



Sun Java System Application Server 9.1 クイックスタートガイド



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

Part No: 820-3028
2007年9月

本製品および本書は著作権法によって保護されており、その使用、複製、頒布、および逆コンパイルを制限するライセンスのもとにおいて頒布されます。サン・マイクロシステムズ株式会社による事前の許可なく、本製品および本書のいかなる部分も、いかなる方法によっても複製することが禁じられます。フォント技術を含む第三者のソフトウェアは、著作権により保護されており、提供者からライセンスを受けているものです。

本製品の一部は Berkeley BSD システムより派生したもので、カリフォルニア大学よりライセンスを受けています。UNIX は、X/Open Company, Ltd. が独占的にライセンスしている米国ならびにほかの国における登録商標です。

Sun、Sun Microsystems、Sun のロゴマーク、docs.sun.com、AnswerBook、AnswerBook2、および Solaris は、米国およびその他の国における米国 Sun Microsystems, Inc. (以下、米国 Sun Microsystems 社とします) の商標もしくは登録商標です。Sun のロゴマークおよび Solaris は、米国 Sun Microsystems 社の登録商標です。すべての SPARC 商標は、米国 SPARC International, Inc. のライセンスを受けて使用している同社の米国およびその他の国における商標または登録商標です。SPARC 商標が付いた製品は、米国 Sun Microsystems 社が開発したアーキテクチャーに基づくものです。

OPEN LOOK および SunTM Graphical User Interface は、米国 Sun Microsystems 社が自社のユーザーおよびライセンス実施権者向けに開発しました。米国 Sun Microsystems 社は、コンピュータ産業用のビジュアルまたはグラフィカルユーザーインターフェースの概念の研究開発における米国 Xerox 社の先駆者としての成果を認めるものです。米国 Sun Microsystems 社は米国 Xerox 社から Xerox Graphical User Interface の非独占的ライセンスを取得しており、このライセンスは、OPEN LOOK GUI を実装するか、または米国 Sun Microsystems 社の書面によるライセンス契約に従う米国 Sun Microsystems 社のライセンス実施権者にも適用されます。

U.S. Government Rights – Commercial software. Government users are subject to the Sun Microsystems, Inc. standard license agreement and applicable provisions of the FAR and its supplements.

本書は、「現状のまま」をベースとして提供され、商品性、特定目的への適合性または第三者の権利の非侵害の黙示の保証を含みそれに限定されない、明示的であるか黙示的であるかを問わない、なんらの保証も行われぬものとします。

目次

はじめに	5
1 基本機能のクイックスタート	13
Application Server の管理ツール	13
サーバーの起動	15
▼ Solaris および Linux でドメインを起動するには	16
▼ Windows でドメインを起動するには	16
▼ 管理コンソールにログオンするには	18
▼ ドメイン管理サーバーのログファイルを表示するには	19
▼ ノードエージェントを起動するには	21
▼ ドメインにクラスタサポートを追加する	22
▼ バンドルされた Java DB サーバーを起動するには	23
次の学習ステップ	23
2 クラスタの設定	25
クラスタの作成	25
▼ 管理コンソールを使用してクラスタを作成するには	25
▼ CLI コマンドを使用してクラスタを作成するには	26
クラスタ化されたサーバーインスタンスのポートの表示	27
▼ クラスタ化されたサーバーインスタンスのポートを表示するには	27
3 アプリケーションの配備	29
サンプル Web アプリケーションの配備	29
▼ 自動配備を使用して Hello アプリケーションを配備するには	29
▼ 管理コンソールを使用して Hello アプリケーションを配備するには	31
サンプルのエンタープライズアプリケーションのクラスタへの配備	32
▼ 管理コンソールを使用してサンプルのエンタープライズアプリケーションを配備	

