 CGI 요청이 처리되는 방식을 나타냅니다.



서버 시스템에 CGI 프로그램을 저장하는 두 가지 방법이 있습니다.

- CGI 프로그램만 포함하는 디렉토리를 지정합니다. 모든 파일은 파일 확장자와 관계없이 프로그램으로 실행됩니다.
- CGI 프로그램이 모든 특정 파일 유형임을 지정합니다. 즉, 모두 파일 확장자 `.cgi`, `.exe` 또는 `.bat`를 사용합니다. 문서 루트 디렉토리나 그 아래에 있는 모든 디렉토리에서 프로그램을 찾을 수 있습니다.

원하는 경우 두 가지 옵션을 동시에 사용할 수 있습니다.

두 가지 구현에는 각기 장점이 있습니다. 특정 사용자 집합만 CGI 프로그램을 추가할 수 있게 하려면 CGI 프로그램을 지정된 디렉토리에 유지하고 해당 디렉토리에 대한 액세스를 제한합니다. HTML 파일을 추가할 수 있는 사람은 누구나 CGI 프로그램을 추가할 수 있게 하려면 파일 유형 방법을 사용합니다. 사용자는 CGI 파일을 HTML 파일과 같은 디렉토리에 유지할 수 있습니다.

디렉토리 옵션을 선택하면 서버는 해당 디렉토리의 모든 파일을 CGI 프로그램으로 해석합니다. 동일한 토큰으로 파일 유형 옵션을 선택하면 서버는 파일 확장자가 `.cgi`, `.exe` 또는 `.bat`인 모든 파일을 CGI 프로그램으로 처리합니다. 파일이 이러한 확장자 중 하나를 가지지만 CGI 프로그램이 아닌 경우 사용자가 액세스를 시도하면 오류가 발생합니다.

주 – 기본적으로 CGI 프로그램의 파일 확장자는 `.cgi`, `.exe` 및 `.bat`입니다. 그러나 MIME 유형 파일을 수정하여 CGI 프로그램을 나타내는 확장자를 변경할 수도 있습니다. 서버 기본 설정 탭을 선택하고 MIME 유형 링크를 눌러 이 작업을 수행할 수 있습니다.

서버에 CGI 하위 시스템 구성

Sun Java System Web Server를 사용하면 관리 콘솔 GUI를 사용하여 CGI 문서 디렉토리를 추가할 수 있습니다.

새 CGI 문서 디렉토리를 추가하려면 다음 작업을 수행합니다.

1. **구성 탭**을 누르고 구성 목록에서 구성을 선택합니다.
2. **가상 서버 하위 탭**을 누르고 가상 서버 목록에서 가상 서버를 선택합니다.
3. **내용 처리 하위 탭**과 **CGI 하위 탭**을 누릅니다.
4. **새로 만들기 버튼**을 눌러 새 CGI 문서 디렉토리를 추가합니다.
5. 설명된 필드에 필요한 값을 지정합니다. **확인 버튼**을 누릅니다. 필요한 경우 구성에 대해 **배포** 버튼을 눌러야 할 수도 있습니다.

다음 표에서는 새 CGI 문서 디렉토리를 추가하는 동안 필요한 필드에 대해 설명합니다.

표 9–2 CGI 매개 변수

매개 변수	설명
접두어	이 디렉토리에 사용할 URL 접두어를 입력합니다. 입력하는 텍스트는 URL에서 CGI 프로그램의 디렉토리로 표시됩니다. 예를 들어 URL 접두어로 `cgi-bin`을 입력하는 경우 해당 CGI 프로그램에 대한 모든 URL의 구조는 다음과 같습니다. `http://`*`yourserver.domain.com`* `/cgi-bin/`*`program-name`*
CGI 디렉토리	CGI 디렉토리 텍스트 필드에서 디렉토리의 위치를 절대 경로로 입력합니다. 이 디렉토리가 반드시 문서 루트 아래에 있어야 하는 것은 아닙니다. 이것이 URL 접두어를 지정해야 하는 이유입니다. **주 –** 지정하는 URL 접두어는 실제 CGI 디렉토리와 다를 수 있습니다.
사용자	CGI 프로그램을 실행할 사용자의 이름을 지정합니다.
그룹	CGI 프로그램을 실행할 그룹의 이름을 지정합니다.

표 9–2 CGI 매개 변수 *(계속)*

매개 변수	설명
Chroot	실행이 시작되기 전 chroot할 디렉토리를 지정합니다.
Nice	서버에 대한 상대적인 CGI 프로그램의 우선 순위를 결정하는 증가분인 nice 값을 지정합니다. 보통 서버는 값이 0인 nice로 실행되며 nice의 증가분은 0(CGI 프로그램이 서버와 동일한 우선 순위로 실행)에서 19(CGI 프로그램이 서버보다 매우 낮은 우선 순위로 실행) 사이입니다. nice 증가분을 -1로 지정하여 CGI 프로그램의 우선 순위를 서버보다 높게 설정할 수 있지만 권장되지 않습니다.

기존 CGI 디렉토리를 제거하려면 CGI 디렉토리를 선택하고 삭제 버튼을 누릅니다. 기존 디렉토리의 URL 접두어 또는 CGI 디렉토리를 변경하려면 디렉토리 링크를 누릅니다.

CGI 프로그램을 지정한 디렉토리에 복사합니다. 해당 디렉토리의 모든 파일이 CGI 파일로 처리되므로 HTML 파일은 CGI 디렉토리에 넣지 않도록 합니다.

CGI를 파일 유형으로 지정하려면 다음 작업을 수행합니다.

1. **구성 탭**을 누르고 구성 목록에서 구성을 선택합니다.
2. **가상 서버 하위 탭**을 누르고 가상 서버 목록에서 가상 서버를 선택합니다.
3. **내용 처리 하위 탭**과 **CGI 하위 탭**을 누릅니다.
4. **CGI를 파일 유형으로** 라디오 버튼을 눌러 활성화합니다.

CGI 파일의 파일 확장자는 `.bat`, `.exe` 또는 `.cgi`여야 합니다. 이러한 확장자를 가진 CGI가 아닌 파일을 서버에서 CGI 파일로 처리하면 오류가 발생합니다.

주 – CLI 사용

서버에서 처리되는 CGI 프로그램을 포함하는 CGI 디렉토리를 만들 수 있습니다. CGI 프로그램은 .cgi, .exe 또는 .bat와 같은 특정 파일 유형을 가집니다. 문서 루트 디렉토리나 그 아래에 있는 디렉토리에서 프로그램을 찾을 수 있습니다.

CLI를 통해 CGI 디렉토리를 추가하려면 다음 명령을 실행합니다.

```
wadm> create-cgi-dir --user=admin --password-file=admin.pwd --host=serverhost
--port=8989 --config=config1 --vs=config1_vs_1 --uri-prefix=/config1_urlprefix
--directory=/cgi-dir
```

CLI 참조 `create-cgi-dir`(1)을 참조하십시오.

실행 파일 다운로드

.exe를 CGI 파일 유형으로 사용하는 경우 .exe 파일을 실행 파일로 다운로드할 수 없습니다.

이 문제에 대한 한 가지 해결책은 사용자가 다운로드할 수 있게 허용할 실행 파일을 압축하여 .exe 이외의 확장자를 갖게 하는 것입니다. 이러한 해결책에는 다운로드 시간이 짧아진다는 장점도 있습니다.

또 다른 가능한 해결책은 magnus-internal/cgi 유형에서 .exe를 파일 확장자로 제거하고 대신 application/octet-stream 유형(일반 다운로드 가능 파일에 대한 MIME 유형)에 추가하는 것입니다. Server Manager에서 서버 기본 설정 탭을 선택하고 MIME 유형 링크를 눌러 이 작업을 수행할 수 있습니다. 그러나 이 방법의 단점은 변경을 한 후에 .exe 파일을 CGI 프로그램으로 사용할 수 없다는 것입니다.

또 다른 해결책은 서버의 obj.conf 파일을 편집하여 디렉토리의 모든 파일이 자동으로 다운로드되는 다운로드 디렉토리를 설정하는 것입니다. 서버의 나머지 부분은 영향을 받지 않습니다. 자세한 내용은 다음을 참조하십시오.

```
http://developer.netscape.com/docs/manuals/enterprise/admnunix/programs.htm
```

Windows용 쉘 CGI 프로그램 설치

Windows용 CGI 프로그램의 개요

쉘 CGI는 Windows에 설정된 파일 연결을 사용하여 CGI 응용 프로그램을 실행할 수 있게 해 주는 서버 구성입니다.

예를 들어 서버가 hello.pl이라는 쉘 CGI 파일에 대한 요청을 받으면 서버는 Windows 파일 연결을 통해 .pl 확장자와 연결된 프로그램을 사용하여 파일을 실행합니다. .pl 확장자가 프로그램 C:\\bin\\perl.exe와 연결된 경우 서버는 hello.pl 파일을 다음과 같이 실행하려고 시도합니다.

```
c:\\bin\\perl.exe hello.pl
```

쉘 CGI를 구성하는 가장 쉬운 방법은 쉘 CGI 파일만 포함하는 디렉토리를 서버의 문서 루트 아래에 만드는 것입니다. 그러나 Sun ONE Web Server에서 MIME 유형을 편집하여 특정 파일 확장자가 쉘 CGI와 연결되도록 서버를 구성할 수도 있습니다.

주 – Windows 파일 확장자의 설정에 대한 자세한 내용은 Windows 설명서를 참조하십시오.

오류 응답 사용자 정의

가상 서버에서 오류가 발생하면 클라이언트에게 자세한 메시지를 전송하는 사용자 정의 오류 응답을 지정할 수 있습니다. 전송할 파일이나 실행할 CGI 프로그램을 지정할 수 있습니다.

예를 들어, 서버가 특정 디렉토리에서 오류를 수신한 경우 작동하는 방식을 변경할 수 있습니다. 클라이언트가 액세스 제어로 보호된 서버의 일부에 연결하려 하면 계정을 얻는 방법에 대한 정보를 담은 오류 파일을 반환할 수 있습니다.

사용자 정의 오류 응답을 사용하려면 먼저 오류에 대한 응답으로 전송할 HTML 파일이나 실행할 CGI 프로그램을 만들어야 합니다.

사용자 정의 오류 페이지를 추가하려면 다음 단계를 수행합니다.

1. **구성 탭**을 누르고 구성 목록에서 구성을 선택합니다.
2. **가상 서버 하위 탭**을 누르고 가상 서버 목록에서 가상 서버를 선택합니다.
3. **내용 처리 하위 탭**과 **오류 페이지 하위 탭**을 누릅니다.
4. **새로 만들기** 버튼을 눌러 사용자 정의 오류 페이지를 추가합니다.

   변경하려는 각 오류 코드에 대해 오류 응답을 포함하는 파일 또는 CGI의 절대 경로를 지정합니다.
5. **확인**을 눌러 오류 페이지 목록으로 돌아갑니다.

주 – CLI 사용

CLI를 통해 오류 페이지를 사용자 정의하려면 다음 명령을 실행합니다.

```
wadm> set-error-page --user=admin --password-file=admin.pwd --host=serverhost
--port=8989 --config=config1 --vs=config1_vs_1 --code=500
--error-page=/server-error-uri-new
```

CLI 참조 `set-error-page`(1)를 참조하십시오.

문자 집합 변경

문서의 문자 집합은 부분적으로 문서가 작성된 언어에 의해 결정됩니다. 자원을 선택하고 해당 자원에 문자 집합을 입력하여 문서, 문서 집합 또는 디렉토리의 클라이언트 기본 문자 집합을 대체할 수 있습니다.

대부분의 브라우저는 HTTP에서 MIME 유형 `charset` 매개 변수를 사용하여 문자 집합을 변경할 수 있습니다. 서버의 응답에 이 매개 변수가 포함되어 있는 경우 브라우저가 그에 맞게 문자 집합을 변경합니다. 예:

- `Content-Type: text/html;charset=iso-8859-1`

- Content-Type: text/html;charset=iso-2022-jp

일반적으로 사용되는 몇 가지 브라우저에서 인식할 수 있는 다음과 같은 charset 이름이 RFC 17.000에 지정되어 있습니다(x-로 시작되는 이름 제외).

■ us-ascii	■ iso-8859-1
■ iso-2022-jp	■ x-sjis
■ x-euc-jp	■ x-mac-roman

또한 us-ascii에서는 다음과 같은 별칭이 인식됩니다.

■ ansi_x3.4-1968	■ iso-ir-6
■ ansi_x3.4-1986	■ iso_646.irv:1991
■ ascii	■ iso646-us
■ us	■ ibm367.0
■ cp367.0	

iso_8859-1에서는 다음과 같은 별칭이 인식됩니다.

■ latin1	■ iso_8859-1
■ iso_8859-1:1987.0	■ iso-ir-100
■ ibm819	■ cp819

문자 집합을 변경하려면 다음 단계를 수행합니다.

▼ 문자 집합 변경

1. **가상 서버 페이지에서 내용 처리 탭을 누릅니다.**
2. **일반 탭을 누릅니다.**
3. **기타 섹션 아래에서 기본 문자 집합을 설정합니다.**
   이 필드를 비워 두면 문자 집합이 NONE으로 설정됩니다.
4. **저장을 누릅니다.**

문서 바닥글 설정

문서 바닥글을 지정하여 서버의 특정 섹션 내 모든 문서에 대해 마지막으로 수정된 시간을 포함할 수 있습니다. 이 바닥글은 CGI 스크립트의 출력이나 구문 분석된 HTML(.shtml) 파일을 제외한 모든 파일에 사용됩니다. CGI 스크립트 출력이나 구문 분석된 HTML 파일에 문서 바닥글을 표시해야 하는 경우 바닥글 텍스트를 별도의 파일에 입력하고 코드 라인을 추가하거나, 다른 서버측에서 이 파일을 페이지의 출력에 포함하도록 합니다.

문서 바닥글을 설정하려면 다음 단계를 수행합니다.

▼ 문서 바닥글을 설정하는 방법

1. **가상 서버 페이지에서 내용 처리 탭을 누릅니다.**
2. **일반 하위 탭을 누르고 문서 바닥글 섹션으로 이동합니다.**
3. **바닥글에 포함할 파일의 유형을 지정합니다.**
4. **날짜 형식을 지정합니다.**
5. **바닥글에 표시할 텍스트를 입력합니다.**
   문서 바닥글에 사용할 수 있는 최대 문자 수는 7.065자입니다. 문서가 마지막으로 수정된 날짜를 포함시키려면 문자열 `:LASTMOD:`를 입력합니다.
6. **저장을 누릅니다.**

주 – CLI 사용

CLI를 통해 문서 바닥글을 설정하려면 다음 명령을 실행합니다.

```
wadm> enable-document-footer --user=admin --password-file=admin.pwd
--host=serverhost --port=8989 --config=config1 --vs=config1_vs_1
--mime-type=text/html --date-format=%B --footer="config1 footer"
```

CLI 참조 `enable-document-footer`(1)를 참조하십시오.

심볼 링크(UNIX/Linux) 제한

서버 내의 파일 시스템 링크 사용을 제한할 수 있습니다. 파일 시스템 링크는 다른 디렉토리 또는 파일 시스템에 저장된 파일을 참조합니다. 참조를 통해 파일이 현재 디렉토리에 있는 것처럼 원격으로 액세스할 수 있습니다. 파일 시스템 링크에는 두 가지 유형이 있습니다.

- 하드 링크—하드 링크는 실제로 같은 데이터 블록 집합을 가리키는 두 개의 파일 이름으로, 원래 파일과 링크가 동일합니다. 따라서 하드 링크는 다른 파일 시스템에 적용할 수 없습니다.
- 심볼릭(소프트) 링크—심볼릭 링크는 데이터를 포함하는 원래 파일과 원래 파일을 가리키는 또 다른 파일의 두 가지 파일로 구성됩니다. 심볼릭 링크는 하드 링크보다 더 유연하므로 서로 다른 파일 시스템 전체에서 사용할 수 있으며 디렉토리로 연결될 수 있습니다.

하드 링크와 심볼릭 링크에 대한 자세한 내용은 UNIX/Linux 시스템 설명서를 참조하십시오.

파일 시스템 링크는 기본 문서 디렉토리 외부의 문서에 대해 포인터를 만들 수 있는 쉬운 방법으로, 누구나 이러한 링크를 만들 수 있습니다. 이로 인해 사람들이 중요한 파일(예: 기밀 문서 또는 시스템 비밀번호 파일)에 대해 포인터를 만드는 문제를 염려할 수 있습니다.

심볼릭 링크를 제한하려면 다음 단계를 수행합니다.

▼ 심볼릭 링크를 제한하는 방법

1. **가상 서버 페이지에서 내용 처리 탭을 누릅니다.**
2. **일반 하위 탭을 누릅니다.**
3. **기타 섹션 아래의 심볼릭 링크 섹션으로 이동합니다.**
4. **소프트 및/또는 하드 링크의 활성화 여부와 시작할 디렉토리를 선택합니다.**
5. **저장을 누릅니다.**

주 – CLI 사용

CLI를 통해 심볼릭 링크를 제한하려면 다음 명령을 실행합니다.

```
wadm> set-symlinks-prop --user=admin --password-file=admin.pwd
--host=serverhost --port=8989 --config=config1 --vs=config1_vs_1
allow-soft-links=true allow-hard-links=false directory=/abc
```

CLI 참조 `set-symlinks-prop`(1)를 참조하십시오.

서버 파싱 HTML 설정

HTML은 보통 아무런 서버 작업 없이 디스크에 있는 그대로 클라이언트에 전송됩니다. 그러나 서버는 문서를 보내기 전에 HTML 파일에서 특수한 명령 검색 즉, HTML 구문 분석을 수행할 수 있습니다. 서버에서 이러한 파일을 구문 분석하고 요청별 정보나 파일을 문서에 삽입하도록 하려면 먼저 HTML 구문 분석을 사용 가능하게 설정해야 합니다.

HTML을 구문 분석하려면 다음 단계를 수행합니다.

▼ 서버 구문 분석 HTML을 설정하는 방법

1. **가상 서버 페이지에서 내용 처리 탭을 누릅니다.**
2. **일반 하위 탭을 누릅니다.**
3. **구문 분석된 HTML/SSI 설정에서 서버 구문 분석 HTML의 활성화 여부를 선택합니다.**
   HTML 파일은 활성화하면서 exec 태그는 활성화하지 않거나 또는 HTML 파일과 exec 태그를 모두 활성화할 수 있는데, 이렇게 하면 HTML 파일이 서버의 다른 프로그램을 실행하도록 할 수 있습니다.
4. **구문 분석할 파일을 선택합니다.**
   shtml 확장자를 가진 파일만 구문 분석할지 또는 성능이 저하되더라도 모든 HTML 파일을 구문 분석할지를 선택할 수 있습니다. 또한 UNIX/Linux를 사용하는 경우, 신뢰성이 떨어지더라도 실행 권한이 설정된 UNIX/Linux 파일을 구문 분석하도록 선택할 수 있습니다.
5. **저장을 누릅니다.**
   서버 구문 분석 HTML의 사용에 대한 자세한 내용은 Sun Java System Web Server **Developer's Guide**를 참조하십시오.

주 – CLI 사용

CLI를 통해 서버 구문 분석 HTML을 설정하려면 다음 명령을 실행합니다.

```
wadm> enable-parsed-html --user=admin --password-file=admin.pwd
--host=serverhost --port=8989 --config=config1 --vs=config1_vs1
```

CLI 참조 `enable-parsed-html`(1)을 참조하십시오.

캐시 제어 지시문 설정

캐시 제어 지시문은 Sun Java System Web Server에서 프록시 서버에 의해 캐시되는 정보를 제어하는 방법입니다. 캐시 제어 지시문을 사용하면 프록시의 기본 캐시 작업을 대체하여 중요한 정보가 캐시되거나 이후 검색되지 않도록 보호할 수 있습니다. 이 지시문을 사용하려면 프록시 서버가 HTTP 1.1을 사용해야 합니다.

HTTP 1.1에 대한 자세한 내용은 다음 웹 페이지의 Hypertext Transfer Protocol--HTTP/1.1 사양(RFC 2068)을 참조하십시오.

`http://www.ietf.org/`

캐시 제어 지시문을 설정하려면 다음 단계를 수행합니다.

▼ 캐시 제어 지시문을 설정하는 방법

1. **가상 서버 페이지에서 내용 처리 탭을 누릅니다.**
2. **일반 탭을 누르고 기타 섹션 아래의 캐시 제어 지시문 필드로 이동합니다.**
3. **필드에 값을 입력합니다. 응답 지시문의 유효한 값은 다음과 같습니다.**
   - **Public.** 임의의 캐시로 응답을 캐시할 수 있습니다. 이것이 기본값입니다.
     - **Private.** 응답은 개인용(비공유) 캐시로만 캐시할 수 있습니다.
     - **No Cache.** 응답을 캐시하지 않습니다.
     - **No Store.** 캐시가 요청이나 응답을 영구적 저장소에 저장할 수 없습니다.
     - **Must Revalidate.** 원래 서버에서 캐시 항목을 다시 확인해야 합니다.
     - **Maximum Age (sec).** 클라이언트는 이 지속 시간보다 오래된 응답을 허용하지 않습니다.
4. **저장을 누릅니다.**

주 – CLI 사용

CLI를 통해 캐시 제어 지시문을 설정하려면 다음 명령을 실행합니다.

```
wadm> set-cache-control-directives --user=admin --password-file=admin.pwd
--host=serverhost --port=8989 --config=config1 --vs=config1_vs_1 public=true
private=true must-revalidate=true
```

CLI 참조 `set-cache-control-directives`(1)를 참조하십시오.

내용 압축용으로 서버 구성

Sun Java System Web Server 7.0은 HTTP 내용 압축을 지원합니다. 내용 압축을 사용하면 클라이언트로의 전송 속도가 빨라지고 하드웨어 비용을 늘리지 않으면서 용량이 더 큰 내용을 서비스할 수 있습니다. 내용 압축은 내용 다운로드 시간을 줄여주므로 전화 접속 및 트래픽이 많은 연결을 사용하는 사용자에게 가장 많은 혜택을 줍니다.

내용 압축을 사용하여 웹 서버는 압축된 데이터를 전송하고 브라우저에게 전송 중에 데이터를 압축 해제할 것을 지시하여 전송된 데이터 양을 줄이고 페이지 표시 속도를 높입니다.

서버가 미리 압축된 내용을 서비스하도록 구성

Sun Java System Web Server가 미리 압축된 버전의 파일을 생성하여 지정된 디렉토리에 저장하도록 구성할 수 있습니다. 서버를 구성하면 `Accept-encoding: gzip` 헤더만 수신되는 경우에 한해 미리 압축된 내용을 서비스하도록 구성된 디렉토리의 파일에 대한 모든 요청이 해당 디렉토리의 상응하는 압축 파일에 대한 요청으로 리디렉션됩니다(해당 파일이 존재하는 경우). 예를 들어 웹 서버가 `myfile.html`에 대한 요청을 수신하고 `myfile.html` 및 `myfile.html.gz`가 모두 존재하는 경우 적절한 `Accept-encoding` 헤더를 가진 이러한 요청에서 압축된 파일을 수신합니다.

미리 압축된 내용을 서비스하도록 서버를 구성하려면 다음 단계를 수행합니다.

▼ 미리 압축된 내용 설정을 변경하는 방법

1. **가상 서버 페이지에서 내용 관리 탭을 누릅니다.**
2. **일반 하위 탭을 누릅니다.**
3. **압축 > 미리 압축된 내용 섹션으로 이동하여 다음 옵션 중 하나를 선택합니다.**
   - **미리 압축된 내용** — 사용 가능/사용 불가. 서버가 선택된 자원에 대해 미리 압축된 내용을 서비스하도록 지시할 수 있습니다.

- **사용 기간 검사** — 압축된 버전이 압축되지 않은 버전보다 오래되었는지 여부를 확인하도록 지정합니다.

  이 옵션을 선택하면 압축된 버전이 압축되지 않은 버전보다 오래된 경우 선택되지 않습니다.

  이 옵션을 선택하지 않으면 압축된 버전이 압축되지 않은 버전보다 오래된 경우에도 항상 선택됩니다.
- **Vary 헤더 삽입** — `Vary: Accept-encoding` 헤더의 사용 여부를 지정합니다.

  이 옵션을 선택하면 압축된 버전의 파일을 선택하는 경우 항상 `Vary: Accept-encoding` 헤더가 삽입됩니다.

  이 옵션을 선택하지 않으면 `Vary: Accept-encoding` 헤더가 삽입되지 않습니다.

4 저장을 누릅니다.

요청 시에 내용을 압축하도록 서버 구성

Sun Java System Web Server 7.0이 전송 중에 전송 데이터를 압축하도록 구성할 수도 있습니다. 동적으로 생성되는 HTML 페이지는 사용자가 요청하기 전에는 존재하지 않습니다. 이것은 전자 상거래 기반의 웹 응용 프로그램과 데이터베이스 기반 사이트에 특히 유용합니다.

요청 시에 내용을 압축하도록 서버를 구성하려면 다음 단계를 수행합니다.

▼ 요청 시 내용을 압축하는 방법

1 가상 서버 페이지에서 내용 처리 탭을 누릅니다.

2 일반 하위 탭을 누릅니다. 압축 섹션 아래의 요청 시 내용 압축 섹션으로 이동합니다.

3 다음 옵션 중에서 선택합니다.

- **요청 시 압축** — 선택한 자원의 요청 시 압축을 활성화/비활성화합니다.
- **Vary 헤더 삽입** — `Vary: Accept-encoding` 헤더의 삽입 여부를 지정합니다.

  이 옵션을 선택하면 압축된 버전의 파일을 선택하는 경우 항상 `Vary: Accept-encoding` 헤더가 삽입됩니다.

  이 옵션을 선택하지 않으면 `Vary: Accept-encoding` 헤더가 삽입되지 않습니다.
- **단편 크기** — 압축 라이브러리(zlib)에서 사용할 메모리 단편 크기를 바이트 단위로 지정하여 한 번에 압축할 양을 제어합니다. 기본값은 `8096`입니다.
- **압축 수준** — 압축 수준을 지정합니다. 1 - 9 사이의 값을 선택합니다. 값 1은 속도가 가장 빠르며 값 9는 압축율이 최고입니다. 기본값은 6으로, 속도와 압축율이 조화된 값입니다.

4 **저장을 누릅니다.**

주 – CLI 사용

CLI를 통해 요청 시 압축을 활성화하려면 다음 명령을 실행합니다.

```
wadm> enable-on-demand-compression --user=admin
--password-file=admin.pwd --host=serverhost --port=8989 --config=config1
--vs=config1_vs_1 --insertvaryheader=true
--fragment-size=100 --compression-level=5
```

CLI 참조 `enable-on-demand-compression`(1)을 참조하십시오.

역방향 프록시 구성

역방향 프록시는 웹 서버(원래 서버)에 클라이언트로 표시되지만 사실은 수신하는 요청을 하나 이상의 원래 서버로 전달하는 프록시입니다. 역방향 프록시가 원래 서버로 표시되기 때문에 역방향 프록시를 사용하도록 클라이언트를 구성할 필요는 없습니다. 지정된 역방향 프록시를 구성하여 비슷하게 구성된 여러 원래 서버로 요청을 전달하면 역방향 프록시가 응용 프로그램 수준 소프트웨어 로드 밸런서의 역할을 수행할 수 있습니다.

주 – 일반적인 배포에서는 브라우저와 원래 서버 사이에 하나 이상의 역방향 프록시가 배포됩니다.

▼ 프록시 URI를 추가하는 방법

1 **구성 탭을 누른 다음 구성을 선택합니다.**

2 **가상 서버 탭을 누르고 가상 서버를 선택합니다.**

3 **내용 처리 > 역방향 프록시 탭을 누릅니다.**

4 **새 프록시 URI 버튼을 누릅니다.**
다음 매개 변수 값을 지정합니다.

- **URI** — 역방향 프록시 URI
- **서버 URL** — 쉼표로 구분된 원격 서버의 URL여러 값이 지정된 경우 서버는 지정된 서버 사이에 로드를 분산합니다.

▼ 역방향 프록시 매개 변수를 수정하는 방법

1. **구성 탭을 누른 다음 구성을 선택합니다.**
2. **가상 서버 탭을 누르고 가상 서버를 선택합니다.**
3. **내용 처리 > 역방향 프록시 탭을 누릅니다.**
4. **URI를 누릅니다.**
   편집할 수 있는 매개 변수는 다음과 같습니다.
   - **URI** — 역방향 프록시 URI
   - **서버 URL** — 쉼표로 구분된 원격 서버의 URL여러 값이 지정된 경우 서버는 지정된 서버 사이에 로드를 분산합니다.
   - **고정 쿠키** — 응답에 있는 경우 이후의 요청을 원래 서버에 고정시키는 쿠키의 이름
   - **고정 URI 매개 변수** — 라우팅 정보를 검색할 URI 매개 변수의 이름요청 URI에 URI 매개 변수가 있고 그 값에 콜론(':')과 라우팅 아이디가 있으면 요청은 라우팅 아이디에 해당하는 원래 서버에 "고정"됩니다.
   - **라우팅 헤더** — 라우팅 아이디를 원래 서버에 전달할 때 사용되는 HTTP 요청 헤더의 이름
   - **라우팅 쿠키** — 서버에서 응답에 있는 고정 쿠키를 발견한 경우 서버에서 생성되는 쿠키의 이름. 라우팅 쿠키는 서버에서 이후의 요청을 동일한 원래 서버에 전달할 수 있게 해 주는 라우팅 아이디를 저장합니다.

주 – CLI 사용

1. create-reverse-proxy 명령을 실행합니다.

```
wadm> create-reverse-proxy --user=admin --password-file=admin.pwd
--host=serverhost --port=8989 --config=test --vs=test --uri-prefix=//
--server=http://rick.india.sun.com:8080
```

CLI 참조 create-reverse-proxy(1)를 참조하십시오.

2. obj.conf 파일을 수정합니다.

```
NameTrans fn="map" from="/" name="reverse-proxy-/" to="http:/"
...
<Object name="reverse-proxy-/">
Route fn="set-origin-server" server="http://rick.india.sun.com:8080"
</Object>

<Object ppath="http:*">
Service fn="proxy-retrieve" method="*"
</Object>
```

보안 사이트로 리디렉션하려면 동일한 단계를 수행하고 --server 옵션에 https 주소를 지정합니다.

P3P 설정

P3P(Platform for Privacy Preferences)를 사용하면 웹 사이트에서 개인 정보 보호 방침을 자동으로 검색하고 사용자 에이전트가 쉽게 해석할 수 있는 표준 형식으로 표현할 수 있습니다. P3P 사용자 에이전트를 사용하면 사용자가 사이트의 방침을 파악할 수 있습니다(컴퓨터가 읽을 수 있는 형식과 사람이 읽을 수 있는 형식 모두). 자세한 내용은 http://www.w3.org/P3P/를 참조하십시오.

▼ 가상 서버의 P3P 설정 구성

1 **구성을 선택합니다.**
구성 목록에서 구성을 선택합니다. 구성 탭을 눌러 사용 가능한 구성 목록을 가져옵니다.