するには	32
▼ クラスタ内でインスタンスを起動するには	33
▼ アプリケーションの配備を確認するには	34
4 ロードバランサの操作	35
ロードバランスの設定	36
▼ ロードバランスを設定するには	36
ロードバランスの起動	37
ロードバランスの検証	37
▼ ロードバランスを検証するには	37
インメモリーレプリケーション機能を使用した高可用性とフェイルオーバー	39
5 高可用性フェイルオーバーの設定	41
高可用性クラスタと HADB	42
HADB インストール前の手順	42
▼ HADB に対応するようにシステムを設定するには	43
HADB のインストール	44
▼ HADB をインストールするには	44
HADB の起動	45
▼ Solaris または Linux 上の Java Enterprise System インストールで HADB を起動するには	45
▼ Windows 上の Java Enterprise System インストールで HADB を起動するには	45
▼ Solaris または Linux 上のスタンドアロンインストールで HADB を起動するには	46
▼ Widows 上のスタンドアロンインストールで HADB を起動するには	46
高可用性のためのクラスタおよびアプリケーションの設定	46
クラスタの再起動	46
▼ クラスタを再起動するには	47
HTTP セッションフェイルオーバーの検証	47
▼ HTTP セッションフェイルオーバーを検証するには	47

はじめに

このクイックスタートガイドでは、サーバーを起動し、アプリケーションを配備し、ロードバランスやフェイルオーバーを設定するための基本的な手順について説明します。

ここでは、Sun Java™ System Application Server のマニュアルセット全体に関する情報と表記規則について説明しています。

Application Server のマニュアルセット

Application Server のマニュアルセットは、配備の計画とシステムのインストールについて説明しています。Application Server のマニュアルの URL は、<http://docs.sun.com/coll/1343.4> です。Application Server への導入としては、次の表に示されている順序でマニュアルを参照してください。

表 P-1 Application Server のマニュアルセットの内容

マニュアルタイトル	説明
『Documentation Center』	タスクや主題ごとに整理された Application Server のマニュアルのトピック。
『リリースノート』	ソフトウェアとマニュアルに関する最新情報。サポートされているハードウェア、オペレーティングシステム、Java Development Kit (JDK™)、およびデータベースドライバの包括的な表ベースの概要を含みます。
『クイックスタートガイド』	Application Server 製品の使用を開始するための手順。
『Installation Guide』	ソフトウェアとそのコンポーネントのインストール。
『配備計画ガイド』	最適な方法で確実に Application Server を導入するための、システムニーズや企業ニーズの分析。サーバーを配備する際に注意すべき一般的な問題や懸案事項についても説明しています。
『アプリケーション配備ガイド』	アプリケーションおよびアプリケーションコンポーネントの Application Server への配備。配備記述子に関する情報を含みます。

表 P-1 Application Server のマニュアルセットの内容 (続き)

マニュアルタイトル	説明
『開発者ガイド』	Application Server 上で動作することを目的とし、Java EE コンポーネントおよび API のオープン Java スタンダードモデルに準拠した、Java 2 Platform, Enterprise Edition (Java EE プラットフォーム) アプリケーションの作成と実装。開発者ツール、セキュリティ、デバッグ、ライフサイクルモジュールの作成に関する情報を含みます。
『Java EE 5 Tutorial』	Java EE 5 プラットフォームテクノロジーと API を使用した Java EE アプリケーションの開発。
『Java WSIT Tutorial』	Web サービス相互運用性テクノロジー (WSIT) を使用した Web アプリケーションの開発。WSIT テクノロジーを使用する方法、時期、および理由と、各テクノロジーがサポートする機能およびオプションについて説明します。
『管理ガイド』	設定、監視、セキュリティ、資源管理、および Web サービス管理を含む Application Server のシステム管理。
『高可用性 (HA) 管理ガイド』	高可用性 (HA) データベースのためのインストール後の設定と管理の手順。
『Administration Reference』	Application Server 設定ファイル domain.xml の編集。
『アップグレードと移行』	旧バージョンの Application Server からのアップグレード、または競合するアプリケーションサーバーからの Java EE アプリケーションの移行。このガイドでは、直前の製品リリースとの違いと、製品仕様との互換性がなくなる可能性のある設定オプションについても説明します。
『パフォーマンスチューニングガイド』	パフォーマンスを向上させるための Application Server の調整。
『トラブルシューティングガイド』	Application Server の問題の解決。
『Error Message Reference』	Application Server のエラーメッセージの解決。
『Reference Manual』	Application Server で使用できるユーティリティコマンド。マニュアルページのスタイルで記述されています。asadmin コマンド行インタフェースを含みます。

関連マニュアル

Application Server は、単体で購入することが可能です。あるいは、ネットワークまたはインターネット環境にわたって分散しているエンタープライズアプリケーションをサポートするソフトウェアインフラストラクチャーである Sun Java™ Enterprise System (Java ES) のコンポーネントとして購入することもできます。Application Server を Java ES のコンポーネントとして購入した場合は、<http://docs.sun.com/app/docs/coll/1657.1> にあるシステムマニュアルをよく読むことをお勧めします。Java ES およびそのコンポーネントに関するすべてのマニュアルの URL は <http://docs.sun.com/prod/entsys.5> です。

その他のスタンドアロンの Sun Java System サーバー製品については、次のマニュアルを参照してください。

- Message Queue マニュアル (<http://docs.sun.com/app/docs/prod/message#hic>)
- Directory Server マニュアル (<http://docs.sun.com/app/docs/prod/dirsrv.ee#hic>)
- Web Server マニュアル (<http://docs.sun.com/app/docs/prod/websrv#hic>)

Application Server とともに提供されるパッケージの Javadoc™ ツールのリファレンスは、<http://glassfish.dev.java.net/nonav/javaee5/api/index.html> にあります。また、次の資料も有用です。

- Java EE 5 Specifications (<http://java.sun.com/javaee/5/javatech.html>)
- Java EE Blueprints (<http://java.sun.com/reference/blueprints/index.html>)

NetBeans™ Integrated Development Environment (IDE) におけるエンタープライズアプリケーションの作成については、<http://www.netbeans.org/kb/55/index.html> を参照してください。

Application Server に含まれる Java DB データベースについては、<http://developers.sun.com/javadb/> を参照してください。

GlassFish Samples プロジェクトは、広い範囲の Java EE テクノロジーの実例を示すサンプルアプリケーションを集めたものです。GlassFish Samples は Java EE Software Development Kit (SDK) にバンドルされており、GlassFish Samples プロジェクトページ (<https://glassfish-samples.dev.java.net/>) から入手できます。

デフォルトのパスおよびファイル名

次の表は、このマニュアルで使用されているデフォルトのパス名とファイル名について説明したものです。

表P-2 デフォルトのパスおよびファイル名

プレースホルダ	説明	デフォルト値
<i>install-dir</i>	Application Server のベースインストールディレクトリを表します。	<p>Java ES とともに Solaris™ プラットフォームへインストールされる Application Server で、root ユーザーである場合:</p> <p><i>/opt/SUNWappserver/appserver</i></p> <p>Java ES とともに Linux プラットフォームへインストールされる Application Server で、root ユーザーである場合:</p> <p><i>/opt/sun/appserver/</i></p> <p>Solaris および Linux プラットフォームへのそのほかのインストールで、root ユーザーでない場合:</p> <p><i>user's-home-directory/SUNWappserver</i></p> <p>Solaris および Linux プラットフォームへのそのほかのインストールで、root ユーザーである場合:</p> <p><i>/opt/SUNWappserver</i></p> <p>Windows のすべてのインストールの場合:</p> <p><i>SystemDrive:\Sun\AppServer</i></p>
<i>domain-root-dir</i>	すべてのドメインを含むディレクトリを表します。	<p>Java ES Solaris プラットフォームへのインストールで、root ユーザーである場合:</p> <p><i>/var/opt/SUNWappserver/domains/</i></p> <p>Java ES Linux プラットフォームへのインストールで、root ユーザーである場合:</p> <p><i>/var/opt/sun/appserver/domains/</i></p> <p>そのほかのすべてのインストールの場合:</p> <p><i>install-dir/domains/</i></p>
<i>domain-dir</i>	ドメインのディレクトリを表します。 設定ファイルには、次のように表される <i>domain-dir</i> があります。 <i>#{com.sun.aas.instanceRoot}</i>	<i>domain-root-dir/domain-dir</i>
<i>instance-dir</i>	サーバーインスタンスのディレクトリを表します。	<i>domain-dir/instance-dir</i>