2 **가상 서버를 선택합니다.**
가상 서버 목록에서 가상 서버를 선택합니다. 가상 서버 탭을 눌러 선택한 구성에 사용할 수 있는 가상 서버를 가져옵니다.

3 **일반 탭을 누릅니다. P3P 섹션에서 다음 설정을 구성합니다.**

- **사용 가능** — 선택된 가상 서버에 대해 P3P를 활성화합니다.
- **정책 URL**— 관련 P3P 정책 파일의 위치를 입력합니다.
- **압축 정책** — 압축 정책은 사용자 에이전트(브라우저 또는 기타 P3P 응용 프로그램)에 힌트를 제공하여 사용자 에이전트가 정책 적용에 대해 신속하고 동기화된 결정을 내릴 수 있게 해 줍니다. 압축 정책은 P3P 사양에서 사용자 에이전트 또는 서버에 대해 선택적으로 사용할 수 있는 성능 최적화 옵션입니다.

주 – CLI 사용

가상 서버에서 P3P를 사용하려면 다음 명령을 실행합니다.

```
wadm> enable-p3p --user=admin --password-file=admin.pwd --host=serverhost
--port=8989 --config=config1 --vs=config1_vs_1 --policy-url=http://xyz.com/policyurl
```

CLI 참조 `enable-p3p`(1)를 참조하십시오.

◆ ◆ ◆ 10장

10 WebDAV를 사용하여 웹 게시

Sun Java System Web Server 7.0에서는 웹 기반 공동 작업의 표준인 WebDAV(Web-based Distributed Authoring and Versioning)를 지원합니다. WebDAV는 클라이언트가 원격 웹 컨텐트 저작 작업을 수행하도록 하는 HTTP/1.1 프로토콜의 확장입니다.

완전한 WebDAV 트랜잭션에는 WebDAV 자원에 대한 요청을 서비스할 수 있는 Sun Java System Web Server 7.0과 같은 WebDAV 사용 가능한 서버뿐만 아니라 WebDAV 사용 가능한 웹 게시 요청을 지원하는 Adobe® GoLive® 또는 Macromedia® DreamWeaver®와 같은 WebDAV 사용 가능 클라이언트가 포함됩니다.

서버측에서는 Sun Java System Web Server 7.0에서 WebDAV 요청을 서비스할 수 있도록 활성화하고 구성해야 합니다.

몇 가지 이유로 인해 WebDAV를 구성할 수 있습니다. 예를 들어, 서버 성능을 조정하고 보안 위험을 제거하고 또는 충돌 없는 원격 저작을 제공할 수 있습니다.

각각의 구성 요구 사항에 맞추기 위해 WebDAV 자원에서 서버가 잠금을 유지하는 최소 시간, 모음의 PROPFIND 요청 수준, 요청 본문에 허용된 XML 내용의 최대 크기 등을 변경할 수 있습니다.

기본 WebDAV 속성은 가상 서버 아래의 모든 모음에 대해 가상 서버 수준에서 구성될 수 있습니다. 여기에서 구성된 값은 `server.xml` 파일의 `DAV` 요소에 해당합니다.

WebDAV 속성은 모음 수준에서 구성되어 모음에 대해 구성된 가상 서버 수준 속성을 대체할 수 있습니다. 모음 수준에서 구성된 속성 값은 `server.xml` 파일의 `DAVCOLLECTION` 요소에 해당합니다.

WebDAV 정보

WebDAV는 HTTP/1.1 프로토콜의 확장 프로그램으로서 HTTP 및 XML뿐만 아니라 텍스트, 그래픽, 스프레드시트 및 모든 기타 형식을 포함하는 모든 유형의 웹 자원에 대해 저작 지원을 제공하는 새로운 HTML 메소드 및 헤더를 추가합니다. WebDAV를 사용하여 수행할 수 있는 작업은 다음과 같습니다.

- **등록 정보(meta-data) 조작**. WebDAV 메소드 PROPFIND 및 PROPPATCH를 사용하여 저작자 및 작성 날짜와 같은 웹 페이지에 대한 정보를 만들고, 제거하고, 쿼리할 수 있습니다.
- **모음 및 자원 관리**. WebDAV 메소드인 GET, PUT, DELETE 및 MKCOL을 사용하여 문서 세트를 만들고 계층적 구성원 목록(파일 시스템의 디렉토리 목록과 유사)을 검색할 수 있습니다.
- **잠금**. WebDAV를 사용하여 두 사람 이상이 동시에 한 문서에서 작업하지 못하도록 할 수 있습니다. WebDAV 메소드 LOCK 및 UNLOCK을 통해 상호 배타적인 잠금이나 공유 잠금을 사용하면 '업데이트 유실'(변경 사항 덮어쓰기) 문제를 방지할 수 있습니다.
- **이름 공간 작업**. WebDAV를 통해 WebDAV 메소드 COPY 및 MOVE를 사용하여 웹 자원을 복사 및 이동하도록 서버에 지시할 수 있습니다

Sun Java System Web Server 7.0의 WebDAV 지원은 다음과 같은 기능을 제공합니다.

- RFC 2518 표준 호환 및 RFC 2518 클라이언트와의 상호 운용성
- 게시를 위한 보안 및 액세스 제어
- 파일 시스템 기반 WebDAV 모음 및 자원에서의 효율적인 게시 작업

일반 WebDAV 용어

이 절에서는 WebDAV 작업 중에 접할 수 있는 일반 용어에 대해 간략하게 설명합니다.

URI. URI(Uniform Resource Identifier)는 URL 약자를 사용하여 추가 보안 계층을 제공하는 파일 아이디입니다. URL의 첫 부분은 URL 매핑으로 대체되기 때문에 사용자는 파일의 실제 경로 전체를 알 수 없게 됩니다.

소스 URI. 소스 URI라는 용어는 액세스할 수 있는 자원의 소스를 가리킵니다. 소스 URI의 개념을 이해하기 위해 다음 예를 살펴볼 수 있습니다.

JSP 페이지 `foo.jsp`는 URI `/docs/date.jsp`에 위치합니다. 이 페이지에는 HTML 마크업과 Java 코드가 들어 있는데, 이것을 실행하면 클라이언트 브라우저에 오늘 날짜가

인쇄됩니다. 서버가 클라이언트로부터 `foo.jsp`에 대한 GET 요청을 받으면 페이지를 제공하기 전에 Java 코드를 실행합니다. 클라이언트가 받는 것은 서버에 상주하는 `foo.jsp`가 아니라 동적으로 생성되어 현재 날짜를 표시하는 페이지입니다.

소스 URI, 즉 `/publish/docs`를 만들고 `foo.jsp`가 포함된 `/docs` 디렉토리로 매핑하면 `/publish/docs/foo.jsp`에 대한 요청은 `/docs/foo.jsp` JSP 페이지의 소스 코드에 대한 요청이 됩니다. 이 경우 서버는 Java 코드를 실행하지 않고 페이지를 제공합니다. 클라이언트는 디스크에 저장된 것과 정확하게 일치하는, 처리되지 않은 페이지를 받습니다.

따라서 소스 URI에 대한 요청은 자원의 소스에 대한 요청입니다.

모음. WebDAV 모음은 WebDAV 작업에 대해 사용 설정된 자원 또는 자원 집합입니다. 모음에는 구성원 URI라고 하는 일련의 URI가 포함되어 있어 WebDAV를 사용하는 구성원 자원을 나타냅니다.

구성원 URI. 모음에 들어 있는 일련의 URI 구성원에 해당되는 URI입니다.

내부 구성원 URI. 모음의 URI에 직접 관련된 구성원 URI입니다. 예를 들어 URL이 `http://info.sun.com/resources/info`인 자원에서 WebDAV가 사용 가능하고 URL이 `http://info.sun.com/resources/`인 자원 역시 WebDAV가 사용 가능한 경우 URL이 `http://info.sun.com/resources/`인 자원은 모음이고 `http://info.sun.com/resources/info`를 내부 구성원으로 포함합니다.

등록 정보. 자원에 대한 설명적 정보를 포함하는 이름/값 쌍입니다. 등록 정보는 자원의 효율적인 디스커버리와 관리를 위해 사용됩니다. 예를 들어 'creationdate' 등록 정보를 사용하면 자원이 작성된 날짜별로 모든 자원의 색인을 만들 수 있고, 'author' 등록 정보를 사용하면 작성자 이름별로 색인을 만들 수 있습니다.

라이브 등록 정보.서버가 집행하는 등록 정보입니다. 예를 들어 라이브 `getcontentlength` 등록 정보는 GET 요청이 반환하는 엔티티의 길이를 값으로 가지며, 이 값은 서버에서 자동으로 계산됩니다. 라이브 등록 정보는 다음을 포함합니다.

- 등록 정보 값은 읽기 전용이며 서버에서 유지 관리됩니다.
- 등록 정보 값은 클라이언트에서 유지 관리되지만 제출된 값의 구문 확인은 서버가 수행합니다.

데드 등록 정보.서버가 집행하지 않는 등록 정보입니다. 서버는 데드 등록 정보의 값을 기록만 하며 클라이언트에서 일관성을 유지 관리해야 합니다.

Sun Java System Web Server는 다음과 같은 라이브 등록 정보를 지원합니다.

- `creationdate`
- `displayname`
- `getcontentlanguage`
- `getcontentlength`
- `getcontenttype`

- gettag
- getlastmodified
- lockdiscovery
- resourcetype
- supportedlock
- executable

주 – Sun Java System Web Server는 클라이언트가 자원과 연결된 파일 권한을 변경할 수 있게 하는 라이브 등록 정보 executable을 지원합니다.

executable 라이브 등록 정보에 대한 PROPPATCH 요청의 예:

```
PROPPATCH /test/index.html HTTP/1.1

Host: sun

Content-type: text/xml

Content-length: XXXX

<?xml version="1.0"?>

<A:propertyupdate xmlns:A="DAV:" xmlns:B="http://apache.org/dav/props/">

<A:set>

<A:prop>

<B:executable>T</B:executable>

</A:prop>

</A:set>

</A:propertyupdate>
```

잠금. 리소스를 잠그는 기능은 한 사용자가 자원을 편집하는 동안 다른 사용자가 자원을 수정할 수 없게 하는 기법을 제공합니다. 잠금을 사용하면 덮어쓰기 충돌을 방지하고 "업데이트 유실" 문제를 해결할 수 있습니다.

Sun Java System Web Server는 공유 잠금과 전용 잠금의 두 가지 잠금 유형을 지원합니다.

새 HTTP 헤더. WebDAV가 HTTP/1.1 프로토콜을 확장하여 작업합니다. 여기서는 클라이언트에서 WebDAV 자원에 대한 요청을 전달할 수 있는 새 HTTP 헤더를 정의합니다. 다음과 같은 헤더가 있습니다.

- Destination:

- `Lock-Token:`
- `Timeout:`
- `DAV:`
- `If:`
- `Depth:`
- `Overwrite:`

새 HTTP 메소드.WebDAV는 WebDAV 사용 가능 서버에 요청 처리 방법을 지시하는 몇 가지 새 HTTP 메소드를 도입합니다. 이러한 메소드는 `GET`, `PUT` 및 `DELETE`와 같은 기존 HTTP 메소드에 추가로 사용되며 WebDAV 트랜잭션을 수행합니다. 새 HTTP 메소드에 대한 내용은 아래에서 간략하게 설명합니다.

- `COPY`. 자원을 복사하는 데 사용됩니다. 모음 복사에는 `Depth:` 헤더가 사용되며 `Destination:` 헤더는 대상을 지정합니다. COPY 메소드에서는 해당되는 경우 `Overwrite:` 헤더도 사용합니다.
- `MOVE`. 자원을 이동하는 데 사용됩니다. 모음 이동에는 `Depth:` 헤더가 사용되며 `Destination:` 헤더는 대상을 지정합니다. MOVE 메소드에서는 해당되는 경우 `Overwrite:` 헤더도 사용합니다.
- `MKCOL`. 새 모음을 만드는 데 사용됩니다. 이 메소드를 사용하면 PUT 메소드의 오버로드를 방지할 수 있습니다.
- `PROPPATCH`. 단일 자원에 있는 등록 정보를 설정, 변경 또는 삭제하는 데 사용됩니다.
- `PROPFIND`. 하나 이상의 자원에 속한 하나 이상의 등록 정보를 가져오는 데 사용됩니다. 클라이언트에서 모음에 있는 PROPFIND 요청을 서버로 제출하는 경우 요청에는 Depth: 헤더(값은 `0`, `1` 또는 `infinity`)를 포함할 수 있습니다.
  - `0`. 지정된 URI의 모음 등록 정보를 가져오도록 지정합니다.
  - `1`. 지정된 URI 바로 아래에 있는 모음 및 자원의 등록 정보를 가져오도록 지정합니다.
  - `infinity`. 모음의 등록 정보와 포함되어 있는 모든 구성원 URI를 가져오도록 지정합니다. Depth 값이 infinite인 요청은 전체 모음을 탐색하기 때문에 서버에 큰 부담을 줄 수 있습니다.

  LOCK. 자원에 잠금을 추가합니다. `Lock-Token:` 헤더를 사용합니다.
- UNLOCK. 자원에서 잠금을 제거합니다. `Lock-Token:` 헤더를 사용합니다.

인스턴스 수준에서 WebDAV 활성화

Administration Server를 사용하여 전체 서버에 대해 WebDAV를 활성화할 수 있습니다. 이렇게 하면 다음 지시문이 WebDAV 플러그인을 로드하는 `magnus.conf` 파일에 추가됩니다.

```
Init fn="load-modules" shlib="/s1ws6.1/lib/libdavplugin.so" funcs="init-dav,ntrans-dav,pcheck-dav,service-dav"
shlib_flags="(global|now)"
Init fn="init-dav" LateInit=yes
```

`init-dav` Init 기능은 WebDAV 하위 시스템을 초기화 및 등록합니다.

WebDAV를 활성화하려면 CLI에서 다음 명령을 실행합니다.

```
wadm> enable-webdav --user=admin --password-file=admin.pwd
--host=serverhost --port=8989 --config=test
```

CLI 참조 `enable-webdav`(1)를 참조하십시오.

WebDAV 모음 관리

WebDAV 모음 활성화

WebDAV 모음을 활성화하려면 다음 명령을 실행합니다.

```
wadm> enable-dav-collection --user=admin --password-file=admin.pwd
--host=serverhost --port=8989 --config=config1 --vs=config1_vs_1 --uri=/dav_config1
```

CLI 참조 `enable-dav-collection`(1)을 참조하십시오.

WebDAV 모음 비활성화

WebDAV 모음을 비활성화하려면 다음 명령을 실행합니다.

```
wadm> disable-dav-collection --user=admin --password-file=admin.pwd
--host=serverhost --port=8989 --config=config1 --vs=config1_vs_1 --uri=/dav_config1
```

CLI 참조 `disable-dav-collection`(1)을 참조하십시오.

WebDAV 모음 추가

WebDAV 모음을 추가하려면 다음 명령을 실행합니다.

```
wadm> create-dav-collection --user=admin --password-file=admin.pwd
--host=serverhost --port=8989 --config=config1 --vs=config1_vs_1 --uri=/dav_config1
--source-uri=/dav_config1
```

CLI 참조 `create-dav-collection`(1)을 참조하십시오.

WebDAV 모음 나열

WebDAV 모음을 모두 나열하려면 다음 명령을 실행합니다.

```
wadm> list-dav-collections --user=admin --password-file=admin.pwd --host=serverhost
--port=8989 --config=config1 --vs=config1_vs_1
```

CLI 참조 `list-dav-collections`(1)를 참조하십시오.

WebDAV 모음 제거

WebDAV 모음을 제거하려면 다음 명령을 실행합니다.

```
wadm> delete-dav-collection --user=admin --password-file=admin.pwd --host=serverhost
--port=8989 --config=config1 --vs=config1_vs_1 --uri=/dav_config1
```

CLI 참조 `delete-dav-collection`(1)을 참조하십시오.

WebDAV 등록 정보 구성

WebDAV 등록 정보 설정

서버 수준에서 WebDAV 등록 정보를 설정하려면 다음 명령을 실행합니다.

```
wadm> set-webdav-prop --user=admin --password-file=admin.pwd --host=serverhost
--port=8989 --config=config1 acl-max-entries=120
```

CLI 참조 `set-webdav-prop`(1)를 참조하십시오.

WebDAV 등록 정보 보기

서버 수준에서 WebDAV 등록 정보를 보려면 다음 명령을 실행합니다.

```
wadm> get-webdav-prop --user=admin --password-file=admin.pwd --host=serverhost
--port=8989 --config=config1
```

CLI 참조 `get-webdav-prop`(1)를 참조하십시오.

WebDAV 모음 등록 정보 설정

WebDAV 모음 등록 정보를 설정하려면 다음 명령을 실행합니다.

```
wadm> set-dav-collection-prop --user=admin --password-file=admin.pwd --host=serverhost
--port=8989 --config=config1 --vs=config1_vs_1 --uri=/dav_config1 min-lock-timeout=1
```

CLI 참조 `set-dav-collection-prop`(1)를 참조하십시오.

WebDAV 모음 등록 정보 보기

WebDAV 모음 등록 정보를 보려면 다음 명령을 실행합니다.

```
wadm> get-dav-collection-prop --user=admin --password-file=admin.pwd --host=serverhost
--port=8989 -config=config1 --vs=config1_vs_1 --uri=/dav_config1
```

CLI 참조 `get-dav-collection-prop`(1)를 참조하십시오.

WebDAV 매개 변수 수정

다음 표에는 일반적인 WebDAV 등록 정보 중 일부가 나열되어 있습니다.

표 10–1 WebDAV 매개 변수

매개 변수	설명
잠금 데이터베이스 경로	잠금 데이터베이스를 유지 관리하는 디렉토리를 지정합니다.
최소 잠금 시간 초과	잠금의 최소 사용 시간을 초 단위로 지정합니다. **-1** 값은 잠금 시간이 초과되지 않음을 나타냅니다. 이 값은 잠금이 자동으로 제거되기 전에 요소가 잠기는 시간 길이를 나타냅니다.
최대 요청 크기	XML 요청 본문의 최대 크기를 지정합니다. 서비스 거부 공격의 가능성을 예방하려면 이 값을 구성해야 합니다. 기본값은 **8192(8K)**입니다.
최대 확장 등록 정보 깊이	PROPFIND 요청의 깊이를 지정합니다. **0**은 지정된 자원에만 적용되며 기본값입니다. **1**은 지정된 자원과 그 다음 수준에 적용됩니다. 무제한은 지정된 자원과 여기에 포함된 모든 자원에 적용됩니다. 이 매개 변수의 크기를 제한하여 과도한 메모리 소모를 방지합니다.
기본 소유자	모음의 기본 소유자입니다.
URI	WebDAV를 활성화할 기존 루트 URI입니다.
최대 PROPFIND 깊이	모음으로 전송되는 PROPFIND 요청의 최대 깊이입니다.
잠금 데이터베이스 업데이트 간격	WebDAV 잠금 데이터베이스를 디스크와 동기화하는 간격입니다. WebDAV 잠금 정보를 캐시하지 않으려면 0을 사용합니다.
인증 데이터베이스	사용할 ACL 인증 데이터베이스입니다.
인증 방법	사용할 인증 방법입니다. 기본 인증 방법은 Basic입니다.

표 10–1 WebDAV 매개 변수 *(계속)*

매개 변수	설명
인증 프롬프트 텍스트	인증을 요청할 때 클라이언트에 표시할 프롬프트입니다.
DAV ACL 데이터베이스	
최대 항목 수	단일 자원에서 허용할 최대 ACE 수입니다. 0–2147.0483647.0. 제한이 없는 경우 —1을 지정합니다.
최대 크기	모음의 WebDAV ACL 데이터베이스 메모리 표현에 허용되는 최대 크기입니다. 0–2147.0483647.0. 제한이 없는 경우 —1을 지정합니다.
업데이트 간격	WebDAV ACL 데이터베이스를 디스크와 동기화하는 간격입니다. 0.001–3600초. WebDAV ACL목록을 캐시하지 않으려면 0을 지정합니다.
DAV 등록 정보 데이터베이스	
최대 크기	WebDAV 등록 정보 데이터베이스 파일의 최대 크기입니다. 0–2147.0483647.0. 제한이 없는 경우 —1을 지정합니다.
업데이트 간격	WebDAV 등록 정보 데이터베이스를 디스크와 동기화하는 간격입니다. 0.01–3600초. WebDAV 등록 정보를 캐시하지 않으려면 0을 지정합니다.

서버 수준에서 WebDAV 비활성화

서버 수준에서 WebDAV를 비활성화하려면 다음 명령을 실행합니다.

```
wadm> disable-webdav --user=admin --password-file=admin.pwd --host=serverhost
--port=8989 --config=config1
```

CLI 참조 `disable-webdav(1)`를 참조하십시오.

WebDAV 인증 데이터베이스 관리

관리 콘솔에서 선택한 구성의 **WebDAV 탭**을 눌러 WebDAV 인증 데이터베이스 설정을 편집합니다. 다음 표에서는 페이지의 각 필드에 대한 간단한 설명을 제공합니다.

표 10–2 WebDAV 인증 데이터베이스 등록 정보

등록 정보	설명
인증 데이터베이스	**인증 데이터베이스**를 사용하여 서버가 사용자를 인증하는 데 사용할 데이터베이스를 선택할 수 있습니다. 기본값은 **keyfile**입니다.

표 10–2 WebDAV 인증 데이터베이스 등록 정보 *(계속)*

등록 정보	설명
인증 방법	■ **Basic** — HTTP 기본 메소드를 사용하여 클라이언트에서 인증 정보를 가져옵니다. 서버에 대해 SSL을 사용하는 경우 사용자 이름 및 암호는 네트워크를 통해서만 암호화됩니다. ■ **SSL** — 클라이언트 인증서를 사용하여 사용자를 인증합니다. 이 방법을 사용하려면 반드시 서버에 SSL을 사용해야 합니다. 암호화를 사용하는 경우 Basic과 SSL 메소드를 조합할 수 있습니다. ■ **Digest** — 사용자 이름과 비밀번호를 일반 텍스트로 보내지 않고 브라우저가 사용자 이름과 비밀번호를 기준으로 인증하는 방법을 제공하는 인증 기법을 사용합니다. 브라우저는 *MD5* 알고리즘을 사용하여 웹 서버에서 제공하는 사용자 비밀번호 및 일부 정보를 사용하는 Digest 값을 만듭니다. Digest를 사용하려면 배후의 auth-db에서도 digest를 지원해야 합니다. 즉, digestfile을 사용하는 파일 auth-db 또는 Digest 인증 플러그인이 설치된 LDAP auth-db를 의미합니다. ■ **기타** — 액세스 제어 API를 사용하여 만들어진 사용자 정의 메소드를 사용합니다.
인증 프롬프트 텍스트	**인증 확인 프롬프트** 옵션을 사용하여 인증 대화 상자에 표시되는 메시지 텍스트를 입력할 수 있습니다. 이 텍스트를 사용하여 사용자가 입력해야 하는 내용을 설명할 수 있습니다. 브라우저에 따라 사용자는 프롬프트의 처음 40자 정도만 볼 수 있습니다. 웹 브라우저는 일반적으로 사용자 이름과 비밀번호를 캐시하고 프롬프트 텍스트에 연결합니다. 사용자가 동일한 프롬프트를 가지는 서버의 파일 및 디렉토리에 액세스하는 경우에는 사용자 이름과 비밀번호를 다시 입력하지 않아도 됩니다. 특정 파일 및 디렉토리에 대해 사용자 인증을 원하는 경우 간단히 해당 자원에 대한 ACL용 프롬프트를 변경하면 됩니다.

WebDAV 사용 가능 서버에서 소스 URI 및 Translate:f 헤더 사용

WebDAV 메소드는 자원 또는 모음의 소스에서 작동합니다. GET 및 PUT 등의 HTTP 메소드는 WebDAV 프로토콜에 의해 오버로드되기 때문에 이런 메소드가 있는 요청은 자원의 소스에 대한 요청이 될 수도 있고 자원의 내용(출력)에 대한 요청이 될 수도 있습니다.

Microsoft 및 기타 많은 WebDAV 공급업체에서 요청과 함께 `Translate:f` 헤더를 전송하여 요청이 소스에 대한 것임을 서버에 알리는 방법으로 이 문제를 해결했습니다. 흔히 사용되는 WebDAV 클라이언트인 Microsoft WebFolders와의 상호 운용성을 위해, Sun Java System Web Server 7.0은 `Translate:f` 헤더를 자원의 소스에 대한 요청으로

인식합니다. Translate:f 헤더를 보내지 않는 클라이언트를 수용하기 위해 Sun Java System Web Server는 소스 URI를 정의합니다.

WebDAV 사용 모음의 경우 URI에 대한 요청에서는 자원의 내용(출력)을 검색하고 소스 URI에 대한 요청에서는 자원의 소스를 검색합니다. Translate:f 헤더가 있는 URI에 대한 요청은 소스 URI에 대한 요청으로 처리됩니다.

기본적으로 자원의 소스에 대한 모든 액세스는 서버 인스턴스 특정 ACL 파일에 다음 선언이 있는 dav-src ACL에 의해 거부됩니다.

```
deny (all) user = "anyone";
```

사용자는 소스 URI에 대한 액세스 권한을 추가하여 사용자가 소스에 액세스할 수 있도록 설정할 수 있습니다.

자원 잠금 및 잠금 해제

Sun Java System Web Server를 사용하면 서버 관리자가 자원을 잠궈 해당 자원에 대한 액세스를 일련화할 수 있습니다. 잠금을 사용하면 특정 자원에 액세스하는 사용자가 있을 경우 다른 사용자가 같은 자원을 수정할 수 없습니다. 이 방법을 사용하면 여러 사용자가 서버에서 자원을 공유하기 때문에 발생하는 "업데이트 유실" 문제를 해결할 수 있습니다. 서버에서 유지 관리하는 잠금 데이터베이스는 클라이언트에서 발행하고 사용하는 잠금 토큰을 추적합니다.

Sun Java System Web Server는 항상 모든 자원에 대해 고유하도록 고안된 opaquelocktoken URI 체계를 지원합니다. 여기서는 ISO-1157.08에 기술된 것과 같은 UUID(Universal Unique Identifier) 기법을 사용합니다.

Sun Java System Web Server에서는 두 가지 유형의 잠금 기법을 인식합니다.

- 전용 잠금
- 공유 잠금

전용 잠금

전용 잠금은 단일 사용자에게만 자원 액세스 권한을 부여하는 잠금입니다. 다른 사용자는 자원의 전용 잠금이 제거된 후에만 같은 자원에 액세스할 수 있습니다.

전용 잠금은 자원을 잠글 때 사용하기에 너무 경직되고 비용이 큰 경우가 많습니다. 예를 들어 프로그램이 중단되거나 잠금 소유자가 자원 잠금 해제를 잊어버린 경우 전용 잠금을 제거하려면 잠금 시간이 초과되거나 관리자가 개입해야 합니다.

공유 잠금

공유 잠금을 사용하면 여러 사용자가 자원에 대한 잠금을 받을 수 있습니다. 따라서 적절한 액세스 권한이 있는 사용자는 누구나 잠금을 얻을 수 있습니다.

공유 잠금을 사용하는 경우 잠금 소유자는 다른 통신 채널을 사용하여 작업을 조정할 수 있습니다. 공유 잠금의 목적은 공동 작업자가 자원에서 어떤 사람이 작업 중인지 알 수 있게 하는 것입니다.

최소 잠금 시간 초과

`server.xml` 파일에서 DAV 또는 `DAVCOLLECTION` 객체의 `minlocktimeout` 속성 값을 구성하여 잠금을 제어할 수 있습니다. `minlocktimeout` 속성은 잠금의 최소 사용 시간을 초 단위로 지정합니다. 이 값은 잠금이 자동으로 제거되기 전에 요소가 잠기는 시간 길이를 나타냅니다.

이 속성은 선택적인 속성입니다. 값을 -1로 설정하면 잠금이 만료되지 않습니다. 값을 `0`으로 설정하면 모음의 모든 자원을 요청에 지정된 Timeout 헤더로 잠글 수 있습니다.

Timeout 헤더가 지정되어 있지 않으면 자원은 제한 시간 무한으로 잠깁니다. 요청의 Timeout 헤더가 `Infinite` 값으로 설정되어 있는 경우에도 자원은 제한 시간 무한으로 잠깁니다.

WebDAV 자원에 대한 요청의 Timeout 헤더 값이 `server.xml`에 지정된 `minlocktimeout` 값과 같거나 크면 자원은 요청에 지정된 시간 동안 잠깁니다.

그러나 자원의 Timeout 헤더 값이 `server.xml`에 지정된 `minlocktimeout` 값 미만인 경우에는 자원이 `server.xml`에 지정된 `minlocktimeout` 값으로 잠깁니다.

다음 표에서는 Sun Java System Web Server로 잠금 요청을 처리하는 방법을 보여줍니다.

표 10–3 Sun Java System Web Server에서 잠금 요청을 처리하는 방법

요청의 Timeout 헤더 값 설정	리소스
`Infinite`	시간 초과가 -1(무한)로 설정되어 잠김
None	시간 초과가 -1(무한)로 설정되어 잠김
`Second-`*xxx*	■ *xxx* 값으로 잠김, *xxx*가 `server.xml`에 설정된 `minlocktimeout` 값과 같거나 클 경우 또는 ■ `server.xml`에 지정된 `minlocktimeout` 값으로 잠김, *xxx*가 `server.xml`에 설정된 `minlocktimeout` 값보다 작을 경우.

주 – CLI 사용

CLI를 통해 잠금 만료를 설정하려면 다음 명령을 실행합니다.

```
wadm> expire-lock  --user=admin --password-file=admin.pwd
--host=serverhost--port=8989 --config=config1 --vs=config1
--collection-uri=/dav1 --lock-uri=/dav1/file.html
--opaque-token=opaquelocktoken
```

CLI 참조 `expire-lock`(1)을 참조하십시오.

위의 예에서 opaque-token은 만료되도록 설정할 잠금의 아이디를 지정합니다.

CLI를 통해 기존 잠금을 표시하려면 다음 명령을 실행합니다.

```
wadm> list-locks --user=admin --password-file=admin.pwd --host=serverhost
--port=8989 --config=config1 --vs config1 --collection-uri=/dav1
```

CLI 참조 `list-locks`(1)를 참조하십시오.

◆ ◆ ◆ 11장

11 Java 및 웹 응용 프로그램 작업

이 장에서는 가상 서버의 Java 설정을 편집하는 절차를 설명합니다. 관리 콘솔이나 `wadm` 명령줄 도구에서 Java 설정을 편집할 수 있습니다. 이 장에서는 Sun Java System Web Server에서 구성할 수 있는 다양한 Java 자원에 대해서도 설명합니다.

이 장에서는 Sun Java System Web Server에서 Java 웹 응용 프로그램을 배포하는 방법에 대해서도 설명합니다.