表記上の規則

このマニュアルでは、次のような字体や記号を特別な意味を持つものとして使用します。

表 P-3 表記上の規則

字体または記号	意味	例
AaBbCc123	コマンド名、ファイル名、ディレクトリ名、画面上のコンピュータ出力、コード例を示します。	.login ファイルを編集します。 ls -a を使用してすべてのファイルを表示します。 machine_name% you have mail.
AaBbCc123	ユーザーが入力する文字を、画面上のコンピュータ出力と区別して示します。	machine_name% su Password:
<i>aabbcc123</i>	変数を示します。実際に使用する特定の名前または値で置き換えます。	ファイルを削除するには、rm <i>filename</i> と入力します。
『』	参照する書名を示します。	『コードマネージャー・ユーザーズガイド』を参照してください。
「」	参照する章、節、ボタンやメニュー名、強調する単語を示します。	第 5 章「衝突の回避」を参照してください。 この操作ができるのは、「スーパーユーザー」だけです。
\	枠で囲まれたコード例で、テキストがページ行幅を超える場合に、継続を示します。	sun% grep '^#define \ XV_VERSION_STRING

コード例は次のように表示されます。

- C シェル

```
machine_name% command y|n [filename]
```

- C シェルのスーパーユーザー

```
machine_name# command y|n [filename]
```

- Bourne シェルおよび Korn シェル

```
$ command y|n [filename]
```

- Bourne シェルおよび Korn シェルのスーパーユーザー

```
# command y|n [filename]
```

[] は省略可能な項目を示します。上記の例は、*filename* は省略してもよいことを示しています。

| は区切り文字 (セパレータ) です。この文字で分割されている引数のうち 1 つだけを指定します。

キーボードのキー名は英文で、頭文字を大文字で示します (例: Shift キーを押します)。ただし、キーボードによっては Enter キーが Return キーの動作をします。

ダッシュ (-) は 2 つのキーを同時に押すことを示します。たとえば、Ctrl-D は Control キーを押したまま D キーを押すことを意味します。

記号の表記ルール

この表は、このマニュアルで使用される記号について説明したものです。

表 P-4 記号の表記ルール

記号	説明	例	意味
[]	省略可能な引数やコマンドオプションが含まれます。	ls [-l]	-l オプションは必須ではありません。
{ }	必須コマンドオプションの選択項目が含まれています。	-d {y n}	-d オプションには、y 引数または n 引数のいずれかを使用する必要があります。
\${ }	変数参照を示します。	\${com.sun.javaRoot}	com.sun.javaRoot 変数の値を参照します。
-	同時に実行する複数のキーストロークを結び付けます。	Control-A	コントロールキーを押しながら A キーを押します。
+	連続で複数のキーストロークを行います。	Ctrl + A + N	Control キーを押して離してから、次のキーを押します。
→	グラフィカルユーザーインタフェースでのメニュー項目の選択を示します。	「ファイル」 → 「新規」 → 「テンプレート」	「ファイル」メニューから「新規」を選択します。「新規」サブメニューから、「テンプレート」を選択します。

マニュアル、サポート、およびトレーニング

Sunのサービス	URL	内容
マニュアル	http://jp.sun.com/documentation/	PDF 文書および HTML 文書をダウンロードできます。
サポートおよび トレーニング	http://jp.sun.com/supporttraining/	技術サポート、パッチのダウンロード、および Sun のトレーニングコース情報を提供します。

基本機能のクイックスタート

『クイックスタートガイド』へようこそ。このガイドでは、Application Server を起動して使用するためのサンプルとなる手順について説明します。

このマニュアルに含まれる手順を使用する前に、Application Server ソフトウェアをインストールしておく必要があります。

ここでは、次の内容について説明します。

- 13 ページの「Application Server の管理ツール」
- 15 ページの「サーバーの起動」
- 23 ページの「次の学習ステップ」

Application Server の管理ツール

管理者が複数のホスト上で実行されるサーバーインスタンスとクラスタを管理できるようにするために、Application Server は次のツールを提供します。

- ブラウザベースのグラフィカルユーザーインターフェース (GUI) 管理コンソール
- `asadmin` ユーティリティーなどのコマンド行ツール。Application Server で使用可能なコマンド行ツールの完全なリストについては、表 1-1 を参照してください。
- プログラム可能な Java Management Extensions (JMX™) API