Sun Java System Web Server에서 사용하도록 Java 구성

이 절에서는 선택한 구성에 대해 Java를 활성화하고 Java Home 변수를 설정할 수 있게 합니다.

▼ 구성에 대해 Java 활성화

1 **구성을 선택합니다.**
구성 목록에서 구성을 선택합니다. 구성 탭을 눌러 사용 가능한 구성 목록을 가져옵니다.

2 **Java > 일반 탭을 누릅니다.**

3 **Java 사용 확인란을 선택합니다.**
구성에 대한 Java 지원을 설정하거나 해제합니다. Java를 사용하면 서버에서 Java 응용 프로그램을 처리할 수 있습니다.

4 **Java Home을 설정합니다.**
Java SE의 위치를 지정합니다. 절대 경로 또는 서버의 `config` 디렉토리에 대한 상대 경로를 지정합니다.

5 **고정 첨부를 설정합니다.**
서버에서 각 HTTP 요청 처리 스레드를 JVM에 한 번만 첨부할지 여부를 지정합니다. 그렇지 않을 경우 서버는 각 요청에 대해 HTTP 요청 처리 스레드를 첨부/분리합니다.

주 – CLI 사용

구성에 대해 Java를 활성화하려면 다음 명령을 실행합니다.

```
wadm> enable-java --user=admin --password-file=admin.pwd --host=serverhost
--port=8989 --config=config1
```

CLI 참조 `enable-java`(1)를 참조하십시오.

Java 클래스 경로 설정

이 절에서는 선택된 구성에 대해 JVM 클래스 경로를 추가할 수 있습니다.

▼ Java 클래스 경로를 설정하는 방법

1 **구성을 선택합니다.**
구성 목록에서 구성을 선택합니다. 구성 탭을 눌러 사용 가능한 구성 목록을 가져옵니다.

2 **Java > 경로 설정 탭을 누릅니다.**
다음 매개 변수를 편집합니다.

- **환경 클래스 경로 무시** — 기본적으로 사용됩니다.
- **클래스 경로 접두어** — 시스템 클래스 경로의 접두어입니다. XML 파서 클래스와 같은 시스템 클래스를 대체하려면 시스템 클래스 경로 접두어만 사용해야 합니다. **사용 시 주의**.
- **서버 클래스 경로** — 서버 클래스를 포함하는 클래스 경로입니다. 읽기 전용 목록입니다.
- **클래스 경로 접미어** — 서버 클래스 경로에 추가합니다.

- **원시 라이브러리 경로 접두어** — 운영 체제 원시 라이브러리 경로의 접두어입니다.
- **Bytecode 프로세서 클래스** — `com.sun.appserv.BytecodePreprocessor`를 구현하는 클래스의 정규화된 이름입니다. 런타임 클래스 계측을 수행하는 일반적인 방법은 선행 처리 기법을 사용하는 것입니다. 프로필링 및 모니터링 도구는 클래스 선행 프로세서를 사용하여 계측 코드를 JVM에서 로드하기 직전에 Java 클래스의 필요한 위치에 삽입합니다. 이를 위해 클래스 선행 프로세서는 클래스 로더와 함께 작동합니다.

JVM 구성

관리 인터페이스에서 JVM 명령줄 옵션을 설정하려면 다음 작업을 수행합니다.

▼ JVM 구성

1 **구성을 선택합니다.**
구성 목록에서 구성을 선택합니다. 구성 탭을 눌러 목록을 가져옵니다.

2 **Java > JVM 설정 탭을 누릅니다.**
JVM의 설정을 구성합니다.

JVM 옵션 추가

여기에 값을 지정하여 명령줄 JVM 옵션을 추가/삭제할 수 있습니다.

JVM 옵션 추가 버튼을 눌러 JVM 옵션을 추가합니다.

JVM 옵션에 대한 예는 다음과 같습니다.`-Djava.security.auth.login.config=login.conf, -Djava.util.logging.manager=com.iplanet.ias.server.logging.ServerLogManager and -Xms128m -Xmx256m`

주 – CLI 사용

CLI를 통해 JVM 옵션을 추가하려면 다음 명령을 실행합니다.

```
wadm> create-jvm-options --user=admin --password-file=admin.pwd --host=serverhost
--port=8989 --config=config1 -Dhttp.proxyHost=proxyhost.com -Dhttp.proxyPort=8080
```

CLI 참조 `create-jvm-options`(1)를 참조하십시오.

JVM 프로필러 추가

JVM 프로필러는 Java 응용 프로그램에서 성능 문제, 메모리 누수, 다중 스레딩 문제 및 시스템 자원 사용 문제를 진단하고 해결할 수 있게 하여 응용 프로그램에 대해 최고 수준의 안정성 및 확장 가능성을 보장합니다.

▼ JVM 프로필러 추가

1 **구성을 선택합니다.**
구성 목록에서 구성을 선택합니다. 구성 탭을 눌러 사용 가능한 구성 목록을 가져옵니다.

2 **Java > JVM 설정 탭을 누릅니다.**

3 **프로필러 섹션에서 새로 만들기 버튼을누릅니다.**

4 **다음 매개 변수의 값을 입력합니다.**

- **이름** — 새 JVM 프로필러의 약식 이름을 지정합니다.
- **사용 가능** — 프로필러가 런타임에 사용되는지 여부를 결정합니다.
- **클래스 경로** — 프로필러의 유효한 클래스 경로를 지정합니다. (선택 사항).
- **원시 라이브러리 경로** — 유효한 원시 라이브러리 경로를 지정합니다. (선택 사항).
- **JVM 옵션** — CLI에 대해 추가 JVM 옵션을 지정할 수 있습니다.

주 – CLI 사용

CLI를 통해 JVM 프로필러를 추가하려면 다음 명령을 실행합니다.

```
wadm> create-jvm-profiler --user=admin --password-file=admin.pwd
--host=serverhost --port=8989 --config=config1
```

CLI 참조 `create-jvm-profiler(1)`를 참조하십시오.

서버에 대해 Java 디버깅 활성화

JVM은 디버그 모드에서 시작할 수 있으며 JPDA(Java Platform Debugger Architecture) 디버거에 연결할 수 있습니다. 디버깅을 사용할 때 로컬 및 원격 디버깅을 모두 사용할 수 있습니다.

Sun Java System Web Server의 디버깅은 JPDA 소프트웨어를 기반으로 수행됩니다. 디버깅을 사용하려면 다음 작업을 수행하십시오.

▼ JVM 디버깅 사용

1 **구성을 선택합니다.**
구성 목록에서 구성을 선택합니다. 구성 탭을 눌러 사용 가능한 구성 목록을 가져옵니다.

2 **Java > JVM 설정 탭을 누릅니다.**

3 **디버그 Java 설정에 있는 디버그 사용 확인란을 선택합니다.**

4 **필요에 따라 새로 만들기 버튼을 눌러 JVM 옵션을 입력합니다.**
기본 JPDA 옵션은 다음과 같습니다.

```
-Xdebug -Xrunjdwp:transport=dt_socket,server=y,suspend=n,address=7896
```

`suspend=y`로 대체하면 JVM은 중지 모드로 시작되며 디버거가 첨부될 때까지 중지 상태로 남아 있습니다. JVM이 시작되자마자 디버깅을 시작하려는 경우에 유용합니다. JVM을 디버거에 첨부할 때 사용할 포트를 지정하려면 `address=port_number`를 지정합니다. 디버깅 옵션 목록은 JPDA 문서를 확인하십시오.

Java 웹 응용 프로그램 배포

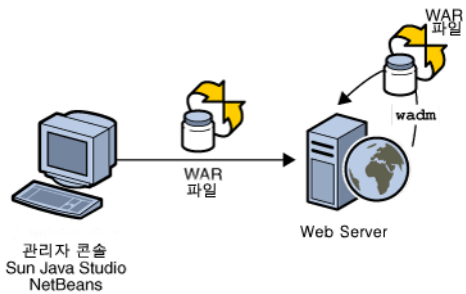
웹 응용 프로그램 추가

기존 가상 서버에 웹 응용 프로그램을 배포할 수 있습니다.

▼ 웹 응용 프로그램을 배포하는 방법

시작하기 전에

- 웹 응용 프로그램을 배포할 가상 서버를 식별해야 합니다.
- 웹 응용 프로그램 아카이브(.war 파일)를 가지고 있거나 서버의 웹 응용 프로그램 경로를 알아야 합니다.



웹 응용 프로그램은 `wadm`, 관리 콘솔 및 기타 지원되는 IDE를 통해 배포할 수 있습니다.

1. **웹 응용 프로그램을 배포하려면 서버 구성 아래에 있는 가상 서버 탭을 누릅니다.**
2. **웹 응용 프로그램을 배포할 가상 서버를 선택합니다.**
3. **웹 응용 프로그램 탭 > 새로 만들기 버튼을 누릅니다.**
4. **웹 응용 프로그램 패키지를 지정합니다.**
   웹 응용 프로그램 아카이브를 업로드하려면 찾아보기 버튼을 누르고 아카이브를 선택합니다. 서버에 있는 웹 응용 프로그램 아카이브를 지정할 수도 있습니다(선택 사항).
5. **웹 응용 프로그램에 대한 URI를 지정합니다. 이는 응용 프로그램 컨텍스트 루트이며 서버 호스트에 상대적입니다.**
6. **웹 응용 프로그램에 대한 간단한 설명을 제공합니다.**
7. **JSP 사전 컴파일을 활성화/비활성화합니다.**
   이 지시문을 활성화하면 웹 응용 프로그램에 있는 모든 JSP를 사전 컴파일하여 성능을 향상시킬 수 있습니다.
8. **응용 프로그램을 활성화합니다.**
   웹 응용 프로그램 상태가 사용 불가능으로 설정되어 있으면 요청 시 사용할 수 없습니다. 하지만 응용 프로그램을 인스턴스에 다시 배포하지 않고도 이 옵션을 언제든지 전환할 수 있습니다.
9. **응용 프로그램을 배포합니다.**
   배포 버튼을 눌러 웹 응용 프로그램을 배포합니다.

   컨텍스트 루트를 지정하여 응용 프로그램에 액세스할 수 있습니다. 예를 들면 `http://<your-server>:<port>/<URI>`와 같습니다.

주 – CLI 사용

```
wadm> add-webapp --user=admin --password-file=admin.passwd --host=localhost
 --port=8888 --config=config1 --vs=HOSTNAME --uri=/hello /home/test/hello.war
```

CLI 참조 `add-webapp`(1)를 참조하십시오.

웹 응용 프로그램 디렉토리 배포

`–file-on-server` 옵션을 사용하여 Administration Server 호스트 시스템에 있는 디렉토리를 구성에 배포할 수 있습니다. 다음 명령을 실행합니다.

```
wadm> add-webapp --user=admin-user --password-file=admin.passwd
--port=8989 --vs=vs1 --config=config1 --file-on-server
--uri=/mywebapp /space/tmp/mywebapp
```

배포 중에 JSP 사전 컴파일

웹 응용 프로그램을 배포하는 동안 웹 응용 프로그램에서 JSP를 사전 컴파일하려면 –precompilejsp 옵션을 다음과 같이 설정하여 명령을 실행합니다.

```
wadm> add-webapp --user=admin-user --password-file=admin.passwd
--port=8989 --vs=vs1 --config=config1 --file-on-server --uri=/mywebapp
--precompilejsp mywebapp.war
```

서블릿 컨테이너 구성

이 절에서는 서블릿 컨테이너를 구성하는 절차에 대해 설명합니다.

▼ 서블릿 컨테이너 설정

1 **구성을 선택합니다.**
구성 목록에서 구성을 선택합니다. 구성 탭을 눌러 목록을 가져옵니다.

2 **Java > 서블릿 컨테이너를 누릅니다.**

서블릿 컨테이너 전역 매개 변수

다음 표에서는 서블릿 컨테이너 페이지에서 사용할 수 있는 매개 변수에 대해 설명합니다.

표 11–1 서블릿 컨테이너 매개 변수

매개 변수	설명
로그 수준	서블릿 컨테이너의 로그 상세 표시입니다. 사용 가능한 값은 최고(가장 많음), 보통, 최저, 정보, 경고, 실패, 구성, 보안 또는 치명적 오류(가장 적음)입니다.
동적 다시 로드 간격	이 매개 변수는 배포된 웹 응용 프로그램을 서버에서 확인하여 수정하기 전에 대기할 시간을 정의합니다. 값의 범위는 1 - 60이며 동적 다시 로드를 사용하지 않는 경우에는 –1입니다.

표 11–1 서블릿 컨테이너 매개 변수 *(계속)*

매개 변수	설명
익명 역할	모든 사용자에게 할당된 기본 또는 익명 역할의 이름입니다. 기본 역할은 ANYONE입니다.
서블릿 풀 크기	`SingleThreadedServlet`당 인스턴스화할 서블릿 인스턴스 수입니다. 값의 범위는 1 - 4096입니다.
디스패처 최대 깊이	중첩된 요청 디스패치를 허용하는 서블릿 컨테이너의 최대 깊이입니다. 가능한 값의 범위는 0 - 2147.0483647.0이며 기본값은 20입니다.
교차 컨텍스트 허용	요청 디스패처에서 다른 컨텍스트로 디스패치할 수 있는지 여부를 지정합니다. 기본값은 false입니다.
쿠키 인코딩	서블릿 컨테이너에서 쿠키 값을 인코딩할지 여부를 지정합니다. 기본값은 true입니다.
세션 아이디 다시 사용	클라이언트의 새 세션을 만들 때 기존 세션 아이디 번호를 다시 사용할지 여부를 지정합니다. 기본값은 false입니다.
보안 세션 쿠키	Dynamic/True/False. 이 매개 변수는 JSESSIONID 쿠키에 보안을 표기하는 조건을 제어합니다. 보안 연결(HTTPS)에서 요청을 받은 경우에만 쿠키에 보안을 표시하려면 Dynamic(기본값)을 사용합니다. 항상 보안 표시를 하려면 True를 사용하고 보안 표시를 하지 않으려면 False를 사용합니다.

서버 라이프사이클 모듈 구성

Java Server 라이프사이클 모듈은 시작 또는 중지와 같은 서버 이벤트가 발생할 때마다 특정 작업을 수행하기 위해 서버 라이프사이클 이벤트를 수신하는 Java 클래스입니다.

서버에서는 웹 서버 환경 내에서 단기 또는 장기 Java 기반 작업 실행을 지원합니다. 이러한 작업은 서버 시작 시 자동으로 시작되고 서버 종료 시 알림이 수신됩니다. 따라서 싱글톤 및 RMI 서버의 인스턴스화와 같은 작업에 연결할 수 있습니다.

다음은 서버 라이프사이클에 대한 간단한 설명입니다.

서버 라이프사이클 소개

- **초기화** — 이 단계에는 구성 읽기, 내장 하위 시스템 초기화(이름 지정, 보안 및 로깅 서비스), 웹 컨테이너 만들기 등이 포함됩니다.
- **시작** — 이 단계에는 배포된 응용 프로그램의 로딩 및 초기화가 포함됩니다.
- **서비스** — 서버에서 요청을 처리할 준비가 되어 있습니다.

- **종료** — 이 단계에서는 로드된 응용 프로그램을 중지하고 완전히 삭제합니다. 시스템이 종료할 준비를 합니다.
- **종료** — 이 단계에서는 내장 하위 시스템과 서버 런타임 환경을 종료합니다. 이 단계 이후에는 더 이상 작업이 실행되지 않습니다.
- **다시 구성** — 서버가 서비스 상태인 동안 서버 스레드가 동적으로 다시 구성되는 임시 서버 상태입니다. 이 단계는 서버의 주기 동안 여러 번 발생할 수 있습니다.

▼ 라이프사이클 모듈을 추가하는 방법

1 **구성을 선택합니다.**
구성 목록에서 구성을 선택합니다. 구성 목록을 보려면 구성 탭을 누릅니다.

2 **Java > 라이프사이클 모듈 탭을 누릅니다.**

3 **새로 만들기 버튼을 누릅니다.**
다음 매개 변수에 대한 값을 제공합니다.

- **이름** — 새 라이프사이클 모듈에 유효한 고유 이름을 제공합니다.
- **사용 가능** — 이 라이프사이클 모듈을 사용하려면 이 옵션을 사용합니다.
- **클래스 이름** — 정규화된 Java 클래스 이름입니다. 클래스는 `com.sun.appserv.server.LifecycleListener` 인터페이스를 구현해야 합니다. 이 인터페이스 사용에 대한 자세한 내용은 *Developer's Guide*를 참조하십시오.
- **클래스 경로** — 선택 사항입니다. Listener 클래스의 클래스 경로를 지정할 수 있습니다.
- **로드 순서** — 100보다 큽니다. 라이프사이클 이벤트 Listener의 로드 순서를 숫자로 나타냅니다. 내부 라이프사이클 모듈과의 충돌을 피하려면 100 이상의 로드 순서를 선택하는 것이 좋습니다.
- **로드 실패 시** — 이 옵션을 사용하면 서버가 Listener 클래스에서 발생한 예외를 치명적인 것으로 처리하지 않으므로 서버가 정상적으로 시작됩니다. 기본적으로 사용 안 함으로 설정됩니다.
- **설명** — 라이프사이클 모듈에 대한 간단한 설명을 제공합니다.
- **등록 정보** — 등록 정보는 Java 라이프사이클 모듈에 인수를 전달하는 데 사용할 수 있습니다. 새 등록 정보를 추가하려면 등록 정보 추가 버튼을 누르고 이름, 값 및 설명 텍스트를 입력합니다.

주의 – 서버 라이프사이클 Listener 클래스는 주 서버 스레드에서 동시에 호출되므로 Listener 클래스가 서버를 차단하지 않도록 특별히 주의해야 합니다. 필요한 경우 Listener 클래스에서 스레드를 만들 수도 있으나 셧다운/종료 단계 동안에는 해당 클래스를 중지해야 합니다.

▼ 라이프사이클 모듈을 삭제하는 방법

1. **구성을 선택합니다.**
   구성 목록에서 구성을 선택합니다. 구성 목록을 보려면 구성 탭을 누릅니다.
2. **Java > 라이프사이클 모듈 탭을 누릅니다.**
3. **라이프사이클 모듈을 선택하고 라이프사이클 모듈 삭제 버튼을 누릅니다.**

주 – CLI 사용

다음 예에서는 com.MyLifecycleModule 클래스로 구현되는 test 구성을 위해 myLifecycleModule이라는 Java 라이프사이클 모듈을 만드는 방법을 보여줍니다.

```
wadm> create-lifecycle-module --user=admin --password-file=admin.pwd
--host=serverhost --port=8989 --config=config1
--class=com.sun.webserver.tests.LifecycleClass LifecycleTest
```

CLI 참조 create-lifecycle-module(1)을 참조하십시오.

Java 라이프사이클 모듈을 나열하려면 다음 명령을 실행합니다.

```
wadm> list-lifecycle-modules --config=test
```

CLI 참조 list-lifecycle-modules(1)를 참조하십시오.

Java 라이프사이클 모듈에 등록 정보를 추가하려면 다음 명령을 실행합니다.

```
wadm> create-lifecycle-module-userprop --user=admin --password-file=admin.pwd
--host=serverhost --port=8989 --config=config1 --module=LifecycleTest info=Testing
```

CLI 참조 create-lifecycle-module-userprop(1)을 참조하십시오.

Java 라이프사이클 모듈 등록 정보를 수정하려면 다음 명령을 실행합니다.

```
wadm> set-lifecycle-module-prop --user=admin --password-file=admin.pwd
--host=serverhost --port=8989 --config=config1 --module=LifecycleTest
class-path=/space
```

CLI 참조 set-lifecycle-module-prop(1)를 참조하십시오.

Java 자원 구성

웹 응용 프로그램은 자원 관리자, 데이터 소스(예: SQL 데이터 소스), 전자 메일 세션 및 URL 연결 팩토리 등 다양한 자원에 액세스할 수 있습니다. Java EE 플랫폼은 JNDI(Java Naming and Directory Interface) 서비스를 통해 응용 프로그램이 이러한 자원을 사용할 수 있도록 합니다.

Sun Java System Web Server를 사용하면 다음 Java EE 자원을 만들고 관리할 수 있습니다.

- JDBC 데이터 소스
- JDBC 연결 풀
- Java 메일 세션
- 사용자 정의 자원

- 외부 JNDI 자원

JDBC 자원 구성

JDBC 데이터 소스는 Sun Java System Web Server를 사용하여 만들고 관리할 수 있는 Java EE 자원입니다.

JDBC API는 관계형 데이터베이스 시스템과의 연결을 위한 API입니다. JDBC API는 두 부분으로 이루어집니다.

- 응용 프로그램 구성 요소가 데이터베이스에 액세스하는 데 사용되는 응용 프로그램 수준 인터페이스
- JDBC 드라이버를 Java EE 플랫폼에 연결하는 서비스 공급자 인터페이스

JDBC 데이터 소스 객체는 Java 프로그래밍 언어로 데이터 소스를 구현한 것입니다. 간단히 말해 데이터 소스는 데이터를 저장하기 쉽게 하는 것입니다. 이는 대규모 기업용의 복잡한 데이터베이스처럼 정교한 것일 수도 있고 행과 열로 이루어진 파일과 같이 간단한 것일 수도 있습니다. JDBC 데이터 소스는 Sun Java System Web Server를 통해 만들고 관리할 수 있는 Java EE 자원입니다.

JDBC API는 일련의 Java용 클래스를 제공하며, 다양한 범위의 관계형 데이터베이스에 동일하게 액세스할 수 있도록 하는 표준 SQL 데이터베이스 액세스 인터페이스를 포함합니다.

JDBC를 사용하면 SQL 문을 거의 모든 데이터베이스 관리 시스템(DBMS)에 보낼 수 있습니다. 또한 관계형 및 객체 DBMS 모두에 대한 인터페이스로 사용됩니다.

JDBC 자원 추가

CLI를 통해 JDBC 자원을 추가하려면 다음 명령을 실행합니다.

```
wadm> create-jdbc-resource --user=admin --password-file=admin.pwd --host=serverhost
--port=8989 --config=config1 --datasource-class=oracle.jdbc.pool.OracleDataSource jdbc
```

CLI 참조 `create-jdbc-resource`(1)를 참조하십시오.

위의 예에서 `com.pointbase.jdbc.jdbcDataSource`는 JDBC 드라이버 클래스를 나타냅니다.

지원되는 JDBC 드라이버의 목록은 167 페이지 “Sun Java System Web Server에서 지원되는 JDBC 드라이버”를 참조하십시오.

Sun Java System Web Server에서 지원되는 JDBC 드라이버

다음 표에서는 새 JDBC 자원을 추가할 때 구성해야 하는 공통 JDBC 드라이버 및 해당 등록 정보 목록을 제공합니다. 169 페이지 “새 JDBC 자원 추가”를 참조하십시오.

표 11–2 공통 및 지원되는 JDBC 드라이버 목록

드라이버	클래스 이름	등록 정보
Oracle 드라이버	`oracle.jdbc.pool.OracleDataSource`	■ url ■ 사용자 ■ 비밀번호
Oracle용 SJS JDBC 드라이버	`com.sun.sql.jdbcx.oracle.OracleDataSource`	■ 서버 이름 ■ 포트 번호 ■ 사용자 ■ 비밀번호 ■ SID
DB2 IBM 드라이버	`com.ibm.db2.jdbc.DB2DataSource`	■ 서버 이름 ■ 데이터베이스 이름 ■ 포트 번호 ■ 사용자 ■ 비밀번호 ■ 드라이버 유형
DB2용 SJS JDBC 드라이버	`com.sun.sql.jdbcx.db2.DB2DataSource`	■ 데이터베이스 이름 ■ 위치 이름 ■ 패키지 이름 ■ 비밀번호 ■ 포트 번호 ■ 서버 이름 ■ 사용자
MS SQLServer 드라이버	`com.ddtek.jdbcx.sqlserver.SQLServerDataSource`	■ 데이터베이스 이름 ■ 비밀번호 ■ 사용자 ■ 서버 이름 ■ 포트 번호

표 11–2 공통 및 지원되는 JDBC 드라이버 목록 *(계속)*

드라이버	클래스 이름	등록 정보
MS용 SJS JDBC 드라이버	`com.sun.sql.jdbcx.sqlserver.SQLServerDataSource`	■ 데이터베이스 이름 ■ 비밀번호 ■ 사용자 ■ 서버 이름 ■ 포트 번호
Sybase 드라이버	`com.sybase.jdbcx.SybDataSource`	■ 데이터베이스 이름 ■ 비밀번호 ■ 포트 번호 ■ 서버 이름 ■ 사용자
Sybase용 SJS JDBC 드라이버	`com.sun.sql.jdbcx.sybase.SybaseDataSource`	■ 데이터베이스 이름 ■ 비밀번호 ■ 사용자 ■ 포트 번호 ■ 서버 이름
MySQL MM 드라이버	`org.gjt.mm.mysql.jdbc2.optional.MysqlDataSource`	■ 서버 이름 ■ 포트 ■ 데이터베이스 이름 ■ 사용자 ■ 비밀번호
Informix 드라이버	`com.informix.jdbcx.IfxDataSource`	■ 포트 번호 ■ 데이터베이스 이름 ■ IfxIFXHOST(Informix 데이터베이스를 실행하는 컴퓨터의 IP 주소 또는 호스트 이름) ■ 서버 이름 ■ 사용자 ■ 비밀번호
Informix용 SJS JDBC 드라이버	`com.sun.sql.jdbcx.informix.InformixDataSource`	■ 데이터베이스 이름 ■ informixServer(연결할 Informix 데이터베이스 서버의 이름) ■ 비밀번호 ■ 포트 번호 ■ 서버 이름

표 11–2 공통 및 지원되는 JDBC 드라이버 목록 *(계속)*

드라이버	클래스 이름	등록 정보
PostgreSQL 드라이버	`org.postgresql.ds.PGSimpleDataSource`	■ 서버 이름 ■ 데이터베이스 이름 ■ 포트 번호 ■ 사용자 ■ 비밀번호
Apache Derby 드라이버	`org.apache.derby.jdbc.EmbeddedDataSource`	■ 데이터베이스 이름 ■ 사용자 ■ 비밀번호

JDBC 자원 관리

▼ 새 JDBC 자원 추가

1 구성을 선택합니다.
구성 목록에서 구성을 선택합니다. 구성 탭을 눌러 목록을 가져옵니다.

2 Java > 자원 탭을 누릅니다.

3 JDBC 자원 섹션 아래에 있는 새로 만들기 버튼을 누릅니다.

4 드라이버 공급업체를 선택합니다.
JNDI 이름에 고유한 값을 지정하고 사용 가능한 목록에서 JDBC 드라이버 공급업체를 선택합니다.

5 JDBC 자원 등록 정보를 제공합니다.
이전 단계에서 선택한 JDBC 드라이버 공급업체를 기반으로 드라이버의 클래스 이름 및 JDBC자원 등록 정보가 자동으로 채워집니다.

6 검토를 수행합니다.
요약을 보고 마침을 눌러 새 JDBC 자원을 만듭니다.

JDBC 연결 풀 관리

JDBC 연결 풀 구성

Sun Java System Web Server 7.0에서 JDBC 연결 풀은 `jdbc-resource` 요소를 통해 구성됩니다. 아래의 단계를 수행하여 가장 간단한 형태의 연결 풀을 구성할 수 있습니다. 이 예에서 연결 풀은 Oracle JDBC 드라이버를 사용합니다.

▼ JDBC 연결 풀을 만드는 방법

1 wadm을 시작합니다.

2 JDBC 자원을 만듭니다.

기본 구성으로 JDBC 자원을 만듭니다. 다른 속성을 사용하여 연결 풀을 미세 조정할 수 있습니다. 추가 속성 및 예는 설명서 페이지를 참조하십시오.

```
wadm> create-jdbc-resource --config=test
--datasourceclass=oracle.jdbc.pool.OracleDataSource jdbc/MyPool
```

3 공급업체별 등록 정보를 구성합니다.

등록 정보는 드라이버의 공급업체별 등록 정보를 구성하는 데 사용됩니다. 아래 예에서는 등록 정보 `url`, `user` 및 `password`가 JDBC 자원에 추가됩니다.

```
wadm> add-jdbc-resource-userprop --config=test --jndi-name=jdbc/MyPool
url=jdbc:oracle:thin:@hostname:1521:MYSID user=myuser password=mypassword
```

4 연결 확인을 활성화합니다.

풀에 대해 연결 확인을 활성화할 수 있습니다. 이 옵션을 사용하면 연결이 응용 프로그램으로 전달되기 전에 확인 과정을 거칩니다. 따라서 네트워크 고장 또는 데이터베이스 서버 장애로 인해 데이터베이스를 사용할 수 없는 경우 서버가 자동으로 데이터베이스 연결을 다시 설정할 수 있습니다. 연결 확인에는 추가 오버헤드가 필요하므로 성능이 약간 저하될 수 있습니다.

```
wadm> set-jdbc-resource-prop --config=test --jndi-name=jdbc/MyPool
connection-validation-table-name=test connection-validation=table
```

5 기본 풀 설정을 변경합니다.

이 예에서는 최대 연결 수를 변경합니다.

```
wadm> set-jdbc-resource-prop --config=test --jndi-name=jdbc/MyPool
max-connections=100
```

6 구성을 배포합니다.

```
wadm> deploy-config test
```

7 JDBC 드라이버를 포함하는 Jar 파일을 지정합니다.

서버에는 드라이버를 구현하는 클래스를 제공해야 합니다. 이 작업은 다음과 같은 두 가지 방법으로 수행할 수 있습니다.

- 드라이버의 jar 파일을 서버 인스턴스 lib 디렉토리에 복사합니다. 인스턴스 lib 디렉토리에 포함된 jar 파일이 자동으로 로드되어 서버에서 사용할 수 있게 되기 때문에 이 방법이 가장 간단한 방법입니다.
- JVM의 *class-path-suffix*가 JDBC 드라이버의 jar 파일을 포함하도록 수정합니다.

```
wadm> set-jvm-prop  --config=test class-path-suffix=/export/home/lib/classes12.jar
```

8 웹 응용 프로그램에서의 사용법

- WEB-INF/web.xml 수정

```
<web-app>
...
  <resource-ref>
    <description>JDBC Connection Pool</description>
    <res-ref-name>jdbc/myJdbc</res-ref-name>
    <res-type>javax.sql.DataSource</res-type>
    <res-auth>Container</res-auth>
  </resource-ref>
...
</web-app>
```

- WEB-INF/sun-web.xml 수정

```
<sun-web-app>
...
  <resource-ref>
    <res-ref-name>jdbc/myJdbc</res-ref-name>
    <jndi-name>jdbc/MyPool</jndi-name>
  </resource-ref>
...
</sun-web-app>
```

- 연결 풀 사용

```
Context initContext = new InitialContext();
    Context webContext = (Context)context.lookup("java:/comp/env");

    DataSource ds = (DataSource) webContext.lookup("jdbc/myJdbc");
    Connection dbCon = ds.getConnection();
```

사용자 정의 자원 등록

다음 작업을 수행하여 인스턴스에 사용자 정의 자원을 등록할 수 있습니다.

▼ 사용자 정의 자원을 추가하는 방법

1 구성을 선택합니다.
구성 목록에서 구성을 선택합니다. 구성 탭을 눌러 목록을 가져옵니다.

2 Java > 자원 탭을 누릅니다.

3 사용자 정의 자원 섹션 아래에 있는 새로 만들기 버튼을 누릅니다.

사용자 정의 자원 등록 정보

다음 표에서는 사용자 정의 자원을 만드는 데 사용할 수 있는 등록 정보에 대해 설명합니다.

표 11–3 사용자 정의 자원 등록 정보

등록 정보	설명
JNDI 이름	사용자 정의 자원에 대해 고유한 JNDI 이름을 제공합니다.
사용 가능	런타임에 이 JDBC 자원을 사용할지 여부를 결정합니다.
자원 유형	이 자원의 정규화된 유형입니다.
팩토리 클래스	이 자원 유형을 인스턴스화하는 클래스입니다. `javax.naming.spi.ObjectFactory`를 구현하는 사용자 작성 팩토리 클래스의 정규화된 이름입니다.
설명	사용자 정의 자원에 대한 간단한 설명을 제공합니다.
등록 정보	등록 정보 추가 버튼을 눌러 CLI 등록 정보를 제공합니다(선택 사항).

주 – CLI 사용

CLI를 통해 사용자 정의 자원을 만들려면 다음 명령을 실행합니다.

```
wadm> create-custom-resource --user=admin --password-file=admin.pwd --host=serverhost
--port=8989 --config=config1 --res-type=samples.jndi.customResource.MyBean
--factory-class=samples.jndi.customResource.MyCustomConnectionFactory custom
```

CLI 참조 `create-custom-resource`(1)를 참조하십시오.

외부 JNDI 자원 작업

외부 JNDI 리소스 생성

이 옵션을 사용하여 외부 JNDI(Java Naming and Directory Interface) 자원을 만들 수 있습니다. 내부 JNDI 저장소에 저장된 자원에 액세스하려면 외부 JNDI 자원이 필요합니다.

▼ 외부 JNDI 자원을 추가하는 방법

1 구성을 선택합니다.

구성 목록에서 구성을 선택합니다. 구성 탭을 눌러 목록을 가져옵니다.

2 **Java > 자원 탭을 누릅니다.**

3 **외부 JNDI 섹션 아래에 있는 새로 만들기 버튼을 누릅니다.**

외부 JNDI 자원 등록 정보

다음 표에서는 새 외부 JNDI 자원을 추가하는 동안 사용할 수 있는 등록 정보에 대해 설명합니다.

표 11–4 외부 JNDI 자원 등록 정보

등록 정보	설명
JNDI 이름	새 외부 JNDI 자원에 대해 고유한 이름을 제공합니다.
사용 가능	런타임에 이 외부 JNDI 자원을 사용할지 여부를 결정합니다.
외부 JNDI 이름	외부 JNDI 자원 이름입니다.
자원 유형	이 자원의 정규화된 유형입니다.
팩토리 클래스	이 자원 유형을 인스턴스화하는 클래스입니다.
설명	사용자 정의 자원에 대한 간단한 설명을 제공합니다.
등록 정보	등록 정보 추가 버튼을 눌러 CLI 등록 정보를 제공합니다(선택 사항).

주 – CLI 사용

CLI를 통해 외부 JNDI 자원을 만들려면 다음 명령을 실행합니다.

```
wadm> create-external-jndi-resource --user=admin
--password-file=admin.pwd --host=serverhost --port=8989 --config=config1
--res-type=org.apache.naming.resources.Resource
--factory-class=samples.jndi.externalResource.MyExternalConnectionFactory
--jndilookupname=index.html external-jndi
```

CLI 참조 `create-external-jndi-resource`(1)를 참조하십시오.

메일 자원 구성

JMS 대상은 Sun Java System Web Server를 통해 만들고 관리할 수 있는 Java EE 자원입니다.

많은 인터넷 응용 프로그램에는 전자 메일 알림을 보낼 수 있는 기능이 필요하므로 Java EE 플랫폼에는 응용 프로그램 구성 요소가 인터넷 메일을 보낼 수 있게 하는 JavaMail 서비스 공급자와 함께 JavaMail API가 포함되어 있습니다.

▼ 메일 자원을 추가하는 방법

1 구성을 선택합니다.
구성 목록에서 구성을 선택합니다. 구성 탭을 눌러 목록을 가져옵니다.

2 Java > 자원 탭을 누릅니다.

3 메일 자원 섹션 아래에 있는 새로 만들기 버튼을 누릅니다.

메일 자원 등록 정보

다음 표에서는 새 메일 자원을 추가하는 동안 사용할 수 있는 등록 정보에 대해 설명합니다.

표 11–5 메일 자원 등록 정보

등록 정보	설명
JNDI 이름	새 메일 자원에 대해 고유한 이름을 제공합니다.
사용 가능	런타임에 이 메일 자원을 사용할지 여부를 결정합니다.
사용자	메일 서버에 등록된 유효한 사용자 이름입니다.
보낸 사람	서버가 메일을 보낸 전자 메일 주소입니다.
호스트	메일 서버의 호스트 이름/IP 주소입니다.
저장소 프로토콜	메시지 검색에 사용된 프로토콜입니다.
저장소 프로토콜 클래스	저장소 프로토콜에 대한 저장소 서비스 공급자 구현입니다. 저장소 프로토콜을 구현하는 클래스의 정규화된 클래스 이름입니다. 기본 클래스는 `com.sun.mail.imap.IMAPStore`입니다.
전송 프로토콜	메시지를 보낼 때 사용되는 프로토콜입니다.
전송 프로토콜 클래스	전송 프로토콜에 대한 전송 서비스 공급자 구현입니다. 전송 프로토콜을 구현하는 클래스의 정규화된 클래스 이름입니다. 기본 클래스는 `com.sun.mail.smtp.SMTPTransport`입니다.

주 – CLI 사용

메일 자원을 만들려면 다음 명령을 실행합니다.

```
wadm> create-mail-resource --config=test --server-host=localhost
--mail-user=nobody --from=xyz@foo.com mail/Session
```

CLI 참조 `create-mail-resource`(1)를 참조하십시오.

SOAP 인증 공급자 구성

컨테이너 사양을 위한 Java Authentication Service 공급자 인터페이스는 인증 기법 공급자가 컨테이너와 통합하는 데 사용할 수 있는 표준 서비스 공급자 인터페이스를 정의합니다. 관리 콘솔을 사용하여 새 SOAP 인증 공급자를 추가할 수 있습니다.

▼ SOAP 인증 공급자를 추가하는 방법

1 **구성을 선택합니다.**
구성 목록에서 구성을 선택합니다. 구성 탭을 눌러 목록을 가져옵니다.

2 **Java > 웹 서비스 탭을 누릅니다.**

3 **SOAP 인증 공급자 섹션 아래에 있는 새로 만들기 버튼을 누릅니다.**

SOAP 인증 공급자 매개 변수

다음 표에서는 새 SOAP 인증 공급자 페이지에서 사용할 수 있는 매개 변수에 대해 설명합니다.

표 11–6 SOAP 인증 공급자 매개 변수

매개 변수	설명
이름	새 SOAP 인증 공급자의 약식 이름을 입력합니다.
클래스 이름	공급자를 구현하는 클래스입니다. `javax.security.auth.XXX`를 구현하는 클래스의 정규화된 클래스 이름입니다.

표 11–6 SOAP 인증 공급자 매개 변수 (계속)

매개 변수	설명
인증 소스 요청	이 속성은 사용자 이름/비밀번호와 같은 메시지 계층 보낸 사람 인증, 또는 요청 메시지에 적용될 디지털 서명과 같은 내용 인증에 대한 요구 사항을 정의합니다. 값(auth-policy)은 보낸 사람 또는 내용 중 하나입니다. 이 인수를 지정하지 않으면 요청의 소스 인증이 필요하지 않습니다.
인증 수신자 요청	이 속성은 보낸 사람에 대한 메시지 수신기의 메시지 계층 인증에 대한 요구 사항을 정의합니다(예: XML 암호화). 값은 내용 전 또는 내용 후 중 하나입니다.
인증 소스 응답	이 속성은 사용자 이름/비밀번호와 같은 메시지 계층 보낸 사람 인증, 또는 응답 메시지에 적용될 디지털 서명과 같은 내용 인증에 대한 요구 사항을 정의합니다. 값(auth-policy)은 보낸 사람 또는 내용 중 하나입니다. 이 인수를 지정하지 않으면 응답의 소스 인증이 필요하지 않습니다.
인증 수신자 응답	이 속성은 보낸 사람에 대한 응답 메시지 수신기의 메시지 계층 인증에 대한 요구 사항을 정의합니다(예: XML 암호화).
등록 정보	등록 정보 추가 버튼을 눌러 다른 CLI 등록 정보를 입력합니다.

주 – CLI 사용

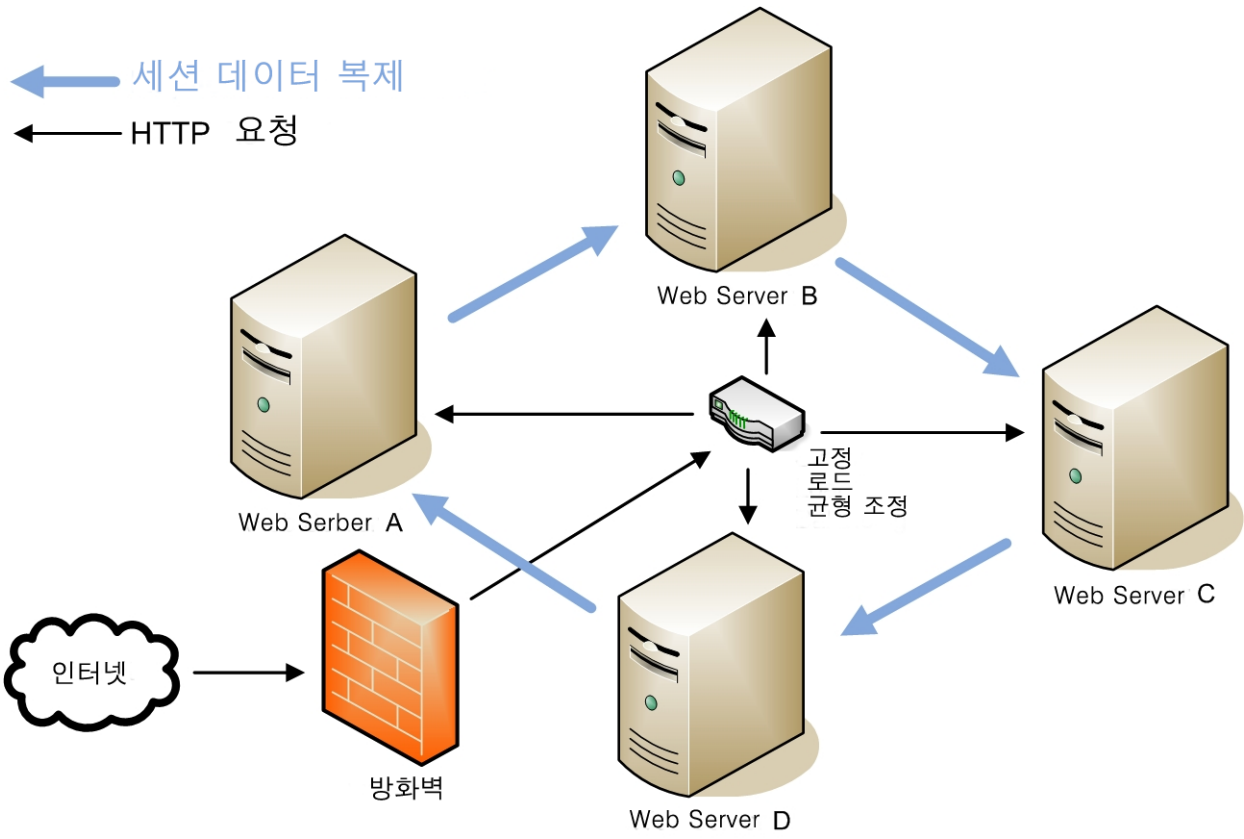
CLI를 사용하여 SOAP 인증 공급자를 추가하려면 다음 명령을 실행합니다.

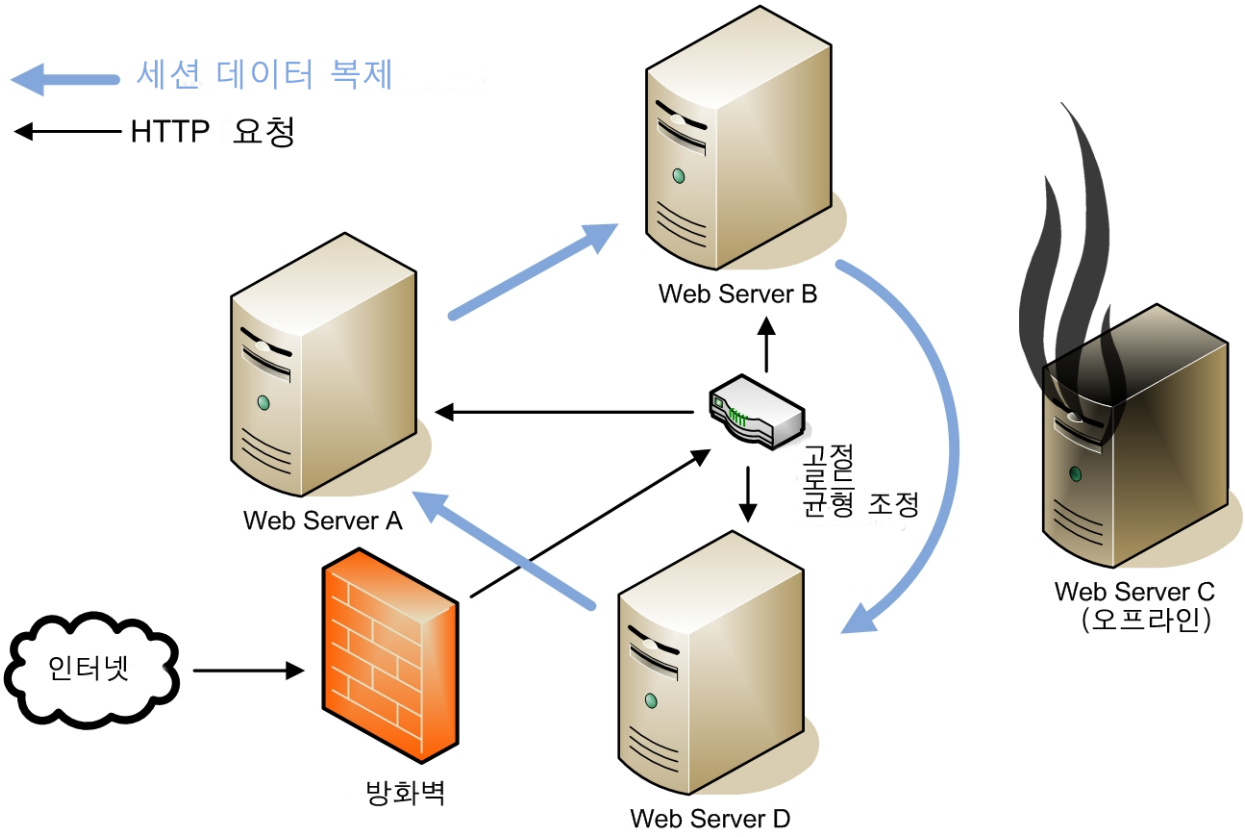
```
wadm> create-soap-auth-provider --user=admin --password-file=admin.pwd
--host=serverhost --port=8989 --config=config1
--class-name=javax.security.auth.soapauthprovider soap-auth
```

CLI 참조 `create-soap-auth-provider`(1)를 참조하십시오.

세션 복제 구성

Sun Java System Web Server 7.0은 웹 응용 프로그램에 고가용성을 제공하는 세션 복제 기능을 지원합니다. 세션 복제는 하나의 인스턴스에서 같은 클러스터의 다른 서버 인스턴스로 HTTP 세션을 복제하여 이러한 목적을 달성합니다. 따라서 각 HTTP 세션에는 원격 인스턴스에 백업 복사본이 있습니다. 클러스터에서 인스턴스 하나를 사용할 수 없는 장애가 발생하는 경우에도 클러스터의 세션 연속성이 계속 유지됩니다.


세션 데이터 복제
HTTP 요청
Web Server B
Web Serber A
고정
로드
균형 조정
Web Server C
인터넷
방화벽
Web Server D



위 그림은 역방향 프록시가 설정된 네 개의 노드 사이에서 세션 복제가 일어나는 전형적인 시나리오를 보여줍니다. Web Server C가 오프라인 상태가 되면 Web Server B에서 Web Server D로 세션 데이터가 복제됩니다.

세션 복제 설정

이 절에서는 선택한 구성에 대해 세션 복제 등록 정보를 설정하는 절차에 대해 설명합니다.

▼ 세션 복제를 설정하는 방법

1. **구성을 선택합니다.**
   구성 목록에서 구성을 선택합니다. 구성 탭을 눌러 목록을 가져옵니다.

2. **Java > 세션 복제를 누릅니다.**

세션 복제 매개 변수 수정

다음 표에서는 세션 복제 페이지에서 사용할 수 있는 매개 변수에 대해 설명합니다.

표 11–7 세션 복제 매개 변수

매개 변수	설명
포트	Administration Server가 수신하고 있는 포트 번호입니다. 기본 포트는 8888입니다.
사용 가능	선택한 구성에 대해 세션 복제를 활성화합니다.
암호화	복제에 앞서 세션 데이터를 암호화할지 여부를 지정합니다. 기본값은 false입니다.
암호	클러스터 구성원이 세션 데이터 복제를 위해 사용하는 암호 제품군(알고리즘, 모드, 채우기)입니다.
Getatrribute 트리거 복제	`HttpSession.getAttribute` 메소드 호출로 세션이 백업되어야 하는지 여부를 지정합니다. 기본값은 true입니다.
복제본 검색 최대 홉 수	세션의 백업 검색을 시도하는 동안 연결해야 할 최대 인스턴스 수입니다. 값의 범위는 1 - 2147.0483647.0이며 제한이 없는 경우에는 -1입니다.
시작 검색 시간 초과	지정된 백업 인스턴스에 연결을 시도할 때 인스턴스가 소요할 최대 시간(초)입니다. 값의 범위는 0.001 - 3600입니다.
쿠키 이름	세션이 있는 인스턴스를 추적하는 쿠키의 이름을 입력합니다.

인증 영역 관리

Java EE 기반 보안 모델은 사용자를 확인하고 인증하는 보안 영역을 제공합니다.

인증 프로세스는 Java 영역을 통하여 사용자를 검증합니다. 영역은 일련의 사용자, 그룹 매핑(선택) 및 인증 요청을 검증할 수 있는 인증 로직으로 구성됩니다. 구성된 영역에서 인증 요청이 검증되고 보안 컨텍스트가 설정되면 이 아이디가 모든 후속 인증 결정에 적용됩니다.

주 – Java 영역은 auth-db(인증 데이터베이스)와 비슷하지만 auth-db는 ACL 엔진(ACL 파일에 있는 규칙 기반)에서 사용되는 반면 Java 영역은 Java 서블릿 액세스 제어 규칙(각 웹 응용 프로그램의 web.xml에서 지정)에서 사용됩니다.

서버 인스턴스의 구성된 영역 수에는 제한이 없습니다. 구성 정보는 `server.xml` 파일의 `auth-realm` 요소에 있습니다.

다음 표에서는 Sun Java System Web Server 7.0에서 지원되는 여러 영역 유형에 대해 정의합니다.

표 11–8 영역 유형

영역	설명
파일	`file` 영역은 Sun Java System Web Server를 처음 설치할 때 기본적으로 설정되는 영역입니다. 이 영역은 설정이 쉽고 간단하여 개발자에게 매우 편리합니다. `file` 영역은 텍스트 파일에 저장된 사용자 데이터를 기준으로 사용자를 인증합니다. Java 영역은 auth-db(인증 데이터베이스)와 비슷하지만 auth-db는 ACL 엔진(ACL 파일에 있는 규칙 기반)에서 사용되는 반면 Java 영역은 Java 서블릿 액세스 제어 규칙(각 웹 응용 프로그램의 web.xml에서 지정)에서 사용됩니다.
LDAP	`ldap` 영역을 통해 사용자 보안 정보용 LDAP 데이터베이스를 사용할 수 있습니다. LDAP 디렉토리 서비스는 고유한 아이디를 가진 속성의 모음입니다. `ldap` 영역은 작업 시스템 배포에 이상적입니다. `ldap` 영역에 대해 사용자를 인증하려면 LDAP 디렉토리에 원하는 사용자를 만들어야 합니다. Administration Server의 사용자 및 그룹 탭에서 이 작업을 실행할 수 있습니다. 이 작업은 LDAP 디렉토리 제품의 사용자 관리 콘솔에서도 수행할 수 있습니다.
PAM	PAM(Solaris) 영역은 인증을 Solaris PAM 스택에 위임합니다. PAM auth-db와 마찬가지로 이 영역은 Solaris 9 및 10에서만 지원되며 서버 인스턴스가 루트로 실행 중이어야 합니다.
인증서	**인증서** 영역은 SSL 인증을 지원합니다. 인증서 영역은 Sun Java System Web Server의 보안 컨텍스트에 사용자 신분을 설정하며 클라이언트 인증서에 있는 사용자 데이터를 입력합니다. 그럼 다음 Java EE 컨테이너는 각 사용자의 인증서에 있는 사용자 DN을 기준으로 인증 프로세스를 처리합니다. 이 영역은 X.509 인증서를 통한 SSL 또는 TLS 클라이언트 인증으로 사용자를 인증합니다.
원시	`native` 영역은 특별 영역으로, 핵심 ACL 기반 인증 모델과 Java EE/서블릿 인증 모델을 연결하는 역할을 합니다. Java 웹 응용 프로그램용 원시 영역을 사용하여 ACL 하위 시스템이 인증을 수행할 수 있도록 할 뿐 아니라 Java 웹 응용 프로그램에서 이를 사용할 수 있도록 합니다. 인증 작업이 시작되면 원시 영역이 이 인증을 해당 핵심 인증 하위 시스템에 위임합니다. 사용자 관점에서 예를 들자면, 이는 구성된 LDAP 서버에 인증을 위임하는 LDAP 영역과 같습니다. 원시 영역이 그룹 구성원 조회를 처리하는 경우에도 핵심 인증 하위 시스템에 이 작업을 위임합니다. Java 웹 모듈 및 개발자의 관점에서 원시 영역은 웹 모듈과 함께 사용할 수 있는 기타 Java 영역과 다르지 않습니다.
사용자 정의	Oracle 등의 기타 데이터베이스용 영역을 구축하면 플러그인 가능한 JAAS 로그인 모듈과 영역 구현을 사용하여 필요에 맞는 영역을 만들 수 있습니다.

다음 절에서는 새 인증 영역 추가와 관련된 단계에 대해 설명합니다.

▼ 인증 영역을 추가하는 방법

1 **구성을 선택합니다.**
새 인증 영역을 추가할 구성을 선택합니다. 구성 탭을 누른 다음 구성을 선택합니다.

2 **Java > 보안 탭을 누릅니다.**

3 **새 인증 버튼을 누릅니다.**

4 **영역 상세 정보를 입력합니다.**

- **이름** — 영역의 약식 이름을 입력합니다. 이 이름은 web.xml 등에서 영역을 참조하는 데 사용됩니다.
- **클래스** — 사용자 정의 영역을 구성하는 경우 사용자 정의 영역을 구현하는 전체 Java 클래스 이름을 입력합니다. 내장 영역에 대해서는 클래스를 입력할 필요가 없습니다.
- **유형** — 영역 유형을 선택합니다. Java 영역 유형에 대해 설명한 이전 절을 참조하십시오.
- **등록 정보** — 영역별 등록 정보를 추가합니다. 예를 들면 `property name="file" value="instance_dir/config/keyfile" and property name="jaas-context" value="fileRealm`과 같습니다.

주 – CLI 사용

CLI를 통해 인증 영역을 추가하려면 다음 명령을 실행합니다.

```
wadm> create-auth-realm --user=admin --password-file=admin.pwd --host=serverhost
--port=8989 --config=config1 basic
```

CLI 참조 `create-auth-realm`(1)을 참조하십시오.

내장 인증 영역 유형의 이름을 지정합니다. 유형은 `file`, `ldap`, `pam`, `native` 또는 `certificate`가 될 수 있습니다.

◆ ◆ ◆ 12장

12 검색 모음 작업

Sun Java System Web Server 7.0에는 사용자가 서버의 문서를 검색하고 웹 페이지에 결과를 표시하도록 하는 검색 기능이 포함되어 있습니다. 서버 관리자는 사용자가 검색할 문서(**모음**이라고 함)의 색인을 만들고 검색 인터페이스를 사용자 정의하여 사용자 요구를 충족시킬 수 있습니다.

검색 모음 쿼리에 대한 자세한 내용은 **검색 온라인 도움말**을 참조하십시오.

검색 정보

검색 기능은 Sun Java System Web Server를 설치하는 동안 다른 웹 구성 요소와 함께 설치됩니다. 검색은 서버 인스턴스 수준이 아닌 가상 서버 수준에서 구성 및 관리됩니다.

관리 콘솔의 가상 서버 탭 아래에 있는 검색 탭에서 다음을 수행할 수 있습니다.

- 검색 기능 활성화/비활성화
- 검색 모음 작성, 수정, 삭제 및 다시 색인화
- 검색 모음의 예약된 유지 관리 작업 작성, 수정 및 제거

관리 인터페이스에서 얻은 정보는 *<server-root>*`/config/server.xml` 파일에 저장되며, 이 파일에서 해당 정보는 VS 요소 내부에 매핑됩니다.

서버 관리자는 검색 쿼리 및 검색 결과 페이지를 사용자 정의할 수 있습니다. 이 작업에는 회사 로고로 페이지를 사용자 정의하거나 검색 결과를 표시하는 방식을 바꾸는 등의 작업이 포함될 수 있습니다. 이전 릴리스에서는 패턴 파일을 사용하여 이 작업을 수행했습니다.

검색에는 전역 "설정" 또는"해제" 기능이 없습니다. 대신 기본 검색 웹 응용 프로그램이 제공된 후 특정 가상 서버에서 활성화 또는 비활성화됩니다. 이 검색 응용 프로그램은 모음을 쿼리하고 결과를 보는 데 사용되는 기본 웹 페이지를 제공합니다. 검색 응용 프로그램에는 검색 태그 라이브러리를 사용하여 사용자 정의 검색 인터페이스를 작성하는 방법을 보여주는 샘플 JSP가 포함되어 있습니다.

주 – Sun Java System Web Server는 검색 결과에 대한 액세스 확인 기능을 제공하지 않습니다. 가능한 보안 모델과 영역의 수가 많기 때문에 검색 응용 프로그램 내에서 보안 확인과 결과 필터링을 수행할 수는 없습니다. 내용을 보호할 수 있는 적절한 보안 기법이 있는지 확인하는 것은 서버 관리자의 책임입니다.

검색 등록 정보 구성

서버에 포함된 검색 응용 프로그램을 활성화하면 가상 서버에서 검색을 사용할 수 있습니다.

주 – 검색을 활성화하려면 Java 웹 컨테이너를 사용 가능으로 설정해야 합니다.

구성하려는 가상 서버에 대해 Java를 사용 가능으로 설정한 후 다음 단계를 수행하여 검색을 활성화하십시오.

1. **구성 탭**을 누릅니다.
2. 구성 목록에서 구성을 선택합니다.
3. **가상 서버 탭**을 누릅니다.
4. 가상 서버 목록에서 가상 서버를 선택합니다.
5. **검색 탭**을 누릅니다.
6. **검색 응용 프로그램 섹션** 아래에서 **사용 가능 확인란**을 눌러 검색 응용 프로그램을 활성화합니다.

구성 가능한 기타 매개 변수는 다음과 같습니다.

- **URI**. 사용자 정의 검색 응용 프로그램을 사용하려면 URI를 입력합니다. 기본 검색 응용 프로그램을 사용하는 경우에는 여기에 값을 지정할 필요가 없습니다.
- **최대 적중 횟수**. 검색 쿼리에서 검색되는 최대 결과 수를 지정합니다.
- **사용 가능**. 이 매개 변수를 선택하여 기본 검색 응용 프로그램을 사용하도록 할 수 있습니다.

주 – CLI 사용

CLI를 통해 검색 등록 정보를 설정하려면 CLI에서 다음 명령을 수행합니다.

```
wadm> set-search-prop --user=admin --password-file=admin.pwd --host=serverhost
--port=8888 --no-ssl --rcfile=null --config=config1 --vs=config1_vs_1
enabled=true max-hits=1200
```

CLI 참조 `set-search-prop`(1)를 참조하십시오.

검색 모음 구성

검색을 수행하려면 사용자가 검색을 수행할 검색 가능한 데이터의 데이터베이스가 필요합니다. 서버 관리자가 이 데이터베이스(모음이라고 함)를 만들며 서버의 문서에 대한 정보를 색인화하고 저장합니다. 서버 관리자가 서버 문서의 전부 또는 일부를 색인화하면 제목, 작성일, 작성자와 같은 정보를 검색할 수 있습니다.

주 – 검색 모음 정보

- 컬렉션은 관리되고 있는 가상 서버에 특정적입니다.
- 가상 서버에 표시되는 문서만 관리 인터페이스에 표시되고 색인화할 수 있습니다.
- 서버에 존재할 수 있는 모음 수에는 제한이 없습니다.
- 검색 모음의 문서는 하나의 문자 인코딩에 종속되지 않습니다. 즉 검색 모음은 여러 인코딩에 관련될 수 있습니다.

지원되는 형식

다음과 같은 형식의 파일을 색인화하고 검색할 수 있습니다.

1. HTML 문서 — .html 및 .htm
2. ASCII 일반 텍스트 — .txt
3. PDF

검색 모음 추가

새 모음을 추가하려면 다음 작업을 수행하십시오.

1. **구성 탭**을 누릅니다.
2. 구성 목록에서 구성을 선택합니다.
3. **가상 서버 탭**을 누릅니다.

4. 가상 서버 목록에서 가상 서버를 선택합니다.
5. **검색 탭**을 누릅니다.
6. **검색 모음 섹션**에서 **검색 모음 추가 버튼**을 눌러 새 검색 모음을 추가합니다.

다음 절에서는 새 검색 모음을 만드는 페이지의 필드에 대해 설명합니다.

1. **검색 모음 정보 제공**
   a. **모음 이름** — 검색 모음에 고유한 이름을 입력합니다.