これらのツールは「ドメイン管理サーバー」と呼ばれるサーバーに接続します。このサーバーは、すべての管理タスクに介在する特別に指定された Application Server インスタンスです。ドメイン管理サーバー (DAS) は、使用するインターフェースに関係なく、管理コマンドを検証し、実行するためのセキュリティで保護された単一のインターフェースを提供します。

「ドメイン」とは、設定データ、配備されたアプリケーション、指定された管理者によるマシンの集合体です。ドメイン定義は、複数のアプリケーション、スタンドアロンのアプリケーションサーバーインスタンス、およびクラスタの動作を記述

し、その制御を可能にし、複数のマシンに配布することができます。DAS をインストールすると、*domain1* という名前のデフォルトドメインが常にインストールされます。このマニュアルでは、デフォルトドメインを使用します。

Application Server ソフトウェアは、管理コンソールに加えて、管理機能を実行するためのさまざまなコマンド行ツールを提供しています。ツールを起動するには、コマンドウィンドウでツールの名前を入力します。表 1-1 で、ツールの名前(左の列)順に各ツールの説明(右の列)を示します。

表 1-1 コマンド行ツール

ツールの名前	説明
appclient	アプリケーションクライアントコンテナを起動し、アプリケーション JAR (Java アーカイブ) ファイルにパッケージ化されたクライアントアプリケーションを呼び出します。
asadmin	Application Server ソフトウェアを設定するためのサブコマンドのセットを提供する Application Server 管理ツールを起動します。
asant	Jakarta Ant ツールを起動して、反復性のある開発および配備タスクを自動化できるようにします。
asapt	Java EE 注釈付きの Java ソースをコンパイルします。wsimport コマンドがこのツールから自動的に呼び出されます。
asupgrade	Application Server ソフトウェアをアップグレードするための Application Server 管理ツールです。
capture-schema	データベースからスキーマ情報を抽出し、サーバーでコンテナ管理による持続 (CMP) に利用できるスキーマファイルを生成します。
jspc	JSP ページをコンパイルします。
package-appclient	アプリケーションクライアントのコンテナライブラリおよび JAR ファイルをパッケージ化します。
schemagen	Java クラスで参照する名前空間ごとにスキーマファイルを作成します。
verifier	DTD を使用して Java EE 配備記述子を検証します。 このツールには、グラフィカルユーザーインターフェースもあります。GUI を表示するには、-u オプションを指定します。 一部の Windows システムでは、同じ名前のドライババリエーションユーティリティを起動します。Application Server の verifier を起動するには、 <i>install-dir/bin</i> ディレクトリから操作する必要があります。

表 1-1 コマンド行ツール (続き)

ツールの名前	説明
wscmpile	サービス定義インタフェースを取得して、JAX-RPC用のクライアントスタブまたはサーバー側スケルトンを生成します。または、指定したインタフェース用の WSDL (Web Services Description Language) 記述を生成します。
wsdeploy	JAX-RPCを使用する Web サービスアプリケーションの、実装に固有ですぐに配備できる WAR ファイルを生成します。
wsgen	Web サービスエンドポイントクラスを読み取り、Web サービスの配備と呼び出しに必要なすべてのアーティファクトを生成します。
wsimport	SEI (サービスエンドポイントインタフェース)、サービス、 <code>wSDL:fault</code> および <code>soap:headerfault</code> タグからマップされた例外クラス、 <code>wSDL:message</code> タグから派生した非同期応答 Bean、JAXB 生成値型などの JAX-WS ポータブルアーティファクトを生成します。
xjc	ソース XML スキーマを Java プログラミング言語の JAXB コンテンツクラスの集合に変換またはバインドします。

注 -

- Windows でこれらのコマンド行ツールを実行するには、PATH と呼ばれる環境変数が `install-dir/bin/` ディレクトリを指していることを確認します。
- `asadmin` サブコマンドを実行するには、すべてのサブコマンドに `asadmin` を接頭辞として付けるか、`install-dir/bin/` ディレクトリに `asadmin` プロンプトを入力 (`asadmin` と入力して Return キーを押す) します。