   주 – 멀티바이트 문자는 모음 이름으로 사용할 수 없습니다.

   b. **디스플레이 이름** — (선택 사항) 검색 쿼리 페이지에서 모음 이름으로 표시됩니다. 표시 이름을 지정하지 않으면 모음 이름이 표시 이름으로 사용됩니다.
   c. **설명** — (선택 사항) 새 모음에 대해 설명하는 텍스트를 입력합니다.
   d. **경로** — 기본 위치에 모음을 만들거나 모음을 저장할 유효한 경로를 입력할 수 있습니다.
2. **색인화 정보 제공**
   a. **색인화할 디렉토리** — 문서를 모음으로 색인화할 디렉토리를 입력합니다. 이 가상 서버에 표시되는 디렉토리만 색인화됩니다.
   b. **하위 디렉토리**— 문서를 모음으로 색인화할 하위 디렉토리를 입력합니다. 하위 디렉토리 경로는 이전에 지정한 디렉토리 경로에 대한 상대 경로여야 합니다.
   c. **패턴** — 색인화할 파일을 선택하기 위해 와일드카드를 지정합니다.

      와일드카드 패턴을 적절하게 사용하여 특정 파일만 색인화되도록 합니다. 예를 들어 *.실행 파일과 Perl 스크립트까지 색인화됩니다.
   d. **하위 디렉토리**— 사용 가능/사용 불가능. 이 옵션을 선택하면 선택한 디렉토리의 하위 디렉토리에 있는 문서도 색인화됩니다. 기본 작업입니다.
   e. **기본 문서 인코딩** —

      모음의 문서는 단일 언어/인코딩에 제한되지 않습니다. 문서를 추가할 때마다 단일 인코딩만 지정되지만 이후 모음에 문서를 추가할 때는 다른 기본 인코딩을 선택할 수 있습니다.
3. **단계 3: 요약 보기**
   a. 요약을 보고 **마침** 버튼을 눌러 새 모음을 추가합니다.

주 – CLI 사용

CLI를 통해 검색 모음을 추가하려면 다음 명령을 실행합니다.

```
wadm> create-search-collection --user=admin --password-file=admin.pwd
--host=serverhost --port=8989 --config=config1 --vs=config1_vs_1 --uri=/search_config1
--document-root=../docs searchcoll
```

CLI 참조 `create-search-collection`(1)을 참조하십시오.

검색 모음 삭제

검색 모음을 삭제하려면 다음 작업을 수행하십시오.

1. **구성 탭**을 누릅니다.
2. 구성 목록에서 구성을 선택합니다.
3. **가상 서버 탭**을 누릅니다.
4. 가상 서버 목록에서 가상 서버를 선택합니다.
5. **검색 탭**을 누릅니다.
6. **검색 모음 섹션** 아래에서 모음 이름을 선택하고 **삭제 버튼**을 눌러 모음을 삭제합니다.

주 – CLI 사용

CLI를 통해 검색 모음을 삭제하려면 다음 명령을 실행합니다.

```
wadm> delete-search-collection --user=admin --password-file=admin.pwd
--host=serverhost --port=8989 --config=config1 --vs=config1_vs_1 searchcoll
```

CLI 참조 `delete-search-collection`(1)을 참조하십시오.

모음 업데이트 예약

모음에서 정기적으로 수행되는 유지 관리 작업을 예약할 수 있습니다. 예약할 수 있는 작업은 다시 색인화 및 업데이트입니다. 관리 인터페이스는 특정 모음에 대한 작업을 예약하는 데 사용됩니다. 다음 작업을 지정할 수 있습니다.

- 실행할 작업(다시 색인화 또는 업데이트)
- 작업을 수행할 시간
- 작업을 수행할 날짜

모음의 이벤트를 예약하려면 다음 작업을 수행하십시오.

1. **구성 탭**을 누릅니다.
2. 구성 목록에서 구성을 선택합니다.
3. **가상 서버 탭**을 누릅니다.
4. 가상 서버 목록에서 가상 서버를 선택합니다.
5. **검색 탭**을 누릅니다.
6. **예약된 이벤트 탭**을 누릅니다.
7. **검색 이벤트 탭**에서 **새로 만들기 버튼**을 누릅니다.

다음 표에서는 새 검색 이벤트 일정 페이지에 있는 필드에 대해 설명합니다.

표 12–1 필드 설명 > 새 검색 이벤트 일정

필드	설명
모음	유지 관리를 예약하려는 모음을 드롭다운 목록에서 선택합니다.
이벤트	1. 모음 다시 색인화 — 이 이벤트를 예약하면 지정한 시간에 지정한 모음을 다시 색인화합니다. 2. 모음 업데이트 — 모음을 만든 다음 파일을 추가 또는 제거할 수 있습니다. 문서는 모음을 만드는 동안 지정된 디렉토리에서만 추가할 수 있습니다. 문서를 제거하는 경우 파일의 항목과 해당 메타데이터만 모음에서 제거됩니다. 실제 파일 자체는 파일 시스템에서 제거되지 않습니다. 이 이벤트를 예약하면 지정한 시간에 모음을 업데이트합니다. 3. 패턴 — 색인화할 파일을 선택하기 위해 와일드카드를 지정합니다. 4. 하위 디렉토리 포함 — 이 옵션을 선택하면 선택한 디렉토리의 하위 디렉토리에 있는 문서도 색인화됩니다. 기본 작업입니다. 5. 인코딩 — 문서를 색인화할 문자 인코딩을 지정합니다. 기본값은 ISO-8859-1입니다. 색인화 엔진은 내장된 메타 태그에서 HTML 문서의 인코딩을 확인합니다. 이것이 지정되지 않으면 기본 부호화가 사용됩니다.
시간	이벤트를 시작하도록 구성된 시간입니다. 드롭다운 상자에서 시간 및 분 값을 선택합니다. **매일** — 지정된 이벤트를 매일 지정된 시간에 시작합니다. **특정 일** — 지정된 이벤트를 특정 날짜에 시작합니다. 1. **요일** — 일요일부터 토요일까지의 요일을 지정합니다. 2. **날짜** — 쉼표로 항목을 구분하여 1일부터 31일까지의 날짜를 지정합니다. (예: 4,23,9). **특정 월** — 지정된 이벤트를 특정 시간 및 월에 시작합니다. 1월부터 12월까지의 월을 지정합니다.
간격	지정된 이벤트를 이 기간 후에 시작합니다. 1. *1* **시간마다** — 드롭다운 상자에서 시간 단위를 선택합니다. 2. *1* **초마다** — 드롭다운 상자에서 초 단위를 선택합니다.

검색 수행

사용자는 주로 검색 모음에서 데이터에 관한 질문을 하고 응답으로 문서 목록을 받습니다. Sun Java System Web Server와 함께 설치된 검색 웹 응용 프로그램은 기본 검색 쿼리 및 검색 결과 페이지를 제공합니다. 이 페이지는 그대로 사용할 수도 있고 *Customizing Search Pages*에 설명된 대로 일련의 JSP 태그를 사용하여 사용자 정의할 수도 있습니다.

사용자는 서버 관리자가 만든 모음을 검색합니다. 다음을 수행할 수 있습니다.

- 검색할 일련의 키워드 및 선택 쿼리 연산자 입력
- 가상 서버에서 볼 수 있는 모음만 검색
- 단일 모음 또는 가상 서버에서 보이는 모음 집합에서 검색

서버 관리자는 사용자에게 가상 서버의 검색 쿼리 페이지 액세스에 필요한 URL을 제공해야 합니다.

검색 페이지

최종 사용자가 검색 기능 액세스를 위해 사용할 수 있는 기본 URL은 다음과 같습니다.

`http://<`*server-instance* `>:`*port number*`/search`

예:

```
http://plaza:8080/search
```

최종 사용자가 이 URL을 호출하면 Java 웹 응용 프로그램인 검색 페이지가 시작됩니다.

주 – 키워드와 선택 쿼리 연산자에 대한 내용을 포함하여 기본 및 고급 검색을 수행하는 데 대한 자세한 내용은 검색 엔진과 함께 제공되는 도움말을 참조하십시오. 이 정보에 액세스하려면 검색 페이지에서 도움말 링크를 누릅니다.

쿼리 만들기

검색 쿼리 페이지는 모음에 대한 검색에 사용됩니다. 사용자가 일련의 키워드와 선택적인 쿼리 연산자를 입력하면 브라우저의 웹 페이지에 결과가 표시됩니다. 결과 페이지에는 검색 기준에 일치하는 서버 문서에 대한 링크가 포함됩니다.

주 – 서버 관리자는 "Customizing Search Pages"에 설명된 대로 이 검색 쿼리 페이지를 사용자 정의할 수 있습니다.

쿼리를 만들려면 다음과 같이 합니다.

▼ 쿼리 만들기

1 **브라우저의 위치 표시줄에 다음 형식으로 URL을 입력하여 검색 웹 응용 프로그램에 액세스합니다.**

`http://<server-instance>:port number /search`

2 **검색 쿼리 페이지가 나타나면 "검색 분야" 필드에서 검색할 모음을 나타내는 확인란을 선택합니다.**

3 **쿼리를 나타내는 단어 몇 개를 입력하고 'Enter' 키를 누르거나 검색 버튼을 눌러 관련 웹 페이지 목록을 표시합니다.**

더 정밀한 검색이 필요한 경우에는 다음 절의 설명과 같이 고급 검색 페이지에 제공되는 검색 매개 변수를 사용합니다.

고급 검색

사용자는 키워드를 미세 조정하는 연산자를 추가하여 검색의 정확도를 높일 수 있습니다. 이 옵션은 고급 검색 페이지에서 선택할 수 있습니다.

고급 검색 쿼리를 만들려면 다음 단계를 수행합니다.

▼ 고급 검색 쿼리를 만드는 방법

1 **브라우저의 위치 표시줄에 다음 형식으로 URL을 입력하여 검색 웹 응용 프로그램에 액세스합니다.**

`http://<server-instance>:port number /search`

2 **고급 링크를 누릅니다.**

3 **다음 정보 중 일부 또는 전부를 입력합니다.**

- 검색 분야. 검색하려는 컬렉션을 선택합니다.
  - 찾을 단어. 4개 옵션이 지원됩니다.
    - 모든 단어 포함. 찾을 단어에 지정된 모든 키워드를 포함하는 페이지를 찾습니다.
    - 아무 단어 포함. 찾을 단어에 지정된 키워드 중 일부를 포함하는 페이지를 찾습니다.
    - 정확한 구문 포함. 찾을 단어에 사용된 문구에 정확하게 일치하는 페이지를 찾습니다.

- 부분 검색. 검색된 페이지에서 키워드 또는 단어를 포함하는 부분을 강조합니다.

다음 단어 없이. 검색에서 지정된 단어가 포함된 웹 페이지를 제외합니다.

- 제목이 다음 단어를 "포함/포함하지 않음". 지정된 키워드를 포함하는 제목을 가진 페이지로 검색을 제한합니다.
- 다음 시기 이후. 선택한 기간에 색인된 웹 페이지로 검색 작업을 제한합니다.

문서 필드

Sun Java™ System Web Server는 문서의 색인을 유지 관리하며 이 색인에는 각 문서에 대한 항목이 포함되어 있습니다. 각 색인 항목에는 제목, 제작자, URL 등의 필드가 하나 이상 포함되어 있습니다. 쿼리를 특정 문서 필드로 제한할 수 있으며 지정된 필드에서 기준과 일치하는 경우에만 문서를 찾습니다.

예를 들어 단순히 Einstein을 찾는다면 제목, 제작자 또는 키워드 필드 중 하나에 Einstein이라는 단어가 포함된 모든 문서를 찾게 됩니다. 그렇게 되면 Einstein에 대한 문서, Einstein을 참고하는 문서 및 Einstein이 쓴 문서도 포함됩니다. 하지만 `Author = "Albert Einstein"`을 지정하면 Albert Einstein이 쓴 문서만 찾게 됩니다.

기본적으로 검색할 수 있는 색인 필드는 다음과 같습니다:

1. **작성자** — `<author>` 메타 태그로 지정된, 문서를 만든 작성자 또는 조직입니다.
2. **키워드** — `<keywords>` 메타 태그로 지정된 키워드입니다.
3. **날짜** — 이 문서를 마지막으로 편집 또는 수정한 날짜입니다.
4. **제목** — HTML `<title>` 태그로 지정된 문서 제목입니다.

검색 쿼리 연산자

검색 쿼리 연산자의 자세한 목록은 **관리 콘솔 검색 온라인 도움말**을 참조하십시오.

검색 결과 보기

검색 결과는 검색 기준과 일치하는 서버의 문서에 대한 HTML 하이퍼링크를 포함하는 웹 페이지로 사용자 브라우저에 표시됩니다. 각 페이지에는 기본적으로 1개의 레코드(적중)가 표시되며, 레코드는 관련도에 따라 내림차순으로 정렬됩니다. 각 레코드는 파일 이름, 크기, 만든 날짜 등의 정보를 나열합니다. 일치하는 단어 역시 강조 표시됩니다.

검색 페이지 사용자 정의

Sun Java System Web Server에는 기본 검색 쿼리 및 검색 결과 페이지를 제공하는 기본 검색 응용 프로그램이 포함되어 있습니다. 이러한 웹 페이지는 그대로 사용할 수도 있고 필요에 따라 사용자 정의할 수도 있습니다. 사용자 정의는 웹 페이지에 다른 로고를 넣는 간단한 작업이 될 수도 있고 검색 결과의 표시 순서를 변경하는 복잡한 작업이 될 수도 있습니다.

기본 검색 응용 프로그램은 검색 태그 라이브러리를 사용하여 사용자 정의 검색 인터페이스를 작성하는 방법을 보여주는 샘플 JSP를 제공합니다. 사용자 정의 검색 태그 사용을 보여주는 샘플 응용 프로그램으로 `/bin/https/webapps/search`에 있는 기본 검색 응용 프로그램을 살펴볼 수 있습니다.

기본 검색 인터페이스는 헤더, 바닥글, 쿼리 양식, 결과의 네 가지 구성 요소로 이루어집니다.

이런 기본 요소는 태그의 속성 값만 변경하여 쉽게 사용자 정의할 수 있습니다. 태그 라이브러리를 사용하면 좀 더 자세한 사용자 정의가 가능합니다.

검색 인터페이스 구성 요소

검색 인터페이스는 다음 구성 요소로 이루어집니다.

Header

헤더에는 로고, 제목 및 짧은 설명이 포함됩니다.

Footer

바닥글에는 저작권 정보가 포함됩니다.

Form

쿼리 양식에는 검색 모음, 쿼리 입력 상자, 제출 및 도움말 버튼을 나타내는 일련의 확인란이 포함되어 있습니다.

Results

결과는 기본적으로 페이지당 10개의 레코드로 표시됩니다. 각 레코드에 대해 제목, 구절, 크기, 만든 날짜, URL 등의 정보가 표시됩니다. 구절은 일치하는 단어가 강조 표시된 페이지의 짧은 부분입니다.

검색 쿼리 페이지 사용자 정의

쿼리 양식에는 검색 모음, 쿼리 입력 상자, 제출 버튼의 확인란 목록이 포함되어 있습니다. 양식은 <s1ws:form> 태그와 <collElem>, <queryBox> 및 <submitButton> 태그를 기본값과 함께 사용하여 만듭니다.

```
<s1ws:form>
    <s1ws:collElem>
    <s1ws:queryBox> <s1ws:submitButton>
</s1ws:form>
```

쿼리 양식은 중간, 세로 막대 등 페이지의 어느 곳에나 배치할 수 있습니다. 또한 모음 선택 상자, 쿼리 문자열 입력 상자, 제출 버튼이 수직으로 나열된 크로스바 또는 모음이 확인란으로 나타나고 쿼리 입력 상자 및 제출 버튼이 그 아래 위치하는 블록과 같은 다른 형식으로도 표시될 수 있습니다.

다음 예는 <searchForm> 태그 집합을 사용하여 다른 형식으로 쿼리 형식을 작성하는 방법을 보여줍니다.

수평 막대

아래의 샘플 코드는 모든 모음의 선택 상자, 쿼리 입력 상자, 제출 버튼이 한 행에 나란히 표시된 양식을 만듭니다.

```
<s1ws:form>
    <table cellspacing="0" cellpadding="3" border="0">
    <tr class="navBar">
        <td class="navBar"><s1ws:collElem type=  select  ></td>
        <td class="navBar">
            <s1ws:querybox size="30">
            <s1ws:submitButton class="navBar" style="padding: 0px; margin: 0px; width: 50px">
        </td>
    </tr>
    </table>
</s1ws:form>
```

세로 막대 블록에서

양식 요소를 세로 막대에 정렬할 수 있으며 세로 막대의 다른 항목과 같은 형식을 사용한 "Search"라는 제목이 있는 양식 블록을 만들 수 있습니다. 이러한 정렬의 효과는 다음 그림에 표시된 것과 같습니다.

세로 막대에 양식 요소가 있는 사용자 정의된 쿼리 페이지

아래의 샘플 코드에서 양식 본문에는 사용 가능한 검색 모음을 나열하며 한 열로 정렬된 세 개의 확인란이 포함됩니다. 아래에는 쿼리 입력 상자와 제출 버튼이 배치됩니다.

```
<s1ws:searchForm>
    <table>
<!--... other sidebar items ... -->
    <tr class="Title"><td>Search</td></tr>
    <tr class="Body">
        <td>
        <table cellspacing="0" cellpadding="3" border="0">
        <tr class="formBlock">
            <td class="formBlock"> <s1ws:collElem type="checkbox" cols="1" values="1,0,1,0" /> </td>
        </tr>
        <tr class="formBlock">
            <td class="formBlock"> <s1ws:querybox size="15" maxlength="50"> </td>
        </tr>
        <tr class="formBlock">
            <td class="formBlock"> <s1ws:submitButton class="navBar" style="padding: 0px; margin: 0px; width: 50px"> <
        </tr>
        </table>
        </td>
    </tr>
    </table>
</s1ws:searchForm>
```

검색 결과 페이지 사용자 정의

검색 결과는 다음과 같이 생성됩니다.

- <formAction> 태그는 모든 양식 요소에서 값을 검색하고 기본적인 확인을 수행합니다.
- <search> 태그, <resultIteration> 태그 및 기타 태그는 <formAction> 태그 내부에 나타나고 모든 양식 요소의 값에 액세스할 수 있습니다.
- <search> 태그는 <formAction>에서 쿼리 문자열 및 모음으로 검색을 실행하고 검색 결과를 pageContext에 저장합니다.
- 그런 다음 <resultIteration> 태그는 검색을 수행하고 결과 집합을 통해 반복합니다.

태그의 속성 값만 변경하면 검색 결과 페이지를 사용자 정의할 수 있습니다.

다음 샘플 코드는 제목 표시줄로 시작한 다음 지정된 수의 레코드를 표시하고 마지막으로 검색 표시줄을 표시합니다. 제목 표시줄에는 검색에 사용되는 쿼리 문자열과 반환되는 총 레코드 범위(예: 1– 10)가 포함됩니다. 각 레코드의 레코드 섹션에는 제목과 파일 링크, 키워드가 강조 표시된 최대 세 개의 구절, 만든 날짜, 문서 크기가 표시됩니다.

섹션의 끝에 있는 검색 표시줄은 이전 및 다음 페이지에 대한 링크와 현재 페이지 앞뒤로 최대 8개의 추가 페이지에 대한 직접 링크를 제공합니다.

```
<s1ws:formAction />
<s1ws:formSubmission success="true" >
    <s1ws:search scope="page" />
    <!--search results-->
    (...html omitted...)
        <s1ws:resultStat formId="test" type="total" /></b> Results Found, Sorted by Relevance</span></td><td>
        <span class="body"><a href="/search/search.jsp?">Sort by Date</a></span></td>
        <td align="right"><span class="body">
        <s1ws:resultNav formId="test" type="previous" caption="<img border=0 src=\\"images/arrow-left.gif\\" alt=
         <s1ws:resultStat formId="test" type="range" />
         <s1ws:resultNav formId="test" type="next" caption="<img border=0 src=\\"images/arrow-right.gif\\" a
          <!img alt="Next" src="images/arrow-right.gif" border="0" WIDTH="13" HEIGHT="9">
            (...html omitted...)
        <table border=0>
        <s1ws:resultIteration formId="test" start="1" results="15">
            <tr class=body>
                <td valign=top>
                <s1ws:item property='number' />.  
                </td>
                <td>
                    <b><a href="<s1ws:item property='url' />"><s1ws:item property='title' /></a></b>
                    <br>
                    <s1ws:item property='passages' />
                    <font color="#999999" size="-2">
                    <s1ws:item property='url' /> -
                    <s1ws:item property='date' /> -
                    <s1ws:item property='size' /> KB
                    </font><br><br>
                </td>
            </tr>
        </s1ws:resultIteration>
        </table>
        (...html omitted...)
        <s1ws:resultNav formId="test" type="previous" />
        <s1ws:resultNav formId="test" type="full" offset="8" />
        <s1ws:resultNav formId="test" type="next" />
    (...html omitted...)
</s1ws:formSubmission>
```

다음 그림은 사용자 정의 검색 결과 페이지를 나타냅니다.

사용자 정의 검색 결과 페이지

태그를 조작하고 HTML을 수정하면 기본 검색 결과 인터페이스를 쉽게 사용자 정의할 수 있습니다. 예를 들어 검색 표시줄을 복사하여 검색 결과 앞에 배치할 수 있습니다. 사용자는 검색 레코드의 등록 정보 표시 여부를 선택할 수도 있습니다.

양식과 함께 사용하는 것 외에 <search>, <resultIterate> 및 관련 태그를 사용하여 특정 항목을 나열할 수도 있습니다. 다음 샘플 코드는 사이트에서 Java Web Services에 관한 상위 10개의 기사를 나열합니다.

```
<s1ws:search collection="Articles" query="Java Web Services" />
<table cellspacing="0" cellpadding="3" border="0">
  <tr class="Title"><td>Java Web Services</td></tr>
</table>
<table cellspacing="0" cellpadding="3" border="0">
<s1ws:resultIteration>
<tr>
<td><a href="<s1ws:item property='URL' />"> <s1ws:item property='Title'/></a></td>
</tr>
</s1ws:resultIteration>
</table>
```

별도 페이지의 양식 및 결과 사용자 정의

양식 및 결과 페이지를 분리하려면 <form> 태그 집합을 사용하여 양식 페이지를 만들고 <formAction> 태그 세트를 사용하여 결과 페이지를 만들어야 합니다.

페이지의 원활한 흐름을 위해서는 결과 페이지에 양식 페이지에 대한 링크를 추가해야 합니다.

태그 규약

다음 태그 규약에 주의하십시오.

- 태그 클래스는 패키지 com.sun.web.search.taglibs에 속합니다.
- 모든 pageContext 속성에는 접두어 com.sun.web이 있습니다. 검색 결과의 속성에 해당되는 예로는 com.sun.web.searchresults.*form_id*가 있습니다. 여기서 *form_id*는 양식의 이름입니다.
- 태그 라이브러리는 접두어 s1ws로 참조됩니다. 태그 이름 및 해당 속성에는 대소문자가 혼용되며 각 내부 단어의 첫 문자가 대문자로 표시됩니다. 예를 들면 pageContext와 같습니다.

태그 사양

Sun Java System Web Server에는 검색 인터페이스에서 검색 쿼리와 검색 결과 페이지를 사용자 정의하는 데 사용할 수 있는 일련의 JSP 태그가 포함되어 있습니다.

검색 페이지를 사용자 정의하는 데 사용할 수 있는 전체 JSP 태그 목록은 Sun Java System Web Server 7.0 *Developer's Guide to Web Applications*를 참조하십시오.

◆ ◆ ◆ 13장

13 서버 모니터링

이 절에서는 Sun Java System Web Server의 모니터링 기능에 대해 설명하고 인스턴스 및 구성 수준 모두에서 모니터링할 수 있는 서버 매개 변수의 자세한 목록을 제공합니다.

Sun Java System Web Server에서 기능 모니터링

모니터링 상위 탭에서 구성 또는 인스턴스 탭을 선택하면 모니터링할 수 있는 서버 매개 변수가 표시됩니다.

Sun Java System Web Server 관리 콘솔에서는 다음 작업을 수행할 수 있습니다.

- 인스턴스 및 구성 수준에서 서버 통계 보기
- 구성 수준에서 모니터링 활성화/비활성화
- 인스턴스 수준에서 오류/액세스 로그 보기

구성 수준에서 서버 매개 변수를 모니터링하려면 모니터링 > 구성 탭을 누릅니다. 테이블에는 사용 가능한 구성과 함께 다음 정보가 나열됩니다.

- **노드** — 구성이 배포된 노드의 수
- **요청** — 모든 가상 서버에서 수신된 총 요청 수
- **오류** — 모든 가상 서버에서 기록된 총 오류 수
- **응답 시간** — 모든 가상 서버의 최대 응답 시간

구성 수준 통계를 보려면 구성 이름을 누릅니다. 일반 통계는 세 유형으로 나눕니다.

- 요청 통계
- 오류 통계

- 응답 시간 통계

관리 콘솔을 통한 모니터링

관리 콘솔에서 다음 범주의 서버 통계를 볼 수 있습니다.

- General Statistics.
- 인스턴스 통계
- 가상 서버 통계

표 13–1 모니터링 범주

범주	설명
일반 통계	일반 통계는 구성의 전체 요청, 오류 및 응답 통계를 표시합니다.
인스턴스 통계	인스턴스 통계는 구성의 전체 요청, 오류 및 응답 통계와 함께 서버 중단 및 가상 서버 수에 관한 정보를 표시합니다.
가상 서버 통계	가상 서버 통계에는 구성의 전체 요청, 오류 및 응답 통계와 함께 열린 연결의 수와 수신/전송한 총 바이트 수가 표시됩니다.

▼ 통계 보기

1 모니터링 탭을 누릅니다.

2 목록에서 구성을 선택합니다.

3 일반, 인스턴스 및 가상 서버 통계를 봅니다.

주 – CLI 사용

`get-config-stats`, `get-virtual-serevr-stats`, `get-webapp-stats` 및 `get-servlet-stats` 명령을 사용하여 서버를 모니터할 수 있습니다.

- `wadm> get-config-stats --user=admin --password-file=admin.passwd --host=localhost --port=8989 --config=test --node=cat.test.com --ssl=true`

  위의 명령은 해당 인스턴스에 대한 통계를 가져옵니다. 구성 수준에서 통계를 가져오려면 위의 명령을 `--node` 옵션 없이 사용할 수 있습니다.

- `wadm> get-vs-stats --user=admin --password-file=admin.passwd --host=localhost --port=8989 --config=test --vs=www.test.com --node=cat.test.com --ssl=true`

  위 명령은 구성이 배포된 모든 노드에서 지정된 구성의 종합 가상 서버 통계를 가져옵니다. 특정 노드에 배포된 구성의 통계를 가져오려면 `--node` 옵션을 사용할 수 있습니다.

- `wadm> get-webapp-stats --user=admin --password-file=admin.passwd --host=localhost --port=8989 --config=test --node=cat.test.com --vs=www.test.com --uri=/foo --ssl=true`

  위의 명령은 지정된 인스턴스의 해당 가상 서버에 배포된 특정 웹 응용 프로그램의 통계를 가져옵니다. 구성이 배포된 모든 노드에서 특정 구성의 종합 웹 응용 프로그램 통계를 가져오려면 `--node` 옵션 없이 위 명령을 사용할 수 있습니다.

- `wadm> get-servlet-stats --user=admin --password-file=admin.pwd --host=localhost --port=8989 --config=test --node=cat.test.com --vs=www.test.com --uri=/servlet-simple --ssl=true`

  위 명령은 서블릿 `servlet-simple`의 통계를 가져옵니다.

모니터링 매개 변수 수정

서버는 SNMP를 통해 모니터링 작업을 수행합니다. SNMP는 네트워크 작동에 대한 데이터를 교환하는 데 사용하는 프로토콜입니다. SNMP를 사용하면 데이터가 관리 대상 장치와 NMS(Network Management Station) 사이를 이동합니다. 관리 대상 장치는 호스트, 라우터, 웹 서버 및 네트워크의 기타 서버 등 SNMP가 실행되는 모든 장치입니다. NMS는 해당 네트워크를 원격으로 관리하는 데 사용되는 시스템입니다. 일반적으로 NMS 소프트웨어는 수집된 데이터를 표시하는 그래프를 제공하거나 해당 데이터를 사용하여 서버가 특정 허용 한계 내에서 작동하는지 확인합니다.

NMS는 일반적으로 하나 이상의 네트워크 관리 응용 프로그램이 설치된 강력한 워크스테이션입니다. HP OpenView와 같은 네트워크 관리 응용 프로그램은 웹 서버 등의 관리 대상 장치에 대한 정보를 그래픽으로 표시합니다. 예를 들어, 회사에서 작동 또는 정지된 서버를 표시하거나 수신된 오류 메시지의 수와 유형을 표시할 수 있습니다. Sun

Java System Web Server에서 SNMP를 사용하는 경우 이 정보는 **하위 에이전트**와 **마스터 에이전트** 등 두 가지 유형의 에이전트를 통해 NMS와 서버 간에 전송됩니다.

하위 에이전트는 서버에 대한 정보를 수집하여 해당 정보를 서버의 마스터 에이전트에 전달합니다. Administration Server를 제외한 모든 Sun Java System Web Server에는 하위 에이전트가 있습니다.

주 – SNMP 구성을 변경한 후에는 저장 버튼을 누른 다음 SNMP 하위 에이전트를 다시 시작해야 합니다.

구성에 대한 설정을 변경하려면 다음 작업을 수행합니다.

1. **구성 탭**을 누릅니다.
2. 모니터링 설정을 변경할 구성을 선택합니다.
3. **모니터링 설정** 하위 탭을 누릅니다.

모니터링 매개 변수 구성

구성에 대한 일반 모니터링 설정을 변경하려면 일반 설정 섹션 아래에 있는 값을 편집합니다. 다음 표에서는 일반 모니터링 매개 변수의 필드에 대해 설명합니다.

표 13–2 필드 설명 > 일반 모니터링 설정

필드	설명
SNMP 하위 에이전트	일반적으로 SNMP를 사용하려면 시스템에 마스터 에이전트 하나와 하위 에이전트를 하나 이상 설치하여 실행해야 합니다. 하위 에이전트를 사용하려면 먼저 마스터 에이전트를 설치해야 합니다. 이 옵션을 선택하여 SNMP 하위 에이전트를 활성화/비활성화합니다.
간격	폴 간격은 표시되는 통계 정보를 업데이트하는 초 단위 간격입니다. 서버 인스턴스가 실행 중이며 통계를 사용하는 경우 선택한 통계의 종류를 표시하는 페이지가 나타납니다. 이 페이지는 폴 간격에서 선택한 값에 따라 5-15초마다 업데이트됩니다.
프로필링	통계/프로필링 기능을 사용하여 서버의 현재 활동을 모니터할 수 있습니다. 통계에는 서버가 처리하는 요청의 수와 해당 요청을 처리하는 상태 등이 표시됩니다. 개별 가상 서버용 일부 통계와 전체 서버 인스턴스에 대한 기타 통계도 확인할 수 있습니다. 이 옵션을 선택하여 프로필링을 활성화/비활성화합니다.

SNMP 하위 에이전트 매개 변수 구성

구성에 대한 SNMP 하위 에이전트 설정을 변경하려면 SNMP 하위 에이전트 설정 섹션 아래에 있는 값을 편집합니다. 다음 표에서는 SNMP 하위 에이전트 매개 변수의 필드에 대해 설명합니다.

표 13–3 필드 설명 > SNMP 하위 에이전트 설정

필드	설명
사용 가능	일반적으로 SNMP를 사용하려면 시스템에 마스터 에이전트 하나와 하위 에이전트를 하나 이상 설치하여 실행해야 합니다. 하위 에이전트를 사용하려면 먼저 마스터 에이전트를 설치해야 합니다. 이 옵션을 선택하여 SNMP 통계 수집을 활성화/비활성화합니다.
마스터 호스트	서버의 이름과 도메인을 입력합니다(*UNIX* **전용**).
설명	운영 체제 정보를 포함하여 서버에 대한 간단한 설명을 입력합니다.
조직	조직을 나타내는 약식 이름을 입력합니다.
위치	이 필드에 서버의 위치 정보를 입력합니다.
연락처	이 필드에 서버의 연락처 정보를 입력합니다.

SNMP 하위 에이전트 구성

SNMP는 네트워크 작동에 대한 데이터를 교환하는 데 사용하는 프로토콜입니다. SNMP를 사용하면 데이터가 관리 대상 장치와 NMS(Network Management Station) 사이를 이동합니다. 관리 대상 장치는 호스트, 라우터, 웹 서버 및 네트워크의 기타 서버 등 SNMP가 실행되는 모든 장치입니다. NMS는 해당 네트워크를 원격으로 관리하는 데 사용되는 시스템입니다. 일반적으로 NMS 소프트웨어는 수집된 데이터를 표시하는 그래프를 제공하거나 해당 데이터를 사용하여 서버가 특정 허용 한계 내에서 작동하는지 확인합니다.

NMS는 일반적으로 하나 이상의 네트워크 관리 응용 프로그램이 설치된 강력한 워크스테이션입니다. Sun Management Center와 같은 네트워크 관리 응용 프로그램에서는 웹 서버 등의 관리 대상 장치에 대한 정보를 그래픽으로 표시합니다. 예를 들어 회사에서 작동 또는 중지된 서버를 표시하거나 수신된 오류 메시지의 수와 유형을 표시할 수 있습니다. Sun Java System Web Server에서 SNMP를 사용하는 경우 이 정보는 **하위 에이전트**와 **마스터 에이전트** 등 두 가지 유형의 에이전트를 통해 NMS와 서버 사이에 전송됩니다.

하위 에이전트는 서버에 대한 정보를 수집하여 해당 정보를 서버의 마스터 에이전트에 전달합니다.

SNMP 하위 에이전트를 시작하려면 다음 작업을 수행하십시오.

1. **노드 탭**을 누릅니다.
2. 노드 목록에서 사용 가능한 노드를 누릅니다.
3. **SNMP 하위 에이전트 탭**을 누릅니다.
4. **SNMP 하위 에이전트 시작 버튼**을 눌러 하위 에이전트를 시작합니다.

주 – SNMP 하위 에이전트를 시작하기 전에 마스터 에이전트가 실행 중인지 확인합니다. 마스터 에이전트가 실행 중인 경우에만 하위 에이전트가 시작됩니다.

SNMP 하위 에이전트를 중지하려면 다음 작업을 수행하십시오.

1. **노드 탭**을 누릅니다.
2. 노드 목록에서 사용 가능한 노드를 누릅니다.
3. **SNMP 하위 에이전트 탭**을 누릅니다.
4. **SNMP 하위 에이전트 중지 버튼**을 눌러 하위 에이전트를 중지합니다.

일반적으로 SNMP를 사용하려면 시스템에 마스터 에이전트 하나와 하위 에이전트를 하나 이상 설치하여 실행해야 합니다. 하위 에이전트를 사용하려면 먼저 마스터 에이전트를 설치해야 합니다.

SNMP를 설정하는 절차는 시스템에 따라 다릅니다. 다음 표에서는 상황에 따른 절차 개요를 제공합니다. 실제 절차는 이 장의 뒤에서 자세히 설명합니다.

시작하기 전에 두 가지를 확인해야 합니다.

- 시스템에 이미 SNMP 에이전트(운영 체제의 원시 에이전트)가 실행 중인지 여부
- 실행 중인 경우 SNMP 에이전트가 SMUX 통신을 지원하는지 여부(AIX 플랫폼을 사용하는 경우 시스템이 SMUX를 지원합니다.)

이 정보를 확인하는 방법은 시스템 설명서를 참조하십시오.

주 – Administration Server에서 SNMP 설정을 변경하거나, 새 서버를 설치하거나, 기존 서버를 삭제한 후에는 다음 단계를 수행해야 합니다.

- (Windows) Windows SNMP 서비스를 재시작하거나 컴퓨터를 재부팅합니다.
- (UNIX) Administration Server를 사용하여 SNMP 마스터 에이전트를 다시 시작합니다.

표 13–4 일반 지침

서버 조건	수행 절차
■ 현재 실행되는 원시 에이전트 없음	1. 마스터 에이전트를 시작합니다. 2. 서버에 설치된 각 서버용 하위 에이전트를 사용하도록 설정합니다.
■ 현재 원시 에이전트 실행 ■ SMUX 없음 ■ 원시 에이전트를 사용하여 계속할 필요 없음	1. Administration Server용 마스터 에이전트를 설치할 때 원시 에이전트를 중지합니다. 2. 마스터 에이전트를 시작합니다. 3. 서버에 설치된 각 서버용 하위 에이전트를 사용하도록 설정합니다.
■ 현재 원시 에이전트 실행 ■ SMUX 없음 ■ 원시 에이전트를 사용하여 계속	1. 프록시 SNMP 에이전트를 설치합니다. 2. 마스터 에이전트를 시작합니다. 3. 해당 프록시 SNMP 에이전트를 시작합니다. 4. 마스터 에이전트 포트 번호가 아닌 다른 포트 번호를 사용하여 원시 에이전트를 다시 시작합니다. 5. 서버에 설치된 각 서버용 하위 에이전트를 사용하도록 설정합니다.
■ 현재 원시 에이전트 실행 ■ SMUX 지원	1. SNMP 원시 에이전트를 재구성합니다. 2. 서버에 설치된 각 서버용 하위 에이전트를 사용하도록 설정합니다.

CLI를 사용하여 SNMP 구성

▼ Solaris에서 SNMP를 활성화하는 방법

1 SNMP 매개 변수를 구성합니다.

구성에 SNMP 매개 변수를 설정합니다.

```
wadm> set-snmp-prop --user=admin --host=funland --port=1893
--config=test enabled=true master-host=masterhost-name organization=organization-name
location=location-name contact=contact-name description=description-name
```

2 구성을 배포합니다.

```
wadm> deploy-config --user=admin --password-file=admin.pwd
--host=serverhost --port=8989 config1
```

3 서버 인스턴스를 시작합니다.

```
$ ./https-test/bin/startserv
```

4 **마스터 에이전트(magt)를 루트로 실행합니다.**

주 – magt를 실행하려면 원시 snmpd를 중지해야 합니다.

```
$ cd /etc/init.d/
          $ init.dmi stop; init.snmpdx stop; init.sma stop
```

https-admserv/config/logs/pid.masteragt 파일을 제거합니다(있는 경우).

```
$ rm ./https-admserv/config/logs/pid.masteragt
          wadm>  start-snmp-master-agent --snmp-port 161 hostname
```

5 **하위 에이전트를 시작합니다.**

https-admserv/config/logs/pid.httpagt 파일을 제거합니다(있는 경우).

```
$ rm ./https-admserv/config/logs/pid.httpagt
```

httpagt가 이미 실행 중인 경우 종료합니다.

```
wadm> start-snmp-subagent hostname
```

▼ Linux에서 SNMP를 활성화하는 방법

1 **SNMP 매개 변수를 구성합니다.**

구성에 SNMP 매개 변수를 설정합니다.

```
wadm> set-snmp-prop --user=admin --host=funland --port=1893 --config=test
enabled=true master-host=masterhost-name organization=organization-name
location=location-name contact=contact-name description=description-name
```

2 **구성을 배포합니다.**

```
wadm deploy-config --user=admin --password-file=admin.pwd
--host=serverhost --port=8989 config1
```

3 **서버 인스턴스를 시작합니다.**

```
$ ./https-test/bin/startserv
```

4 **원시 마스터 에이전트(snmpd)를 루트로 실행합니다.**

snmpd와의 직접 통신을 허용하려면 /etc/snmp/snmpd.conf에 다음 행을 추가하고 snmpd를 다시 시작합니다.

```
smuxpeer 1.3.6.1.4.1.42.2.190.1
```

```
view systemview included .1.3.6.1.4.1.42.2.190.1
# cd /etc/init.d/
      # ./snmpd stop
      # ./snmpd start
```

5 하위 에이전트를 시작합니다.

`https-admserv/config/logs/pid.httpagt` 파일을 제거합니다(있는 경우).

```
$ rm ./https-admserv/config/logs/pid.httpagt
```

`httpagt`가 이미 실행 중인 경우 종료합니다.

```
wadm> start-snmp-subagent hostname
```

▼ Windows에서 SNMP를 활성화하는 방법

1 SNMP 매개 변수를 구성합니다.

구성에 SNMP 매개 변수를 설정합니다.

```
wadm> set-snmp-prop --user=admin --host=funland --port=1893 --config=test
enabled=true master-host=masterhost-name organization=organization-name
location=location-name contact=contact-name description=description-name
```

2 `install-root/lib` 디렉토리를 시스템 경로 환경 변수에 추가합니다.

3 시스템을 다시 시작합니다.

4 Windows 서비스 옵션을 사용하여 Web Server 인스턴스를 시작합니다.

5 SNMP 서비스를 시작합니다.

▼ 피어 기반 마스터 에이전트(magt)를 구성하는 방법

다음 단계를 수행하여 Solaris 10 및 Linux에서 피어 기반 마스터 에이전트가 OS 원시 마스터 에이전트와 통합되도록 구성할 수 있습니다.

주 – Solaris 10 OS 원시 마스터 에이전트는 snmpd입니다. 이 에이전트는 기본적으로 SNMP 기본 UDP 포트 161에서 실행됩니다. 이 값은 /etc/sma/snmp/snmpd.conf 파일을 통해 구성할 수 있습니다. 여기서는 요청/응답을 다른 마스터 에이전트나 하위 에이전트로 전달하는 프록시 지시문을 제공합니다. 자세한 내용은 snmpd.conf 설명서 페이지를 참조하십시오.

Solaris 8 및 9에서는 OS 원시 마스터 에이전트 snmpd와 쉽게 통합할 수 없습니다. Linux에서는 httpagt가 snmpd와 직접 통합할 수 있습니다. 이 경우에는 magt를 실행할 필요가 없습니다. Windows에서는 Sun Java System Web Server snmp 라이브러리가 Windows SNMP 서비스와 직접 통신합니다.

1. **위의 주에서 언급한 것과 같이 SNMP 포트(11161)를 지정하여 마스터 에이전트를 시작합니다.**

2. **Solaris 10의 경우 /etc/sma/snmp/snmpd.conf에 다음을 추가합니다.**

   ```
   proxy -v 1 -c public myserver:11161 .1.3.6.1.4.1.42.2.190.1
   ```

3. **snmpd를 다시 시작합니다.**

   ```
   # cd /etc/init.d
   # init.dmi stop; init.snmpdx stop; init.sma stop
   # init.dmi start; init.snmpdx start; init.sma start
   ```

4. **SNMP 데이터를 가져오려면 포트에서 snmpwalk를 사용합니다.**

   ```
   $ snmpwalk -c public -v 1 <host-name>:<port> 1.3.6.1.4.1.42.2.190.1
   ```

서버에 로깅 설정

Administration Server 로그 파일은 발생한 오류 유형 및 서버 액세스에 대한 정보를 포함하여 서버에 대한 데이터를 기록합니다. 이 로그를 확인하여 서버 작동을 모니터하고 발생한 오류의 유형이나 특정 파일에 액세스한 시간 등의 데이터를 제공함으로써 문제를 해결할 수 있습니다.

관리 콘솔에서 로그 기본 설정 페이지를 사용하면 Administration Server 로그에 기록되는 데이터의 유형과 형식을 지정할 수 있습니다. 예를 들어 Administration Server에 액세스하는 모든 클라이언트에 관한 데이터를 선택하거나 특정 클라이언트를 로그에서 생략할 수 있습니다. 또한 서버에 대해 고정된 양의 정보를 제공하는 공통 로그 형식을 선택할 수도 있고 요구 사항에 맞는 사용자 정의 로그 파일을 만들 수도 있습니다.

로그 유형

로그 유형은 다음과 같이 광범위하게 분류할 수 있습니다.

1. **액세스 로그** — 액세스 로그는 서버로 오고 가는 요청 및 응답에 대한 정보를 기록합니다.
2. **서버 로그** — 서버 로그는 로그 파일을 만든 후에 서버에서 발생한 모든 오류를 나열합니다. 또한 서버가 시작된 시간 및 서버에 로그인을 시도했으나 실패한 사용자 등 서버에 대한 정보 메시지를 포함합니다.

액세스 및 서버 로그 보기

- **서버 로그 보기**.

  ```
  wadm> get-log --user=admin --password-file=admin.passwd --host=localhost
  --port=8989 --start-date=01/01/2006:09:00:00 --end-date=04/01/2006:10:00:00
  --config=test cat.test.com
  ```

  위의 명령은 `01/Jan/2006:09:00:00`와 `04/Jan/2006:10:00:00` 사이에 해당되는 특정 구성의 서버 로그를 표시합니다.

- **액세스 로그 보기**.

  ```
  wadm> get-access-log --user=admin --password-file=admin.passwd
  --host=localhost --port=8989 --status-code=300 --config=test cat.test.com
  ```

  위 명령은 상태 코드가 300인 특정 구성의 액세스 로그 항목만 표시합니다.

위 명령에서 start-date 및 end-date 옵션의 형식은 — `dd/MM/yyyy:HH:mm:ss`와 같아야 합니다. 날짜 형식을 사용자 정의할 수도 있습니다. 기본 날짜 형식을 사용하지 않고 rcfile에 있는 변수 `wadm_log_date_format`을 사용하여 직접 날짜 형식을 지정할 수 있습니다.

로그 매개 변수 구성

구성에 대한 로그 설정을 활성화하고 편집하려면 다음 작업을 수행하십시오.

1. **구성 탭**을 누릅니다.
2. 로그 설정을 활성화/편집할 구성을 선택합니다.
3. **일반 설정 > 로그 설정 탭**을 누릅니다.

액세스 로그 기본 설정 편집

다음 표에서는 액세스 로그 기본 설정 섹션의 필드에 대해 설명합니다.

표 13–5 필드 설명 > 액세스 로그 기본 설정 편집

필드	설명
액세스 로그	**사용 가능/사용 안 함**. 기본적으로 액세스 로그는 활성화되어 있습니다. 액세스 로그를 비활성화하려면 이 옵션을 선택합니다. 액세스 로그를 활성화하면 서버 성능이 약간 저하됩니다.
파일 위치	액세스 로그 파일이 저장되는 서버 경로입니다. 기본값은 *../logs/access*입니다.
로그 형식	1. 공통 로그 형식 사용 — 이 옵션은 로그 파일의 기본 형식 유형입니다. 서버는 요청 헤더에서 추출한 가장 관련성 있는 정보를 기록합니다. 공통 로그 형식은 `IP address – user [date] "request" status content-length`입니다. 2. 다음 세부 정보만 기록합니다. — 이 옵션을 사용하면 요청 헤더에서 특정 값만 기록할 수 있습니다. 다음 값 중에서 선택합니다. ■ 클라이언트 호스트 이름 ■ 시스템 날짜 ■ HTTP 상태 ■ HTTP 헤더 ■ HTTP 메소드 ■ 쿼리 문자열 ■ 가상 서버 이름 ■ 인증된 사용자 이름 ■ HTTP 요청 완료 ■ 내용 길이 ■ 요청 URI ■ 프로토콜

서버 로그 기본 설정 편집

다음 표에서는 서버 로그 기본 설정 섹션의 필드에 대해 설명합니다.

표 13–6 필드 설명 > 서버 로그 기본 설정 편집

필드	설명
서버 로그 위치	서버 로그 파일이 저장되는 서버 경로입니다. 기본값은 *../logs/errors*입니다.

표 13–6 필드 설명 > 서버 로그 기본 설정 편집 *(계속)*

로그 상세 표시 수준	이 옵션은 로그 세분성을 설정하는 효율적인 방법을 제공합니다. 웹 응용 프로그램을 테스트 및 디버깅하는 경우 권장 수준은 **최고**입니다. 작업 환경의 경우 권장 로그 수준은 *failure* 또는 **보안**입니다. **치명적 오류** 로그 수준은 세부 정보를 거의 기록하지 않습니다.
가상 서버 이름 기록	이 옵션을 선택하면 오류와 함께 요청을 처리하는 가상 서버 이름도 기록됩니다.
시스템 로그에 기록	모든 메시지를 시스템 로그에 기록합니다.
콘솔에 기록	이 옵션을 선택하면 배포된 웹 응용 프로그램에서 발생한 예외를 **콘솔**에 기록합니다. 이 옵션은 기본적으로 사용 가능으로 설정됩니다.
날짜 형식	오류 메시지에 타임스탬프를 추가하는 데 사용되는 시간 형식입니다. 기본값은 *[%d/%b/%Y:%H:%M:%S]*입니다.

로그 파일 보관

로그 파일이 자동으로 아카이브되도록 설정할 수 있습니다. 특정 시간 또는 지정된 시간이 경과한 후 서버는 액세스 로그를 회전합니다. 서버는 이전 로그 파일을 저장하고 저장된 파일 이름을 파일이 저장된 날짜 및 시간을 포함하는 이름으로 표시합니다.

예를 들어, 파일을 매 시간 회전하도록 설정한 경우 서버는 파일의 이름을 "`access.199907.0152400`"으로 지정하여 파일을 저장합니다. 여기서 "*name*|*year*|*month*|*day*|*24-hour time*"은 단일 문자열로 연결됩니다. 액세스 로그 아카이브 파일의 정확한 형식은 설정한 로그 회전 유형에 따라 달라집니다.

액세스 로그 회전은 서버 시작 시 초기화됩니다. 회전을 사용하는 경우 서버는 타임스탬프가 지정된 액세스 로그 파일을 만들고 서버가 시작할 때 회전이 시작됩니다.

회전이 시작되면 서버는 액세스 로그 파일에 기록해야 할 요청이 있는 경우 새로운 타임스탬프 액세스 로그 파일을 만들며, 또한 이 작업은 이 작업은 미리 설정된 "다음 회전 시간"이 경과하면 수행됩니다.

로그 회전 설정

로그 회전 옵션을 사용하여 구성된 인스턴스의 오류/액세스 로그 회전에 대한 일정을 만들 수 있습니다. 로그 회전을 설정하려면 다음 단계를 수행하십시오.

1. **구성 탭**을 누릅니다.
2. 로그 설정을 활성화/편집할 구성을 선택합니다.
3. **일반 설정 > 로그 설정 탭**을 누릅니다.
4. **로그 아카이브 섹션** 아래에서 **새로 만들기 버튼**을 누릅니다.

다음 절에서는 새 로그 회전 페이지의 필드에 대해 설명합니다.

표 13–7 필드 설명 > 로그 회전 설정

필드	설명
이벤트	**액세스 로그 회전/서버 로그 회전**. 이 옵션 중 하나 또는 모두를 선택하여 해당 로그 유형에 대한 회전을 구성합니다.
시간	이벤트를 시작하도록 구성된 시간입니다. 드롭다운 상자에서 시간 및 분 값을 선택합니다. **매일** — 지정된 이벤트를 매일 지정된 시간에 시작합니다. **특정 일** — 지정된 이벤트를 특정 날짜에 시작합니다. 1. **요일** — 일요일부터 토요일까지의 요일을 지정합니다. 2. **날짜** — 쉼표로 항목을 구분하여 1일부터 31일까지의 날짜를 지정합니다. (예: 4,23,9). **특정 월** — 지정된 이벤트를 특정 시간 및 월에 시작합니다. 1월부터 12월까지의 월을 지정합니다.
간격	지정된 이벤트를 이 기간 후에 시작합니다. 1. *1***시간마다** — 드롭다운 상자에서 시간 단위를 선택합니다. 2. *1***초마다** — 드롭다운 상자에서 초 단위를 선택합니다.

예약된 로그 회전을 삭제해야 하는 경우 **로그 아카이브** 섹션에서 **삭제 버튼**을 누릅니다.

Administration Server에 대한 로그 설정 구성

관리 콘솔을 사용하여 수행한 모든 구성 변경 사항은 Administration Server에 의해 기록됩니다. 새 구성 만들기, 가상 서버 만들기 및 인스턴스 설정 구성과 같은 일반 작업이 기록됩니다. 하지만 웹 응용 프로그램 액세스 또는 웹 응용 프로그램 액세스 중 발생한 예외와 같은 구성 수준 세부 정보는 구성에 따라 별도로 기록됩니다.

▼ 서버 로그 위치를 수정하는 방법

1 **Administration Server > 일반 탭을 누릅니다.**

2 **로그 기본 설정 섹션으로 이동합니다.**

3 **서버 로그 위치 필드를 편집합니다.**
오류가 저장되는 로그 위치입니다. 로그 파일을 유지 관리하기 위한 유효한 서버 경로를 제공합니다. 또한 UNIX 시스템의 경우 Administration Server가 지정된 디렉토리에 쓰기 권한을 가지는지 확인합니다.

기본 위치는 `../log/errors`입니다.

▼ 로그 상세 표시 수준을 수정하는 방법

1 **Administration Server > 일반 탭을 누릅니다.**

2 **로그 기본 설정 섹션으로 이동합니다.**

3 **로그 상세 표시 수준을 선택합니다.**

이 옵션은 로그 세분성을 설정하는 효율적인 방법을 제공합니다. 테스트 및 디버깅의 경우 권장되는 수준은 **최고**입니다.

작업 환경의 경우 권장 로그 수준은 **실패** 또는 **보안**입니다. **치명적 오류** 로그 수준은 세부 정보를 거의 기록하지 않습니다.

▼ 로그의 날짜 형식을 수정하는 방법

1 **Administration Server > 일반 탭을 누릅니다.**

2 **로그 기본 설정 섹션으로 이동합니다.**

3 **날짜 형식 필드를 편집합니다.**

오류 메시지에 타임스탬프를 추가하는 데 사용되는 시간 형식입니다. 기본값은 *[%d/%b/%Y:%H:%M:%S]*입니다.

◆ ◆ ◆ 14장

14 국제화 및 현지화

국제화 및 현지화 버전의 Sun Java System Web Server는 복수 언어 및 복수 인코딩을 지원합니다.

멀티바이트 데이터 입력

관리 콘솔 페이지에 멀티바이트 데이터를 입력하려면 다음 사항에 유의해야 합니다.

파일 또는 디렉토리 이름

파일이나 디렉토리 이름이 URL에 표시되는 경우 8비트 또는 멀티바이트 문자를 포함하면 안 됩니다.

LDAP 사용자 및 그룹

전자 메일 주소의 경우 RFC 17.000(`ftp://ds.internic.net/rfc/rfc17.000.txt`)에서 허용하는 문자만 사용합니다. 사용자 아이디와 비밀번호 정보는 ASCII로 저장해야 합니다.

사용자와 그룹에 정확한 형식으로 문자를 입력하려면 `UTF-8` 형식을 사용할 수 있는 클라이언트에서 8비트 또는 멀티바이트 데이터를 입력합니다.

복수 문자 인코딩 지원

Sun Java System Web Server 7.0은 다음 기능에 대한 복수 문자 인코딩을 지원합니다.

- 214 페이지 “WebDAV”
- 214 페이지 “검색”

WebDAV

Sun Java System Web Server를 사용하면 `PROPPPATCH` 및 `PROPFIND` 메소드에서 멀티바이트 등록 정보를 설정 및 검색할 수 있습니다. 요청은 임의의 인코딩 형식을 사용할 수 있는 반면 서버로부터의 응답은 항상 UTF-8입니다.

검색

Sun Java System Web Server 7.0은 Java VM 지원에 포함되는 모든 문자 인코딩의 문서를 전체 텍스트 색인화 및 검색할 수 있는 Java 기반 검색 엔진을 사용합니다. 문서의 기본 인코딩은 검색 모음을 만들 때 지정할 수 있습니다. HTML 문서의 경우 인덱서(indexer)는 HTML 메타 태그에서 인코딩을 도출하려고 시도하며, 도출이 불가능한 경우 기본 인코딩으로 돌아갑니다.

검색 인터페이스는 JSP 태그 라이브러리에 기반하며 원하는 언어 및 인코딩으로 사용자 정의 및 현지화할 수 있습니다. 태그 라이브러리는 Sun Java System Web Server 7.0 *Developer's Guide to Web Applications*에 나열되어 있습니다.

서버가 현지화된 컨텐츠를 서비스하도록 구성

최종 사용자는 브라우저가 액세스하는 컨텐트용 언어 선택을 기술하는 Accept-language 헤더를 보내도록 구성할 수 있습니다. **구성** > **(구성 선택)** > **가상 서버** >**(가상 서버 선택)** > **서버 설정** > **일반** > **현지화** 아래에서 **클라이언트 언어 결정** 확인란을 설정하면 Accept-language 헤더를 기반으로 내용을 제공하도록 서버를 구성할 수 있습니다.

예를 들어 이 옵션이 사용 가능으로 설정된 경우 클라이언트가 다음 URL을 요청할 때 값이 `fr-CH,de`인 Accept-language 헤더를 전송합니다.

```
http://www.someplace.com/somepage.html
```

서버는 다음 순서로 파일을 검색합니다.

▼ 검색 순서

1. **Accept-language에서 fr-CH,de를 나열합니다.**

   http://www.someplace.com/fr_ch/somepage.html

   http://www.someplace.com/somepage_fr_ch.html

   http://www.someplace.com/de/somepage.html

   http://www.someplace.com/somepage_de.html

2. **국가 코드가 없는 언어 코드(fr-CH의 경우 fr):**

   http://www.someplace.com/fr/somepage.html

   http://www.someplace.com/somepage_fr.html

3. **magnus.conf 파일에 정의된 en 등의 DefaultLanguage.**

   http://www.someplace.com/en/somepage.html

   http://www.someplace.com/somepage_en.html

4. **이들 중 검색되는 것이 없으면 서버는 다음을 검색합니다.**

   http://www.someplace.com/somepage.html

주 – 현지화된 파일의 이름을 설정하는 경우 CH 및 TW 등의 국가 코드는 소문자로 변환되며 대시(-)는 밑줄(_)로 변환됩니다.

주의 – acceptlanguage 설정을 활성화하면 서버가 위에서 설명한 알고리즘에 따라 Accept-language에 지정된 모든 언어를 확인해야 하기 때문에 성능이 저하됩니다.

◆ ◆ ◆ 부 록 A

이전 버전에서의 CLI 변경 사항

다음 표에서는 Sun Java System Web Server 7.0 이전 버전을 사용하여 수행할 수 있는 일반 작업 몇 가지를 설명합니다.

표 A–1 이전 버전에서의 CLI 변경 사항

작업	6.1 CLI	7.0 CLI
인스턴스에 대해 배포된 웹 응용 프로그램을 모두 나열	`wdeploy list -i INSTANCE_NAME -v VIRTUAL_SERVER`	`wadm> list-webapps --user=admin --port=8888 --password-file=admin.passwd --no-ssl`
새 웹 응용 프로그램 배포	`wdeploy deploy -i INSTANCE_NAME -v VIRTUAL_SERVER -u URI_PATH war file name`	1. `wadm> add-webapp --user=admin --port=8888 --password-file=admin.passwd --config=HOSTNAME --vs=VIRTUAL_SERVER --uri=URI_PATH war file name` 2. `wadm> deploy-config --user=admin --port=8888 --password-file=admin.passwd 'HOSTNAME'`
실행 중인 인스턴스 다시 구성	지원 안 함	`wadm> reconfig-instance --user=admin --port=8888 --password-file=admin.passwd --config=HOSTNAME`
인스턴스에 대해 모든 가상 서버 나열	`HttpServerAdmin list -v -d INSTALL_DIR -sinst https-INSTANCE_NAME`	`wadm> list-virtual-servers --user=admin --port=8888 --password-file=admin.passwd --config=HOSTNAME`

표 A–1 이전 버전에서의 CLI 변경 사항 (계속)

작업	6.1 CLI	7.0 CLI
모든 JDBC 자원 나열	`HttpServerAdmin list -r -jdbc -d INSTALL_DIR -sintance https-INSTANCE_NAME`	`wadm> list-jdbc-resources --user=admin --port=8888 --password-file=admin.passwd --config=HOSTNAME`
사용자 정의 자원 만들기	`HttpServerAdmin create -r -custom -jndiname -poolname -enabled true`	`wadm> create-custom-resource --user=admin --port=8888 --password-file=admin.passwd --config=HOSTNAME --res-type=type --jndi-name NAME`
인스턴스 시작	지원 안 함	`wadm> start-instance --user=admin --port=8888 --password-file=admin.passwd --config=HOSTNAME NODENAME*`
인스턴스 중지	지원 안 함	`wadm> stop-instance --user=admin --port=8888 --password-file=admin.passwd --config=HOSTNAME NODENAME*`
웹 서버에 역방향 프록시 구성	지원 안 함	1. `wadm> create-reverse-proxy --user=admin --port=8888 --password-file=admin.passwd --config=HOSTNAME --vs='VIRTUAL_SERVER' --from='URI' --server='target-hostname'` 2. `wadm> set-reverse-proxy-prop --user=admin --password-file=admin.pwd --host=serverhost --port=8888 --config=config1 --vs=config1_vs_1 --uri-prefix=/test/ --server=http://java.com:8080 --sticky-cookie=testCookie`
역방향 프록시 비활성화	지원 안 함	`wadm> delete-reverse-proxy --user=admin --port=8888 --password-file=admin.passwd --config=HOSTNAME --vs='VIRTUAL_SERVER' --uri-prefix='URI'`

표 A–1 이전 버전에서의 CLI 변경 사항 *(계속)*

작업	6.1 CLI	7.0 CLI
WebDAV 활성화	지원 안 함	1. `wadm> enable-webdav --user=admin --port=8888 --password-file=admin.passwd --config=HOSTNAME` 2. `wadm> deploy-config --user=admin --port=8888 --password-file=admin.passwd HOSTNAME`
새 웹 서버 만들기	지원 안 함	1. `wadm> create-config --doc-root=[DOCROOT] -http-port=[HTTPPORT] --jdk-home=[JAVAHOME] --server-user=[SERVERUSER] --server-name=[HOSTNAME] CONFIGNAME` 2. `wadm> create-instance --config=CONFIGNAME NODENAME` 3. `wadm> deploy-config CONFIGNAME`

◆ ◆ ◆ 부 록 B

FastCGI 플러그인

개요

FastCGI는 외부 응용 프로그램과 Web Server 간의 표준 인터페이스로 사용되는 기존 CGI(Common Gateway Interface)를 향상시킨 것입니다. CGI와 마찬가지로 FastCGI 응용 프로그램은 별도의 격리된 프로세스에서 실행됩니다. FastCGI를 사용하는 경우의 장점은 다음과 같습니다.