サーバーの起動

ここでは、ドメイン管理サーバー (DAS) の起動、管理コンソールへのログイン、ログファイルの確認、ノードエージェントの起動などの基本的なタスクについて説明します。手順は、それらを完了させるために必要な順序で示されます。

- 16 ページの「Solaris および Linux でドメインを起動するには」
- 16 ページの「Windows でドメインを起動するには」
- 18 ページの「管理コンソールにログオンするには」
- 19 ページの「ドメイン管理サーバーのログファイルを表示するには」
- 21 ページの「ノードエージェントを起動するには」
- 22 ページの「ドメインにクラスタサポートを追加する」
- 23 ページの「バンドルされた Java DB サーバーを起動するには」

▼ Solaris および Linux でドメインを起動するには

ドメイン管理サーバーを起動するには、デフォルトドメイン `domain1` を起動します。

- 1 PATH 環境変数に、`install-dir/bin/` ディレクトリを追加します。
- 2 次のコマンドを入力してサーバーを起動します。

```
asadmin start-domain domain1
```

ユーザー、管理パスワードおよびマスターパスワードの入力を求められた場合は、インストール時に使用したユーザー名とパスワードを入力します。

起動プロセスが完了すると、次のメッセージが表示されます。

ドメイン `domain1` が起動しました。

各ドメインには、エンタープライズ、クラスタ、開発者という、それぞれに対応するプロファイルがあります。`domain1` はデフォルトプロファイルで作成されます。このデフォルトプロファイルは、`asadminenv.conf` ファイルで定義される

`AS_ADMIN_PROFILE` 変数で定義されます。3つのプロファイルの詳細については、

『Sun Java System Application Server 9.1 管理ガイド』の「使用法プロファイル」を参照してください。`domain1` が開発者プロファイルで作成された場合、管理コンソールにログインしたときに、クラスタリングやノードエージェントといった機能は使用できません。クラスタやサーバーインスタンスを使用するには、`create-domain` コマンドを使用してクラスタプロファイルのドメインを作成します。ドメインの作成方法については、`asadmin create-domain --help` と入力するか、『Sun Java System Application Server 9.1 Reference Manual』にある `create-domain(1)` を参照してください。

▼ Windows でドメインを起動するには

ドメイン管理サーバーを起動するには、デフォルトドメイン `domain1` を起動します。

- 1 エクスプローラウィンドウまたはデスクトップで「マイコンピュータ」を右クリックします。
- 2 「プロパティ」を選択して「システムのプロパティ」ダイアログを表示します。
- 3 「詳細設定」タブをクリックします。
- 4 「環境変数」をクリックします。

- 5 「ユーザー環境変数」領域で次の操作を行います。
 - PATH変数が存在する場合は、次のように `install-dir\bin` がパス内に存在していることを確認します。 `install-dir\bin; other_entriesinstall-dir\bin` が PATH 変数内に存在していない場合は、追加します。
 - PATH変数が存在していない場合は、「新規」をクリックします。「変数名」に PATH と入力します。「変数値」にサーバーの bin ディレクトリのパスを入力します。 `install-dir \bin` 「OK」をクリックして変更を適用します。
- 6 新しい環境変数 `AS_ADMIN_USER` を追加して、インストール時に割り当てた管理ユーザー名にこの変数を設定します。
- 7 「OK」をクリックして変更を確定し、開いている残りのウィンドウを閉じます。
- 8 「プログラム」メニューの「管理サーバーの起動」をクリックして、**Application Server** を起動します。
- 9 コマンドプロンプトウィンドウが開き、管理パスワードおよびマスターパスワードの入力を求められた場合は、インストール時に使用したパスワードを入力します。サーバーが起動していることを示す次のメッセージがウィンドウに表示されます。

ドメイン `domain1` を起動しています。お待ちください。 ログは `domain_dir\domain1\logs\server.log` にリダイレクトされます。

起動プロセスが完了すると、次のようなメッセージが追加で表示されます。

ドメイン `domain1` が起動しました。 ... 続行するには何かキーを押してください...

各ドメインには、エンタープライズ、クラスタ、開発者という、それぞれに対応するプロファイルがあります。`domain1` はデフォルトプロファイルで作成されます。このデフォルトプロファイルは、`asadminenv.conf` ファイルで定義される `AS_ADMIN_PROFILE` 変数で定義されます。プロファイルの詳細については、『Sun Java System Application Server 9.1 管理ガイド』の「使用法プロファイル」を参照してください。`domain1` が開発者プロファイルで作成された場合、管理コンソールにログインしたときに、クラスタリングやノードエージェントといった機能は使用できません。クラスタやサーバーインスタンスを使用するには、`create-domain` コマンドを使用してクラスタプロファイルのドメインを作成します。ドメインの作成方法については、`asadmin create-domain --help` と入力するか、『Sun Java System Application Server 9.1 Reference Manual』にある `create-domain(1)` を参照してください。
- 10 任意のキーを押してメッセージウィンドウを閉じます。

▼ 管理コンソールにログオンするには

管理コンソールは、さまざまな管理作業および設定作業を単純化するブラウザインタフェースです。通常、管理コンソールは次の作業に使用されます。

- アプリケーションの配備および配備解除
- アプリケーションの有効化、無効化、および管理
- リソースおよびその他のサーバー設定の構成
- ログファイルの選択と表示

管理コンソールの使用に関する詳細は、オンラインヘルプまたは『Sun Java System Application Server 9.1 管理ガイド』を参照してください。

1 ブラウザで次の URL を入力します:

`http://localhost:4848`

ブラウザがサーバーと異なるシステムにある場合は、`localhost` 変数を、ドメイン管理サーバーの実行されているシステム名に置き換えます。

4848 は管理コンソールのデフォルトのポート番号です。インストール時にポート番号を変更した場合や、別のポート番号でドメインを作成した場合は、その番号を代わりに使用します。

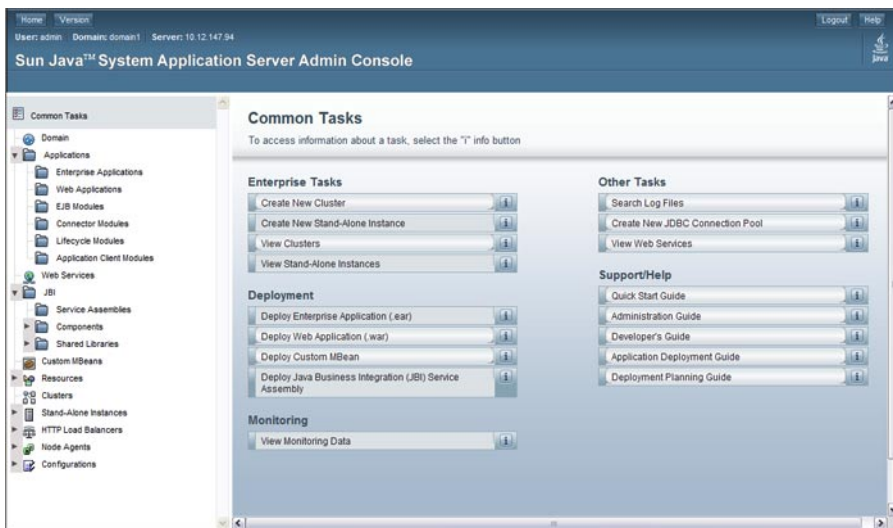
注 - 「不明な認証局により認証された Web サイト」などのメッセージを示すポップアップウィンドウが表示された場合は、「OK」をクリックします。

このメッセージが表示されるのは、セキュア伝送プロトコルで、管理コンソールのサービスを提供するために使用するドメイン管理サーバーが使用する自己署名の証明書をユーザーのブラウザが認識しないためです。