- 응용 프로그램이 클라이언트 요청 사이에 지속될 수 있고, 응용 프로그램 시작 오버헤드가 해소되며, 응용 프로그램에서 클라이언트 호출 사이에 상태를 유지할 수 있습니다.
- 응용 프로그램이 원격 시스템(Web Server가 실행되는 시스템과 다른 시스템)에 상주할 수 있습니다.
- 클라이언트 인증과 입력 필터링을 수행하는 응용 프로그램을 명시적으로 지원하여 응용 프로그램 기능에 유연성을 추가할 수 있습니다.
- 관리자가 FastCGI 서버로 인한 시스템의 영향을 제한할 수 있습니다.

FastCGI 플러그인을 사용하면 Web Server에서 안전하게, 확장 가능한 방식으로 흔히 사용되는 타사의 동적 컨텐트 생성 기술(Perl 및 Python 등)로 작업을 수행할 수 있습니다.

FastCGI에 대한 자세한 내용은 http://www.fastcgi.com/devkit/doc/fcgi-spec.html의 사양을 참조하십시오.

플러그인 기능(SAF)

FastCGI 플러그인은 다음과 같은 서버 응용 프로그램 기능(SAF)을 제공합니다.

FastCGI SAF의 다양한 매개 변수와 "error-reason" 문자열에 대해서는 다음 절에서 설명합니다.

auth-fastcgi

`auth-fastcgi`는 PatchCheck 기능입니다. 이 기능은 요청을 "인증자" FastCGI 응용 프로그램으로 전달하는 데 사용됩니다. 인증에 성공하면 반환 코드 200이 전송됩니다. 그렇지 않은 경우 "인증자" FastCGI 응용 프로그램의 응답이 사용자 에이전트로 돌아갑니다.

FastCGI 역할에 대한 자세한 내용은 http://www.fastcgi.com/devkit/doc/fcgi-spec.html#S6을 참조하십시오.

`auth-fastcgi` SAF에서 허용되는 매개 변수는 다음을 참조하십시오. 223 페이지 “FastCGI SAF 매개 변수”.

다음 `obj.conf` 코드에서는 `auth-fastcgi`의 사용 예를 보여줍니다.

```
PathCheck fn="auth-fastcgi" app-path="/usr/bin/perl"
app-args="/fastcgi/apps/auth/SimpleAuth.pl" bind-path="localhost:3432".
```

responder-fastcgi

`responder-fastcgi`는 서비스 함수입니다. 이 함수는 "응답기"로 사용되는 FastCGI 응용 프로그램으로 요청을 전달하는 데 사용됩니다. 응답기 응용 프로그램의 응답은 사용자 에이전트로 돌아갑니다. FastCGI 역할에 대한 자세한 내용은 http://www.fastcgi.com/devkit/doc/fcgi-spec.html#S6을 참조하십시오.

`responder-fastcgi` SAF에서 허용되는 매개 변수의 목록은 다음을 참조하십시오. 223 페이지 “FastCGI SAF 매개 변수” .

다음 `obj.conf` 코드에서는 `responder-fastcgi`의 사용 예를 보여줍니다.

```
Service fn="responder-fastcgi"
app-path="/fastcgi-enabled-php-installation/bin/php"
bind-path="localhost:3433" app-env="PHP_FCGI_CHILDREN=8"
app-env="PHP_FCGI_MAX_REQUEST=500".
```

filter-fastcgi

`filter-fastcgi`는 서비스 함수입니다. 이 기능은 요청을 "필터" 유형의 FastCGI 응용 프로그램으로 전달하는 데 사용됩니다. "필터" 응용 프로그램은 HTTP 요청과 연결된 정보를 받으며 서버에 저장된 파일의 데이터도 받습니다. 그런 다음 "필터" 응용 프로그램은 응답으로 "필터링된" 버전의 데이터 스트림을 생성합니다. 이 스트림은 사용자 에이전트로 다시 전송됩니다. FastCGI 역할에 대한 자세한 내용은 http://www.fastcgi.com/devkit/doc/fcgi-spec.html#S6을 참조하십시오.

`filter-fastcgi` SAF에서 허용되는 매개 변수의 목록은 다음을 참조하십시오. 223 페이지 “FastCGI SAF 매개 변수” .

다음 `obj.conf` 코드에서는 `filter-fastcgi`의 사용 예를 보여줍니다.

```
Service fn="filter-fastcgi" app-path="/fastcgi/apps/filter/SimpleFilter"
bind-path="localhost:3434"
app-env="LD_LIBRARY_PATH=/fastcgi/fcgi-2.4/libfcgi/.libs" min-procs=2
```

error-fastcgi

`error-fastcgi`는 오류 함수입니다. `error-fastcgi` SAF는 FastCGI 플러그인의 오류를 처리합니다. 그러나 이 함수는 HTTP 오류를 처리하지 않습니다. 오류가 발생한 경우에는 특정 페이지를 표시하거나 요청을 특정 URL로 리디렉션하도록 FastCGI 플러그인을 구성할 수 있습니다.

`error-fastcgi` SAF에서 허용되는 매개 변수의 목록은 다음을 참조하십시오. 223 페이지 “FastCGI SAF 매개 변수”.

다음 `obj.conf` 코드는 `error-fastcgi`의 사용 예를 보여줍니다.

```
Error fn="error-fastcgi" error-reason="Invalid Parameters"
error-url="http://www.foo.com/errorPage.html"
```

`error-fastcgi` 매개 변수에 대한 자세한 내용은 223 페이지 “FastCGI SAF 매개 변수”를 참조하십시오.

FastCGI SAF 매개 변수

SAF의 FastCGI 플러그인인 "`auth-fastcgi`", " `responder-fastcgi`" 및 "`filter-fastcgi`"는 따로 명시되지 않은 한 다음 매개 변수를 모두 허용합니다.

매개 변수 chroot, user, group 및 nice는 UNIX 플랫폼에서만 사용할 수 있습니다. Windows 플랫폼에서는 이런 매개 변수가 무시됩니다.

- app-path - (선택 사항)요청을 처리하는 FastCGI 응용 프로그램 경로입니다. 이 기능은 다음과 같이 bind-path 매개 변수 값에 따라 결정됩니다.
  1. 플러그인은 app-path가 지정된 경우에만 플러그인에서 만든 UNIX 도메인 소켓을 수신하는 FastCGI 응용 프로그램을 만듭니다. 하지만 이 매개 변수는 UNIX 플랫폼에서만 허용됩니다. Windows에서는 오류 메시지가 기록됩니다.
  2. app-path와 bind-path가 모두 지정된 경우 플러그인은 지정된 FastCGI 응용 프로그램 프로세스를 시작하고 지정된 bind-path에 바인드합니다.
  3. bind-path만 지정된 경우에는 FastCGI 응용 프로그램이 원격에서 실행되는 것으로 간주됩니다. 따라서 플러그인은 FastCGI 응용 프로그램 프로세스를 시작하지 않습니다.
  4. "app-path"와 "bind-path"가 모두 지정되지 않은 경우에는 플러그인에서 오류 메시지가 기록됩니다.
- app-args — (선택 사항) FastCGI 응용 프로그램 프로세스에 인수로 전달되는 값입니다. 여러 개의 app-args 매개 변수가 허용됩니다. 여러 app-args 매개 변수를 사용할 때의 형식은 app-args="value" app-args="value" . . 입니다.
- bind-path - (선택 사항) Unix 도메인 소켓 이름이거나 "host:port" 형식일 수 있습니다. "app-path" 매개 변수 설명에서는 "bind-path" 매개 변수의 사용에 대해 설명합니다. Unix 도메인 소켓 이름은 UNIX 플랫폼에서만 사용할 수 있습니다. Windows 플랫폼에서는 bind-path를 "host:port"로 지정해야 합니다.
- min-procs - (선택 사항) 만들어야 할 FastCGI 응용 프로그램 프로세스의 최소 수를 지정하는 정수입니다. 기본값은 1입니다.
- max-procs - (선택 사항) 임의의 시점에 만들 수 있는 FastCGI 응용 프로그램 프로세스의 최대 수를 지정하는 정수입니다. 정수 값은 min-procs 값 이상이어야 합니다. 기본값은 1입니다.
- chroot - (선택 사항) chroot FastCGI 서버 응용 프로그램 프로세스의 루트 디렉토리를 설정하는 데 사용됩니다. 기본값은 Web Server의 루트 디렉토리입니다.
- user - (선택 사항) FastCGI 응용 프로그램 실행에 사용되는 사용자 아이디를 지정합니다. 기본값은 Web Server의 사용자 아이디입니다.
- group - (선택 사항) 지정된 FastCGI 응용 프로그램은 지정된 그룹 아래에서 실행됩니다. 기본값은 Web Server의 그룹입니다.
- nice - (선택 사항) FastCGI 응용 프로그램 프로세스의 nice/priority 값을 지정합니다.
- listen-queue - (선택 사항) 소켓의 수신 대기열 크기를 지정하는 정수입니다. 이 매개 변수의 기본값은 256입니다.
- app-env - (선택 사항) FastCGI 응용 프로그램 프로세스에 환경 변수로 전달되는 값 쌍입니다. 여러 개의 " app-env" 매개 변수가 허용됩니다. 여러 app-env 매개 변수를 사용할 때의 형식은 app-env="name=value" app-env="name=value". . . . 입니다.

- `reuse-connection` - (선택 사항) FastCGI 응용 프로그램에 대한 연결을 다시 사용할 것인지 여부를 결정하는 부울 값입니다. `False(0, false, no)`는 각 요청 뒤에 FastCGI 응용 프로그램에 대한 연결을 닫는 것을 나타냅니다. `True(1, true, yes)`는 새 요청에 기존 연결을 다시 사용하는 것을 나타냅니다. 기본값은 `false`입니다. `connection-timeout`을 참조하십시오.
- `connection-timeout` - (선택 사항) "`reuse-connection`"이 `True`로 설정되어 있으면 이 값은 풀링된 연결의 시간 초과 값을 초 단위로 지정합니다. 연결이 지정된 시간 동안 유휴 상태이면 플러그인에서 연결을 닫습니다. 이 매개 변수의 기본값은 5초입니다. `reuse-connection`을 참조하십시오.
- `resp-timeout` - (선택 사항) FastCGI 서버 응답 시간 초과 값을 초 단위로 나타내는 정수입니다. FastCGI 응용 프로그램에서 지정된 시간 내에 응답이 없는 경우에는 요청이 무시됩니다. 이 매개 변수의 기본값은 5분입니다.
- `restart-interval` - (선택 사항) FastCGI 응용 프로그램이 다시 시작된 후 시간 간격(분)을 나타내는 정수입니다. 이 매개 변수의 기본값은 60분(1시간)입니다. 이 매개 변수 값을 0으로 설정하면 FastCGI 응용 프로그램을 강제로 다시 시작하지 않습니다.
- `req-retry` - (선택 사항) FastCGI 응용 프로그램에서 요청을 거부한 경우에 플러그인이 요청을 다시 보내는 횟수를 나타내는 정수입니다. 이 매개 변수의 기본값은 0입니다.

`error-fastcgi` SAF(Server Application Function)에는 다음과 같은 매개 변수를 사용할 수 있습니다.

- `error-url` - 장애나 오류가 발생한 경우에 표시할 페이지, URI 또는 URL을 지정합니다. 이 매개 변수 값에는 절대 경로, docroot를 기준으로 한 상대 경로 또는 URL이나 URI를 사용할 수 있습니다.
- `error-reason` - (선택 사항) FastCGI 프로토콜 오류를 나타내는 문자열입니다. 이 문자열은 플러그인 오류가 발생한 경우에 표시할 오류 URL을 구분하는 데 사용됩니다.

error-fastcgi SAF 오류 원인 문자열

이 절에서는 모든 유효한 "error-reason" 문자열의 목록과 해당 설명을 제시합니다.

- "Missing or Invalid Config Parameters" : app-path 및 bind-path가 지정되지 않았습니다.
- "Stub Start Error" : Fastcgisub 프로세스를 시작하지 못했습니다.
- "Stub Connection Failure" : Fastcgistub에 연결할 수 없습니다.
- "No Permission" : FastCGI 응용 프로그램 또는 Fastcgisub에 실행 권한이 없습니다.
- "Stub Request Handling Error" : 요청을 스텁으로 보낼 수 없거나, 요청의 스텁에서 잘못된 응답을 받았거나, 응답이 없는 경우 등을 나타냅니다.
- "Set Parameter Failure" : 설정된 사용자, 그룹, chroot, nice 등이 실패했습니다.

- "Invalid user and/or group" : 사용자 또는 그룹이 유효하지 않습니다.
- "Server Process Creation Failure" : FastCGI 응용 프로그램 실행 장애가 발생했거나 FastCGI 응용 프로그램을 지정된 주소에 바인드할 수 없습니다.
- "Fastcgi Protocol Error" : FastCGI 응용 프로그램에 FastCGI 버전 또는 역할이 잘못된 헤더가 포함되어 있습니다.
- "Internal Error" : 필터 응용 프로그램으로 보낼 파일을 열 수 없거나 기타 알 수 없는 오류가 발생했습니다.

Web Server에 FastCGI 플러그인 구성

FastCGI 플러그인은 Web Server 7.0에 번들로 제공됩니다. 플러그인은 다음 위치에 설치됩니다.

32비트 FastCGI 플러그인 바이너리는 <*install_dir*>`/plugins/fastcgi` 디렉토리에 설치됩니다.
64 비트 Solaris SPARC FastCGI 플러그인 바이너리는 <*install_dir*>`/lib/plugins/fastcgi/64` 디렉토리에 설치됩니다.

다음 FastCGI 바이너리가 설치됩니다.

`libfastcgi.so`(Solaris/Linux)
`fastcgi.dll`(Windows)
`Fastcgistub.exe`(Windows)
`libfastcgi.sl`(HP-UX)
`Fastcgistub`(실행 파일)

FastCGI 플러그인은 `<instance-dir>/config` 디렉토리 아래에 있는 Web Server 구성을 통해 구성됩니다. FastCGI 플러그인을 구성하려면 다음 단계를 수행합니다.

- 226 페이지 "magnus.conf 수정"
- 227 페이지 "MIME 유형 수정(선택 사항)"
- 227 페이지 "obj.conf 수정"
- 229 페이지 "FastCGI 플러그인 문제 해결"
- 230 페이지 "FastCGI 응용 프로그램 개발"

magnus.conf 수정

"load-modules" Init 기능을 사용하여 FastCGI 플러그인 공유 라이브러리를 로드합니다.

```
Init fn=flex-init access="access" format.access="%Ses->client.ip%
- %Req->vars.auth-user% [%SYSDATE%] \"%Req->reqpb.clf-request%\"
%Req->srvhdrs.clf-status% %Req->srvhdrs.content-length%"
```

```
Init fn="load-modules" shlib="libJava EEplugin.so" shlib_flags="(global|now)"

Init fn="load-modules" shlib="libfastcgi.so" shlib_flags="(global|now)"
```

MIME 유형 수정(선택 사항)

mime.types 파일을 편집하여 MIME 매핑을 지정합니다. MIME 유형 매핑 수정은 선택적인 단계입니다.

예:

```
#--Sun Microsystems Inc. MIME Information

# Do not delete the above line. It is used to identify the file type.

#

# Copyright 2006 Sun Microsystems, Inc. All rights reserved.

# Use is subject to license terms.

#


type=application/octet-stream exts=bin

type=application/astound exts=asd,asn

...

...

type=magnus-internal/fastcgi exts=php

...

...
```

obj.conf 수정

obj.conf 파일을 편집하여 이전 절에 설명된 플러그인 SAF를 통해 FastCGI 특정 요청을 구성합니다.

다음은 수정된 obj.conf 파일의 예입니다.

```
#

# Copyright 2006 Sun Microsystems, Inc. All rights reserved.

# Use is subject to license terms.

#


# You can edit this file, but comments and formatting changes

# might be lost when you use the administration GUI or CLI.



<object name = "default">

        AuthTrans fn="match-browser" browser="*MSIE*"
                        ssl-unclean-shutdown="true"
        NameTrans fn="ntrans-Java EE" name="Java EE"
        NameTrans fn="pfx2dir" from="/mc-icons"
                        dir="/ws7/lib/icons" name="es-internal"
        NameTrans fn="assign-name" from="/fcgi/*" name="fcgi.config"

</object>

<Object name="fcgi.config">

        AuthTrans fn="auth-fastcgi" app-path="/fastcgi/apps/c/simpleAuth"
                bind-path="localhost:2111"
        Service fn="responder-fastcgi"
                        app-path="/fastcgi_enabled_php_installation_dir/bin/php"
                app-env="name1=abc"

</object>
...
```

서로 다른 URL 패턴에 다른 개체를 정의하거나 SAF를 서로 다른 MIME 유형에 매핑하면 다양한 방법으로 FastCGI SAF를 호출할 수 있습니다.

obj.conf 구성 및 구문에 대한 자세한 내용은 *Administration Configuration File Reference Guide*를 참조하십시오.

FastCGI 플러그인 문제 해결

Fastcgistub은 FastCGI 응용 프로그램 프로세스의 라이프사이클을 관리하는 프로세스 관리자입니다. Fastcgistub은 메시지를 Web Server의 임시 디렉토리 아래의 `Fastcgistub.log` 파일에 기록합니다. 오류가 발생한 경우에는 이 파일을 확인하면 문제를 디버깅하는 데 도움이 됩니다.

문제: FastCGI 요청이 처리되지 않습니다.

가능한 원인과 해결 방법은 다음과 같습니다.

1. FastCGI 플러그인이 로드되었는지 확인합니다. Web Server를 시작할 때 다음 메시지가 나타나면 플러그인이 로드된 것입니다. 그렇지 않은 경우에는 `magnus.conf`에서 플러그인 라이브러리의 경로를 확인합니다. `FCGI1000: Sun Java System Web Server 7.0 FastCGI NSAPI Plugin < build info>`
2. `obj.conf`에서 요청 매핑이 정확하게 지정되었는지 확인합니다. `obj.conf` 파일에 대한 자세한 내용은 Sun Java System Web Server *Administrator's Configuration Reference File*을 참조하십시오.
3. 오류 로그에 오류 메시지가 있는지 확인합니다.
4. 스텁 바이너리와 FastCGI 응용 프로그램의 권한을 확인합니다. 충분한 권한이 지정되지 않은 경우에는 플러그인에서 스텁 또는 응용 프로그램 시작에 실패합니다.
5. `Fastcgistub.log` 파일에서 스텁 부분의 오류가 있는지 확인합니다.
6. 가능한 경우 FastCGI 응용 프로그램을 독립 실행형 모드로 실행하고 문제 없이 실행되는지 확인합니다.

라이브러리 종속성 오류가 발생한 경우에는 `obj.conf`에서 `LD_LIBRARY_PATH`를 `LD_LIBRARY_PATH=<dependency library paths>`인 `app-env` 매개 변수로 지정합니다.

문제: FastCGI 응용 프로그램이 시작되지 않습니다.

가능한 원인과 해결 방법은 다음과 같습니다.

`Fastcgistub.log` 파일에 다음 로그 메시지가 있는지 확인합니다.

```
..
<pid> process startup failure, trying to restart
...
Even after trying <n> time(s), <application path> process failed to start...no more retries
```

시작 장애의 원인 중에는 종속성 라이브러리 로드 실패가 있을 수 있습니다. 이 문제는 obj.conf 파일에 구성된 FastCGI 응용 프로그램에 `app-env` 매개 변수 값으로 적절한 라이브러리 경로를 지정하면 해결할 수 있습니다. 예:

```
Service fn="responder_fastcgi" app-path="/fastcgi/c/tux-app" bind-path="localhost:2112"
app-env="LD_LIBRARY_PATH=/tuxedo/lib"
```

FastCGI 응용 프로그램 개발

FastCGI 응용 프로그램은 Perl, PHP, C 및 Java를 사용하여 개발할 수 있습니다. 다음 절에서는 흔히 사용되는 일부 프로그래밍 언어를 통해 응용 프로그램을 개발하는 절차를 간략하게 설명합니다.

▼ FastCGI 응용 프로그램 실행

1. **Web Server를 중지합니다.**
2. **Web Server를 다시 시작합니다.**
3. **응용 프로그램 루트가 "fcgi"인 응용 프로그램에 액세스합니다.**

   예: `http://localhost/fcgi/ListDir.php`

FastCGI 응용 프로그램의 구조

전형적인 FastCGI 응용 프로그램의 코드 구조는 다음과 같습니다.

```
Initialization code

Start of response loop
        body of response loop
End of response loop
```

초기화 코드는 응용 프로그램을 초기화할 때 한 번만 실행됩니다. 초기화 코드는 보통 데이터베이스 열기 또는 테이블이나 비트맵의 값 계산과 같이 시간이 오래 걸리는 작업을 수행합니다. CGI 프로그램을 FastCGI 프로그램으로 변환할 때의 주된 작업은 초기화 코드를 각 요청에 대해 실행해야 할 코드와 분리하는 것입니다.

응답 루프는 계속 실행되며 클라이언트 요청이 도착하기를 기다립니다. 루프는 FastCGI 라이브러리에 있는 루틴인 `FCGI_Accept`에 대한 호출로 시작됩니다. `FCGI_Accept` 루틴은 클라이언트에서 FastCGI 응용 프로그램을 요청할 때까지 프로그램 실행을 차단합니다. 클라이언트 요청이 들어오면 `FCGI_Accept` 차단이 해제되고 응답 루프 본문이 한 번 실행된 후 다시 차단되어 다른 클라이언트 요청을 기다립니다. 루프는 시스템 관리자 또는 Web Server가 FastCGI 응용 프로그램을 종료한 경우에만 종료됩니다.

Perl 사용

CPAN에서 최신 FCGI를 다운로드하여 설치합니다. ActivePerl의 모듈은 http://aspn.activestate.com/ASPN/Downloads/ActivePerl/PPM/Zips에서 다운로드할 수 있습니다.

Perl을 사용하여 FastCGI 응용 프로그램을 작성하는 경우에 대한 자세한 내용은 http://www.fastcgi.com/devkit/fastcgi-prog-guide/ch3perl.htm#3659를 참조하십시오.

PHP 사용

PHP 4.3.0 이후로 FastCGI가 PHP 엔진용으로 지원되는 구성이 되었습니다. FastCGI을 지원하는 PHP 4.3.x 이상의 엔진을 컴파일하려면 다음과 같이 작성 과정 중에 구성 스위치 `--enable-fastcgi`를 포함합니다.

```
./configure <other-options> --enable-fastcgi
gmake
```

컴파일이 끝나면 `php` 바이너리에 FastCGI를 사용할 수 있게 됩니다.

PHP 버전 5.1.2 이하(PHP 4.x 포함)를 사용하는 경우에는 FastCGI 플러그인을 바인드 경로와 함께 host:port 형식으로 구성해야 합니다. 예를 들면 bind-path = "localhost:3333"과 같습니다.

PHP 버전 5.1.3 이상에서는 bind-path가 선택 사항입니다. 이 값을 지정하는 경우 "host:port" 형식을 사용하면 안 됩니다. 문자열을 지정할 수 있습니다. 예를 들면 bind-path = "myphpbindpath"와 같습니다.

C/Java 사용

FastCGI 개발 키트는 FastCGI C/Java 응용 프로그램을 작성하는 API를 제공합니다. 이 키트는 http://www.fastcgi.com/devkit/doc/fcgi-devel-kit.htm에서 다운로드할 수 있습니다.

다운로드된 FastCGI 개발 키트를 구축하려면 다음 단계를 수행합니다.

1. tar 파일의 압축을 해제합니다. 이 작업으로 `fcgi-devel-kit`라는 새 디렉토리가 만들어집니다.
2. `fcgi-devel-kit` 디렉토리에서 다음 순서로 명령을 실행합니다.
   a. ./configure
   b. make

C를 사용하여 FastCGI 응용 프로그램을 작성하는 방법에 대한 자세한 내용은 http://www.fastcgi.com/devkit/doc/fcgi-devel-kit.htm#S3을 참조하십시오.

Java를 사용하여 FastCGI 응용 프로그램을 작성하는 방법에 대한 자세한 내용은 http://www.fastcgi.com/devkit/doc/fcgi-java.htm을 참조하십시오.

샘플 FastCGI 응용 프로그램

이 절에서는 PHP, Perl 및 C로 작성한 샘플 FastCGI 응용 프로그램을 소개합니다.

PHP로 작성한 응답기 응용 프로그램(ListDir.php)

```
<?php
         $dir = "/tmp/";

        // Open a known directory, and proceed to read its contents
        if (is_dir($dir)) {
            if ($dh = opendir($dir)) {
                 while (($file = readdir($dh)) !== false) {
                     echo "filename: $file : filetype: " . filetype($dir . $file) . "\n";
                 }
                closedir($dh);
            }

        }
?>
```

위 예에 사용되는 obj.conf 코드:

```
<Object name="default">
        NameTrans fn="assign-name" from="/fcgi/*" name="responder.fcgi"
</Object>
<Object name="responder.fcgi">
        Service fn="responder-fastcgi" app-path="/foo/fastcgi-enabled-php-installation/bin/php"
            bind-path="localhost:3431"  min-procs=3
</Object>
```

Perl로 작성한 인증자 응용 프로그램(SimpleAuth.pl)

```
#!/usr/bin/perl

use FCGI;

while (FCGI::accept >= 0) {
        if( $ENV{’HTTP_AUTHORIZATION’} ) {
        # This value can be further decoded to get the actual username and password and then
        # perform some kind of user validation. This program only checks for the presence of
```

```
        # of this environment param and is not really bothered about its value

        print( "Status: 200\r\n" );
        print( "\r\n" );

    } else {

        print( "Status: 401\r\n" );
        print( "WWW-Authenticate: basic realm=\"foo\"\r\n" );
        print( "\r\n" );

   }

}
Example obj.conf settings for the above example:
```

위 예에 사용되는 obj.conf 코드:

```
<Object name="responder.fcgi">
        AuthTrans fn="auth-fastcgi" app-path="/fastcgi/apps/auth/SimpleAuth.pl"
        bind-path="localhost:3432"
        Service fn="responder-fastcgi" app-path="/foo/fastcgi-enabled-php-installation/bin/php"
        bind-path="localhost:3433" app-env="PHP_FCGI_CHILDREN=8" min-procs=1
</Object>
```

`http://localhost/fcgi/php/ListDir.php`에 대한 첫 요청이 수신되면 브라우저에 인증 대화 상자가 표시됩니다. 사용자가 사용자 이름과 비밀번호를 입력하고 나면 "/tmp" 디렉토리의 내용이 나열됩니다.

C로 작성한 필터 응용 프로그램(SimpleFilter.c)

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <fcgi_stdio.h>

        void main(void) {
            size_t PageSize = 1024 * 3;
            char *page;
            FCGX_Stream *in, *out, *err;
            FCGX_ParamArray envp;

             int count=0;
            page = (char *)malloc(PageSize);

             if (page == NULL) {
```

```
    printf("Content-type: text/x-server-parsed-html\r\n");
    printf("<title>malloc failure</title>");
    printf("<h1>Cannot allocate memory to run filter. exiting</h1>");
    printf("\r\n\r\n");
    exit(2);
}

while(FCGI_Accept() >= 0) {

 char *tmp;
 char *execcgi;
 char *dataLenStr = NULL;
 int numchars = 0;
 int stdinDataSize = 0;
 int filterDataLen = 0;
 int dataToBeRead = 0;
 int x = 0;
 int loopCount = 0;


      count++;
 dataLenStr = getenv("FCGI_DATA_LENGTH");

 if(dataLenStr)
     filterDataLen = atoi(dataLenStr);

    /* clear out stdin */
 while (EOF != getc(stdin)) {
     stdinDataSize++;
 }

    dataToBeRead = filterDataLen;
FCGI_StartFilterData();
tmp = page; /** just in case fread or fwrite moves our pointer **/


    //start responding
    printf("Content-type: text/plain\r\n");
    printf("\r\n"); /** send a new line at the beginning **/
    printf("<title>SIMPLE FILTER</title>");
    printf(<h1>This page was Filtered by SimpleFilter FastCGI filter</h1>");
    printf("file size=%d<br>", filterDatalen);
    printf("stdin size=%d<br>, stdinDataSize);


    while(dataToBeRead > 0 ) {
       x = 0;
       page = tmp;
```

```
                if(dataToBeRead > PageSize)
                   x = PageSize;
                else
                    x = dataToBeRead;
                 numchars = fread((void *)(page), 1, x, stdin);

                 if( numchars == 0 )
                     continue;
                 /** at this point your data is in page pointer, so do
                 whatever you want
            with it before sending it back to the server.
                 In this example, no data is manipulated. Only the count of number of
            times the filter data is read and the total bytes read
                 at the end of every
            loop is printed. **/

                 dataToBeRead -= numchars;
                 loopCount++;
                 printf("loop count = %d ... so far read %d bytes <br>", loopCount,
                 (filterDatalen - dataToBeRead));
             }
             printf("\r\n\r\n"); /** send a new line at the end of transfer **/

             fflush(stdout);

             page = tmp; /** restore page pointer **/
             memset(page,NULL,numchars);
        }

        free(page);
}
```

위 예에 사용되는 `obj.conf` 설정의 예.

Web Server를 실행하는 시스템과 같은 시스템에서 이 FastCGI 응용 프로그램을 사용할 수 있는 경우

```
<Object name=<"filter.fcgi">
                     Service fn="filter-fastcgi" app-path="/fastcgi/apps/filter/SimpleFilter.exe"
            bind-path="localhost:3434" app-env="LD_LIBRARY_PATH=/fastcgi/fcgi-2.4/libfcgi/.libs"
</Object>
```

응용 프로그램이 원격 시스템에서 실행되는 경우에는 `obj.conf` 파일에 다음 라인이 포함되어야 합니다.

```
<Object name="filter.fcgi">
                Service fn="filter-fastcgi" bind-path="<remote-host>:<remote-port>"
</Object>
```

Web Server 인스턴스의 docroot 디렉토리에 있는 `fcgi` 디렉토리 아래에 크기가 "26868"바이트인 "FilterThisFile"이 필터링할 파일인 경우 "http://localhost/fcgi/filter/FilterThisFile"에 대한 요청의 출력은 다음과 같습니다.

```
This page was Filtered by SimpleFilter FastCGI filter


file size = 26868
stdin size = 0
loop count = 1... so far read 3072 bytes
loop count = 2... so far read 6144 bytes
loop count = 3... so far read 9216 bytes
loop count = 4... so far read 12288 bytes
loop count = 5... so far read 15360 bytes
loop count = 6... so far read 18432 bytes
loop count = 7... so far read 21504 bytes
loop count = 8... so far read 24576 bytes
loop count = 9... so far read 26868 bytes
```

◆ ◆ ◆ 부록 C

웹 서비스

Sun Java System Web Server 7.0에서 웹 서비스를 실행하는 경우 추가 구성이 필요하지 않습니다. JWSDP는 서버와 통합되며, 따라서 모든 JWSDP 웹 응용 프로그램은 웹 응용 프로그램으로 배포될 때 실행되어야 합니다.

웹 응용 프로그램 배포에 대한 자세한 내용은 159 페이지 "웹 응용 프로그램 추가"를 참조하십시오.

Web Server 7.0에서 JWSDP 2.0 샘플 실행

웹 응용 프로그램 샘플의 구성 파일은 Web Server 7.0에 배포하기 전에 JWSDP 2.0에서 수정해야 합니다. 특히, `jaxws` 샘플에 있는 구성 파일을 Web Server 7.0에 배포할 수 있게 만들려면 편집이 필요합니다.

▼ JWSDP 2.0 샘플 실행

1 **JWSDP 2.0을 다운로드합니다.**

2 **$JWSDP_HOME/jwsdp-shared/bin에 Web Server 특정 `sjsws.props`를 만듭니다.**

다음은 샘플 `sjws.props`입니다. 모든 필드는 필수입니다.

```
ADMIN_USER=admin
ADMIN_PORT=8800
ADMIN_HOST=localhost
ADMIN_PASSWORD_FILE=/tmp/admin.passwd
CONFIG=jwsdp
VS=jwsdp
WS_HOME=/export/ws7.0
WS_PORT=5555
WS_HOST=localhost
```

주 – admin.passwd 파일에는 관리자의 서버 비밀번호가 있습니다. 이 항목의 예는 다음과 같습니다. wadm_password=adminadmin

3 구성 파일을 수정합니다.

실행할 build.xml 및 etc/deploy-targets.xml 파일을 수정합니다. deploy-targets.xml에 필요한 변경 사항은 샘플에 따라 다르지 않습니다. 마스터 복사본을 사용하고 이 복사본을 실행할 응용 프로그램의 etc. 디렉토리에 복사할 수 있어야 합니다.

build.xml 변경 사항.

build.xml 맨 위의 응용 프로그램 서버 lib.home 정의를 주석으로 처리하고 Web Server lib 위치를 추가합니다. 변경된 build.xml 코드는 다음과 같습니다.

```
<!--
   **                                                    **
   ** Comment out the Application Server lib.home declaration **
   **                                                    **
    <property file="../../../jwsdp-shared/bin/sjsas.props"/>
      <condition property="lib.home" value="${DOMAIN_DIR}/../lib">
        <available file="../../../jwsdp-shared/bin/sjsas.props"/>
      </condition>
    <condition property="lib.home" value="${env.JAXWS_HOME}/lib">
      <not>
        <available file="../../../jwsdp-shared/bin/sjsas.props"/>
      </not>
    </condition>
-->
<!--
   ** Add the Web Server library location **
-->
    <property name="lib.home" value="${WS_HOME}/lib" />
```

deploy-targets.xml 변경 사항.

etc/deploy-targets.xml을 Web Server 특정 deploy-targets.xml로 바꿉니다. 이 변경 사항으로 웹 응용 프로그램을 Web Server에 배포합니다. 다음은 deploy-targets.xml 파일의 코드입니다.

```
<property environment="env"/>
<!-- Loading Web Server properties -->
<property environment="env"/>
<property file="../../../jwsdp-shared/bin/sjsws.props"/>
<property name="ws.home" value="${WS_HOME}"/>
<property name="ws.admin" value="${ws.home}/bin/wadm"/>
<property name="lib.sample.home" value="${basedir}/../lib"/>
<property name="build.home" value="${basedir}/build"/>
<property name="build.classes.home" value="${build.home}/classes"/>
```

```
<property name="build.war.home" value="${build.home}/war"/>
<property name="config" value="${CONFIG}"/>

<target name="deploy">
    <exec executable="${ws.admin}" vmlauncher="true">
      <arg value="add-webapp" />
      <arg value="--user=${ADMIN_USER}" />
      <arg value="--password-file=${ADMIN_PASSWORD_FILE}" />
      <arg value="--host=${ADMIN_HOST}" />
      <arg value="--port=${ADMIN_PORT}" />
      <arg value="--config=${CONFIG}" />
      <arg value="--vs=${VS}" />
      <arg value="--uri=/jaxws-${ant.project.name}" />
      <arg value="${build.war.home}/jaxws-${ant.project.name}.war" />
    </exec>

    <antcall target="commit-config" />
</target>

<target name="commit-config">
    <exec executable="${ws.admin}" vmlauncher="true">
      <arg value="deploy-config" />
      <arg value="--user=${ADMIN_USER}" />
      <arg value="--password-file=${ADMIN_PASSWORD_FILE}" />
      <arg value="--host=${ADMIN_HOST}" />
      <arg value="--port=${ADMIN_PORT}" />
      <arg value="--force=true" />
      <arg value="${CONFIG}" />
    </exec>
</target>
```

용어집

ACE (Access Control Entries) 웹 서버가 입중계 액세스 요청을 평가하는 데 사용하는 규칙의 계층 구조입니다.

ACL (액세스 제어 목록) ACE 컬렉션입니다. ACL은 서버에 대한 액세스 권한이 있는 사용자를 정의하는 기법입니다. 특정 파일이나 디렉토리, 또는 하나 이상의 사용자와 그룹에 액세스를 허용 또는 거부하는 ACL 규칙을 정의할 수 있습니다.

Administration Server 모든 Sun Java System Web Servers의 구성에 사용하는 형식이 들어 있는 웹 기반 서버입니다.

`admpw` Enterprise Administrator Server 수퍼유저용 아이디 및 비밀번호 파일입니다.

CGI CGI(Common Gateway Interface)입니다. 외부 프로그램이 HTTP 서버와 통신하는 인터페이스입니다. CGI 사용을 위하여 작성된 프로그램을 CGI 프로그램 또는 CGI 스크립트라 합니다. CGI 프로그램은 형식을 처리하거나 보통의 경우 서버가 처리하거나 파싱하지 않는 출력을 파싱합니다.

chroot 서버를 특정 디렉토리로 제한하기 위해 생성할 수 있는 추가 루트 디렉토리입니다. 이 기능은 보호되지 않은 서버의 안전 장치로 사용할 수 있습니다.

ciphertext 오직 수신자만이 해독할 수 있도록 암호화에 의하여 위장된 정보입니다.

DHCP 동적 호스트 구성 프로토콜(Dynamic Host Configuration Protocol)입니다. 시스템이 네트워크의 개별 컴퓨터에 동적으로 IP를 지정할 수 있게 하는 IPSP(Internet Proposed Standard Protocol)입니다.

Digest 인증 사용자가 명확한 텍스트로 아이디와 암호를 송신하지 않고 인증할 수 있는 인증 방법입니다. 브라우저는 MD5 알고리즘을 사용하여 다이제스트 값을 만듭니다. 서버는 Digest 인증 플러그인을 사용하여 클라이언트가 제공한 다이제스트 값을 비교합니다.

DNS 도메인 이름 시스템입니다. 네트워크의 시스템이 표준 IP 주소(예: 198.93.93.10)를 호스트 이름(예: `www.sun.com`)과 연결하는 데 사용되는 시스템입니다. 컴퓨터는 보통 DNS 서버에서 이 변환된 정보를 받거나 자체 시스템에 보관된 테이블에서 이 정보를 검색합니다.

DNS 별칭 DNS 서버에서 다른 호스트, 특히 DNS CNAME 레코드를 가리키는 지점을 알고 있는 호스트 이름입니다. 컴퓨터에는 항상 하나의 실제 이름이 있으나 별칭은 하나 이상일 수 있습니다. 예를 들어 `www.`*yourdomain.domain*과 같은 별칭은 `realthing.`*yourdomain.domain*과 같은 실제 시스템을 가리킬 수 있으며, 이는 서버가 현재 존재하는 도메인입니다.

drop word stop word를 참조하십시오.

expires 헤더 반환된 문서의 만료 시간으로, 원격 서버에서 지정합니다.

fancy indexing (팬시 색인화)
단순 색인화보다 많은 정보를 제공하는 색인화 방법입니다. 팬시 색인화에서는 이름별 내용 목록을 파일 크기, 최종 수정일자, 파일 유형을 나타내는 아이콘과 함께 표시합니다. 이 때문에 클라이언트 로드 시간이 단순 색인화보다 길어질 수 있습니다.

FORTEZZA
미국 정부가 중요하지만 비밀은 아닌 정보를 관리하는데 사용하는 암호화 시스템입니다.

FTP
파일 전송 프로토콜(File Transfer Protocol)입니다. 인터넷 프로토콜로 파일이 하나의 컴퓨터에서 네트워크의 다른 컴퓨터로 전송될 수 있도록 합니다.

GIF
Graphics Interchange Format의 약자입니다. 원래 CompuServe가 개발한 교차 플랫폼 이미지 형식입니다. GIF 파일은 기타 그래픽 파일 형식(BMP, TIFF)보다 훨씬 크기가 작습니다. GIF는 가장 많이 사용되는 교환 형식입니다. GIF 이미지는 UNIX, Microsoft Windows 및 Apple Macintosh 시스템에서 바로 표시할 수 있습니다.

hard restart (하드 다시 시작)
프로세스 또는 서비스를 종료시키고 다시 시작하는 과정입니다. 소프트 리스타트(soft restart)를 참조하십시오.

HTML
Hypertext Markup Language의 약자입니다. WWW(World Wide Web)에 있는 문서에 사용되는 서식 언어입니다. HTML 파일은 서식 코드가 있는 보통의 텍스트 파일로 Netscape Navigator 등의 브라우저는 서식 코드에 따라 텍스트 표시, 그래픽 및 양식 항목의 위치 및 다른 페이지로 연결되는 링크 등을 표시합니다.

HTTP
HyperText Transfer Protocol의 약자입니다. HTTP 서버와 클라이언트 사이에서 정보를 교환하는 메소드입니다.

HTTP-NG
차세대 HTTP입니다.

HTTPD
HTTP 데몬 또는 서비스의 약자로 HTTP 프로토콜을 사용하여 정보를 서비스하는 프로그램입니다.

HTTPS
안전한 버전의 HTTP로 SSL(Secure Sockets Layer)을 사용하여 구현합니다.

imagemap
이미지의 영역을 활성화시키는 프로세스로 사용자가 마우스를 사용하여 서로 다른 영역을 누르면 해당 정보로 이동하거나 정보를 구할 수 있습니다. Imagemap은 또한 "imagemap"이라는 CGI 프로그램을 말하기도 하는데, 이 경우 다른 HTTPD 구현에서 imagemap 기능을 처리하는 데 사용됩니다.

inittab (UNIX)
UNIX 파일 목록 프로그램으로 어떤 이유이든 정지된 경우 재시작해야 합니다. 이는 프로그램이 지속적으로 실행되도록 합니다. 이 파일의 위치 때문에 `/etc/inittab`이라고 하는 경우도 있습니다. 모든 UNIX 시스템에서 이 파일을 사용할 수 있는 것은 아닙니다.

IP 주소
인터넷 프로토콜 주소입니다. 마침표로 분리된 일련의 번호로 인터넷에 있는 컴퓨터의 실제 위치를 지정합니다(예: 198.93.93.10).

ISDN
종합 정보 통신망(Integrated Services Digital Network)입니다.

ISINDEX
클라이언트에서 검색이 시작되도록 하는 HTML 태그입니다. 문서는 네트워크 네비게이터의 기능을 이용하여 검색 문자열을 받아들이고 이를 서버로 보내 검색 가능한 색인에 액세스합니다. 이 때 양식은 사용하지 않습니다. <ISINDEX>를 사용하려면 쿼리 처리기를 만들어야 합니다.

ISMAP ISMAP은 HTML 문서에서 사용하는 IMG SRC의 확장자로 서버에게 해당 이름의 이미지가 imagemap임을 알려줍니다.

ISP 인터넷 서비스 제공자(Internet Service Provider)입니다. 인터넷 연결을 제공하는 단체입니다.

Java Sun Microsystems가 개발한 개체 지향형 프로그램 언어로 애플릿이라고 하는 실시간 대화형 프로그램을 만들 때 사용합니다.

Java 서블릿 인스턴스화, 초기화, 제거, 기타 구성 요소로부터의 액세스 및 구성 관리 등을 포함하여 모든 Java 서브릿 메타 기능을 사용할 수 있도록 하는 확장 기능입니다. Java 서브릿은 재사용 가능한 Java 응용 프로그램으로 웹 브라우저가 아닌 웹 서버에서 실행됩니다.

JavaScript 클라이언트 및 서버 인터넷 응용 프로그램 개발용의 소형 개체 지향형 스크립트 언어입니다.

JavaServer Pages 인스턴스화, 초기화, 제거, 기타 구성 요소로부터의 액세스 및 구성 관리 등을 포함하여 모든 JavaServer 페이지 메타기능을 사용할 수 있도록 하는 확장 기능. JSP는 재사용 가능한 Java 응용 프로그램으로 웹 브라우저가 아닌 웹 서버에서 실행됩니다.

LDAP 데이터베이스 인증용으로 사용자 및 그룹 목록이 저장된 데이터베이스입니다.

`magnus.conf` 기본 Web Server 구성 파일입니다. 이 파일에는 전역 서버 구성 정보(포트, 보안 등)가 포함됩니다. 이 파일은 초기화 동안 서버를 구성하는 변수의 값을 설정합니다. Enterprise Server는 이 파일을 읽어 시작시 변수 설정을 실행합니다. 서버는 다시 시작할 때까지 이 파일을 다시 읽지 않으므로 이 파일이 변경되는 경우 매번 서버를 다시 시작해야 합니다.

MD5 RSA Data Security에서 만든 메시지 다이제스트 알고리즘입니다. MD5는 고유성과 고도의 이식성을 갖춘 짧은 다이제스트 데이터를 만들 수 있습니다. 동일한 메시지 다이제스트 전자 메일을 수학적으로 만들어 내는 것은 매우 어려운 작업입니다.

MD5 서명 MD5 알고리즘으로 만든 메시지 다이제스트입니다.

MIB 관리 정보 베이스(Management Information Base)입니다.

MIME Multi-Purpose Internet Mail Extensions의 약자입니다. 멀티미디어 전자 메일 및 메시징용으로 새롭게 등장한 표준입니다.

mime 유형 MIME(Multi-purpose Internet Mail Extension) 유형 구성 파일입니다. 이 파일은 파일 확장자와 MIME 유형을 매핑하며, 이에 따라 서버가 요청되는 컨텐츠의 유형을 결정할 수 있습니다. 예를 들어,`html` 확장자를 갖는 자원에 대한 요청은 클라이언트가 HTML 파일을 요청하고 있음을 나타내고, `.gif` 확장자를 갖는 자원에 대한 요청은 클라이언트가 GIF 형식의 이미지 처리 파일을 요청하고 있음을 나타냅니다.

modutil 외부 암호화 또는 하드웨어 가속 장치용으로 PKCS#11을 설치할 때 필요한 소프트웨어 유틸리티입니다.

MTA Message Transfer Agent의 약자입니다. 서버에서 에이전트 서비스를 사용하려면 서버의 MTA Host를 정의해야 합니다.

NIS (UNIX) Network Information Service의 약자입니다. UNIX 컴퓨터가 컴퓨터 네트워크 전체에서 컴퓨터, 사용자, 파일 시스템 및 네트워크 매개 변수 등에 대한 특정 정보를 수집, 대조 및 공유하는데 사용하는 프로그램 및 데이터 파일 시스템입니다.

NNTP 뉴스그룹에 사용되는 Network News Transfer Protocol의 약자입니다. 서버에서 에이전트 서비스를 사용하려면 반드시 뉴스 서버 호스트를 정의해야 합니다.

obj.conf 서버의 객체 구성 파일입니다. 이 파일에는 추가의 초기화 정보, 서버 사용자 정의용 설정 및 서버가 클라이언트(브라우저 등)의 요청을 처리할 때 사용하는 지시문 등이 포함됩니다. Sun Java System Web Server는 클라이언트 요청을 처리할 때마다 이 파일을 읽습니다.

pk12util 내부 컴퓨터에서 인증서와 키 데이터베이스를 내보내고 이를 외부의 PKCS#11 모듈로 가져오기 위하여 필요한 소프트웨어 유틸리티입니다.

RAM Random access memory의 약자입니다. 컴퓨터에 있는 실제의 반도체 메모리입니다.

rc.2.d (UNIX) UNIX 시스템에 있는 파일로 시스템이 시작할 때 실행되는 프로그램을 기술합니다. 저장 위치 때문에 /etc/rc.2.d라고도 합니다.

RFC Request For Comments입니다. 보통 인터넷 커뮤니티로 제출되는 프로시저 또는 표준 문서입니다. 표준을 수용하기 전에 기술에 대한 의견을 보낼 수 있습니다.

simple index (단순 색인) 고급 색인과 반대이며, 이 유형의 디렉토리 목록에는 그래픽 요소가 없는 파일의 이름만 표시됩니다.

SNMP Simple Network Management Protocol의 약자입니다.

SOCKS 내부에서 외부로의 직접 연결이 방화벽 소프트웨어 또는 하드웨어(라우터 구성 등)에 의해 금지된 경우 방화벽 내부에서 외부로 연결을 설정하는 방화벽 소프트웨어입니다.

SSL Secure Sockets Layer의 약자입니다. HTTP의 보안 버전인 HTTPS를 구현할 때 사용되는 양쪽(클라이언트와 서버) 사이에 안전한 연결을 설정하는 소프트웨어 라이브러리입니다.

SSL 인증 클라이언트 인증서에 있는 정보를 아이디의 증거로 사용하거나 LDAP 디렉토리에 게시된 클라이언트 인증서를 확인하여 해당 보안 인증서가 있는 사용자의 아이디를 확인합니다.

stop word 검색 기능에서 검색을 진행하지 않도록 인식하는 단어입니다. 보통 a, an, and 등의 단어가 포함됩니다. 또한 drop word라고도 합니다.

strftime 날짜와 시간을 문자열로 변환하는 함수입니다. 서버에서 접미부를 추가할 때 사용됩니다. strftime에는 서버가 최종 수정 날짜를 보여주기 위해 접미부에서 사용할 수 있는 시간 및 날짜에 대한 특수 형식 언어가 있습니다.

Sun Java System Web Server 관리 콘솔 서버 관리자가 엔터프라이즈 네트워크에 있는 중앙의 위치 한 곳에서 모든 Sun Java System Web Server를 관리할 수 있는 그래픽 인터페이스를 제공하는 Java 응용 프로그램입니다. Sun Java System Web Server 관리 콘솔이 설치된 위치에서 기업의 네트워크에 있는 서버 중 액세스 권한이 부여된 모든 Sun Java System 서버를 확인하고 액세스할 수 있습니다.

Sym-links (UNIX) 심볼 링크의 약자로 UNIX 운영 체제가 사용하는 재지정 유형 중 한 가지입니다. Sym-link를 사용하여 파일 시스템의 일정 부분에서 파일 시스템의 다른 부분에 있는 기존 파일 또는 디렉토리로 향하는 포인터를 만들 수 있습니다.

TCP/IP Transmission Control Protocol/Internet Protocol입니다. 인터넷 및 기업 네트워크용 기본 네트워크 프로토콜입니다.

telnet 네트워크에 있는 두 대의 시스템이 서로 연결되고 원격 로그인용 단말기 에뮬레이션을 지원하는 프로토콜입니다.

TLS Secure Sockets Layer입니다. HTTP의 보안 버전인 HTTPS를 구현할 때 사용되는 양쪽(클라이언트와 서버) 사이에 안전한 연결을 설정하는 소프트웨어 라이브러리입니다.

top (UNIX) 일부 UNIX 시스템에 있는 프로그램으로 시스템 리소스의 현재 사용 상태를 표시입니다.

uid (UNIX) UNIX 시스템에 있는 각 사용자와 연결된 고유 번호입니다.

URI Uniform Resource Identifier의 약자입니다. 단축 URL을 사용함으로써 추가의 보안을 제공하는 파일 아이디입니다. URL의 첫 부분은 URL 매핑으로 대체되기 때문에 사용자는 파일의 실제 경로 전체를 알 수 없게 됩니다. URL 매핑을 참조하십시오.

URL Uniform Resource Locator의 약자입니다. 서버와 클라이언트가 문서를 요청할 때 사용하는 주소 지정 체계입니다. URL은 또한 위치라고도 합니다. URL의 형식은 *protocol://machine:port/document*입니다.

URL의 예로 `http://www.sun.com/index.html`이 있습니다.

URL 데이터베이스 수정 소프트웨어 장애, 시스템 고장, 디스크 손상 또는 파일 시스템의 사용 공간 부족 등으로 손상이 발생한 경우 URL 데이터베이스를 수리하고 업데이트하는 프로세스입니다.

URL 매핑 문서 디렉토리의 실제 경로를 사용자 정의 별칭으로 매핑하여 디렉토리에 있는 파일이 실제 경로 이름이 아닌 디렉토리의 별칭으로 액세스되도록 하는 프로세스입니다. 따라서 파일을 `usr/sun/servers/docs/index.html`로 지정하는 것이 아니라 `/myDocs/index.html`로 지정할 수 있습니다. 사용자가 서버 파일의 실제 위치를 알 필요가 없으므로 추가의 보안 기능을 제공합니다.

WAR (Web Application Archive) 웹 응용 프로그램 전체가 압축된 형식으로 포함되어 있는 아카이브 파일입니다.

Windows CGI (Windows) Visual Basic 등의 Windows 기반 프로그램 언어로 작성된 CGI 프로그램입니다.

가상 서버 설치된 하나의 서버에서 여러 개의 도메인 이름, IP 주소 및 서버 모니터 기능을 설정하는 한 가지 방법입니다.

가상 서버 클래스 obj.conf 파일의 동일한 기본 구성을 공유하는 가상 서버의 집합입니다.

개인 키 공용 키 암호화에서 사용되는 해독 키입니다.

공용 정보 디렉토리 (UNIX) UNIX 사용자의 홈 디렉토리에서 문서 루트 디렉토리가 아닌 다른 위치의 디렉토리, 또는 사용자가 제어하는 디렉토리입니다.

공용 키 공용 키 암호화에서 사용되는 암호화 키입니다.

공통 로그 파일 형식 서버에서 액세스 로그에 정보를 입력하기 위해 사용하는 형식입니다. 형식은 Sun Java System Web Server를 포함하여 모든 주요 서버에서 동일합니다.

네트워크 관리 스테이션 (NMS) 사용자가 원격으로 네트워크를 관리할 때 사용하는 컴퓨터입니다. 관리 대상 장치는 호스트, 라우터 및 웹 서버 등의 SNMP를 실행하는 모든 장치입니다. NMS는 보통 하나 이상의 네트워크 관리 응용 프로그램이 설치된 고기능 워크스테이션입니다.

데몬 (UNIX) 특정 시스템 작업을 담당하는 이면의 프로세스입니다.

루트 (UNIX) UNIX 시스템에서 가장 강력한 권한을 가진 사용자입니다. 루트 사용자는 컴퓨터의 모든 파일에 액세스할 수 있는 완전한 권한을 가집니다.

리디렉션 특정 URL로 액세스하는 클라이언트가 동일한 서버 또는 다른 서버의 다른 위치로 전송되는 시스템입니다. 이 시스템은 리소스가 이동되었으나 클라이언트가 새 위치를 사용할 때 이를 알 수 없도록 하는 경우에 유용합니다. 또한 디렉토리에 액세스할 때 끝에 슬래시가 없는 경우 관련 링크의 무결성을 유지하는 데도 사용됩니다.

모음 단어 목록과 파일 등록 정보 등 설명서에 대한 정보가 포함된 데이터베이스입니다. 검색 기능은 컬렉션을 사용하여 지정된 검색 범주와 일치하는 문서를 검색합니다.

문서 루트 서버에 액세스하는 사용자에게 제시할 파일, 이미지 및 데이터 등이 들어있는 서버 컴퓨터의 디렉토리입니다.

방화벽 보통 하드웨어와 소프트웨어 모두를 사용하는 네트워크 구성으로 조직 안에서 네트워크된 컴퓨터를 외부 액세스로부터 보호합니다. 방화벽은 보통 실제 건물 또는 조직 사이트 내에서 네트워크에 있는 전자 메일 및 데이터 파일 등의 정보를 보호하는 데 사용됩니다.

변조된 키 목록 (CKL) 키를 조작한 사용자에 대한 주요 정보 목록입니다. CA 또한 이 목록을 제공합니다.

비밀번호 파일 (UNIX) UNIX 시스템에 있는 파일로 UNIX 사용자 로그인 이름, 비밀번호 및 사용자 아이디 번호를 저장합니다. 또한 저장 위치 때문에 `/etc/passwd`라고도 합니다.

서버 데몬 실행 중에 클라이언트에서 요청을 수신하고 허용하는 프로세스입니다.

서버 루트 서버 프로그램, 구성, 유지 관리 및 정보 파일 전용으로 할당된 서버 시스템의 디렉토리입니다.

서버 플러그인 API Sun Java System Web Servers의 핵심 기능을 확장 및/또는 사용자 정의하고 HTTP 서버와 백엔드 응용 프로그램 사이의 인터페이스를 구축하는 용도로 확장성 및 효율성을 갖춘 기법을 제공하는 확장 기능입니다. NSAPI라도고 합니다.

서비스 품질 서버 인스턴스, 가상 서버 클래스 또는 가상 서버용으로 설정한 성능 한계입니다.

소프트 리스타트 (soft restart) 서버를 재시작하는 한 가지 방법으로 서버는 내부적으로 재시작하며 해당 구성 파일을 다시 읽습니다. 소프트 리스타트는 `HUP` 신호(signal number one) 프로세스를 보냅니다. 하드 리스타트와는 달리 프로세스 그 자체는 종료되지 않습니다.

수퍼유저 (UNIX) UNIX 시스템에서 가장 강력한 권한을 가진 사용자이며, 루트라고도 합니다. 수퍼유저는 컴퓨터의 모든 파일에 액세스할 수 있는 완전한 권한을 가집니다.

시간 초과 서버가 고착된 것으로 보이는 서비스 루틴에 대한 시도를 포기하도록 지정된 시간입니다.

신 소켓 포트 번호와 IP 주소의 조합. 서버와 클라이언트 사이의 연결은 청취 소켓에서 이루어진다.

암호 암호는 암호화 알고리즘(수학적 함수)으로 암호화 또는 복호화에 사용됩니다.

암호화 대상의 수신자를 제외한 누구도 해독하거나 읽을 수 없도록 정보를 변환하는 프로세스입니다.

에이전트 라우터, 호스트 또는 X 단말기 등 네트워크 장치에서 네트워크 관리 소프트웨어를 실행하는 소프트웨어입니다. 지능형 에이전트를 참조하십시오.

엑스트라넷 회사의 인트라넷을 인터넷으로 확장한 것으로 고객, 공급자 및 원격 작업자가 데이터에 액세스할 수 있습니다.

웹 응용 프로그램 서블릿, JavaServer Page, HTML 문서 및 기타 웹 자원의 집합으로 이미지 파일, 압축된 아카이브 및 기타 데이터가 포함될 수 있습니다. 웹 응용 프로그램은 저장 파일로 패키지화되거나(WAR 파일) 또는 개방형 디렉토리 구조로 존재할 수 있습니다.

유연한 로그 형식 서버가 액세스 로그에 정보를 입력하는 용도로 사용하는 형식입니다.

인증 용어집에서 서버에 대한 아이디를 확인할 수 있게 합니다. 기본 또는 출하시 인증의 경우 사용자가 웹 서버 또는 웹 사이트에 액세스하기 위한 아이디와 비밀번호를 입력해야 합니다. LDAP 데이터베이스에 사용자 및 그룹 목록이 있어야 합니다. 다이제스트 및 SSL 인증을 참조하십시오.

서버 전체 또는 서버의 특정 파일 및 디렉토리에 액세스할 권한을 부여합니다. 인증은 호스트 이름과 IP 주소를 포함한 범주로 제한할 수 있습니다.

인증 기관 (CA) 내부 또는 제3자 조직으로 암호화된 트랜잭션용으로 사용되는 디지털 파일을 발행합니다.

인증서 제 3자가 발행하며 통신하는 양 쪽이 이미 신뢰하는 양도 불가 및 위조 불가 디지털 파일입니다.

인증서 철회 목록 (CRL) CA가 제공하는 모든 철회된 인증서에 대한 CA 목록입니다.

자원 서버가 액세스하여 이를 요청하는 클라이언트에 전송하는 모든 문서(URL), 디렉토리 또는 프로그램입니다.

지능형 에이전트 사용자를 대신하여 다양한 요청(HTTP, NNTP, SMTP 및 FTP 요청)을 수행하는 서버 내의 개체입니다. 어떤 면에서 지능형 에이전트는 서버에 대해 서버가 이행하는 요청을 하는 클라이언트의 역할을 합니다.

최상위 도메인 권한 호스트 이름 분류에 사용되는 가장 높은 범주로, 보통 도메인이 있는 조직(예: `com`은 회사, `edu`는 교육 기관) 또는 해당 도메인의 국가(예: `.us`는 미국, `.jp`는 일본, `.au`는 오스트레일리아, `.fi`는 핀란드)를 나타냅니다.

최종 수정된 헤더 서버에서 HTTP 응답으로 반환된 문서 파일의 최종 수정 시간입니다.

캐시 로컬에 저장된 원본 데이터의 사본입니다. 캐시된 데이터가 다시 요청되는 경우 원격 서버에서 다시 검색할 필요가 없습니다.

클라이언트 WWW(World Wide Web) 자료를 요청 및 확인하는 데 사용하는 Netscape Navigator 등의 소프트웨어입니다.

클라이언트 인증 클라이언트 인증입니다.

클러스터 ”마스터’ 및 Administration Server에 추가되고 이 서버에 의해 제어되는 원격 ”슬레이브’ Administration Server의 그룹입니다. 클러스터의 모든 서버는 동일한 플랫폼에 있어야 하며 동일한 사용자 아이디와 비밀번호가 있어야 합니다.

파일 유형 파일의 형식입니다. 예를 들어 그래픽 파일의 형식은 텍스트 파일의 형식과 다릅니다. 파일 유형은 주로 파일 확장자(`.gif` 또는 `.html`)를 통해 식별합니다.

파일 확장자 파일 이름의 마지막 부분으로 보통 파일의 유형을 정의합니다. 예를 들어 `index.html`이라는 파일 이름에서 파일 확장자는 `html`입니다.

프로토콜 네트워크에 있는 장치가 정보를 교환하는 방법을 기술한 일련의 규칙입니다.

호스트 이름 *machine.domain.dom* 형식의 시스템 이름이며 IP 주소로 변환됩니다. 예를 들어 `www.sun.com`은 `com` 도메인의 `sun` 하위 도메인에 있는 `www` 시스템을 나타냅니다.

홈 페이지 서버에 존재하는 문서로 서버의 내용에 대한 카탈로그 또는 진입 지점의 역할을 합니다. 이 문서의 위치는 서버의 구성 파일에 정의됩니다.

색인