2 ログインウィンドウが表示されたら、管理ユーザー名とパスワードを入力します。

3 「ログイン」をクリックします。

デフォルトのユーザー名は `admin` で、パスワードは `adminadmin` です。

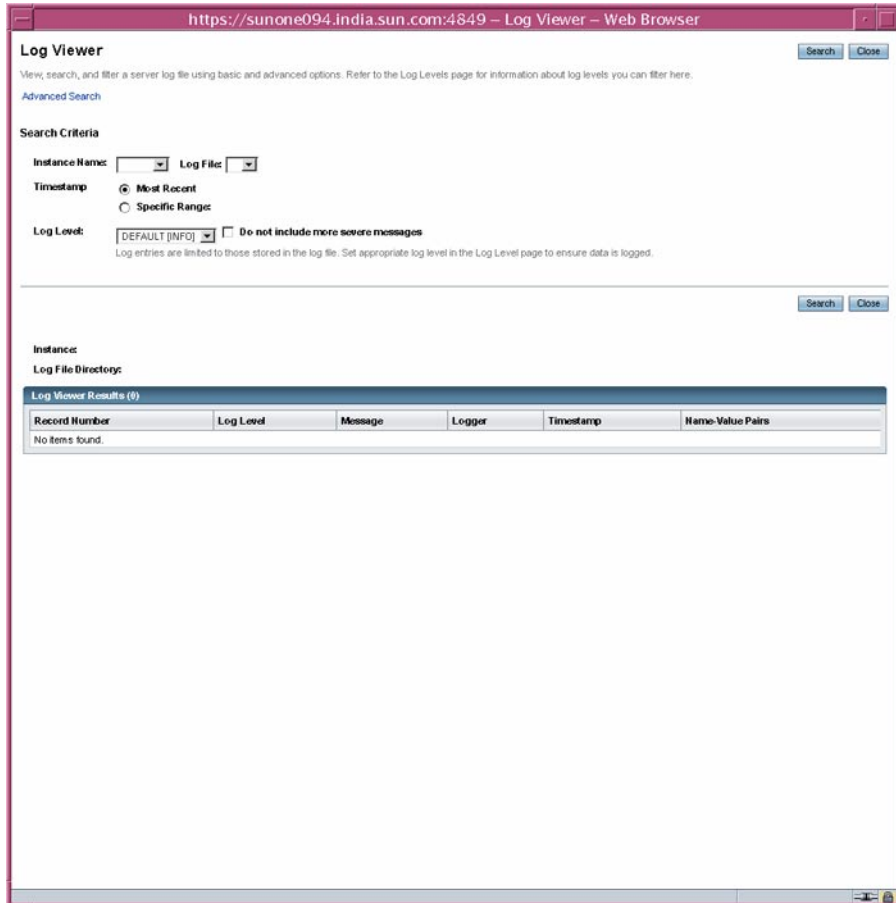


左側の区画では、用意されたツリーを使用して、管理する項目を選択します。右側の区画では、「共通操作」見出しの下に、さまざまな管理タスクが表示されています。

▼ ドメイン管理サーバーのログファイルを表示するには

Application Server インスタンスとドメイン管理サーバーは、注釈の付いたログをファイルシステム上に作成します。デフォルトでは、すべてのエラー、警告、また有用な通知メッセージがログとして記録されます。

- 1 右側の区画の「共通操作」から、「ログファイルを検索します」をクリックして、ログビューア用の新しいブラウザウィンドウを起動します。



- 2 ログビューアウィンドウで、「インスタンス名」ドロップダウンリストから「server」を選択し、「検索」をクリックします。
ドメイン管理サーバーの最近のログファイルエントリが表示されます。
- 3 メッセージをスキャンして、サーバーの起動時に問題が検出されたことを示す「WARNING」メッセージ、または「SEVERE」メッセージを探します。
ログビューアは、いつでも閉じることができます。クラスタを作成し、アプリケーションを配備したあとで、いずれかの操作に失敗していないかどうかログファイルを調べます。ログビューアを使用して、ドメイン内で実行されている任意の Application Server インスタンスのログファイルを表示します。

▼ ノードエージェントを起動するには

「ノードエージェント」は、Application Server 管理ドメインに含まれる各マシン上で実行される軽量プロセスです。ノードエージェントは、ホスト上のサーバーインスタンスの起動および停止を管理します。また、ドメイン管理サーバーとやり取りして、新しい Application Server インスタンスを作成します。

注-ノードエージェントは、クラスタプロファイルまたはエンタープライズプロファイルのドメイン上でのみ使用できます。開発者ドメインのみを実行している場合は、ノードエージェントを作成して使用する前に、クラスタプロファイルでドメインを作成する必要があります。ドメインの作成方法については、`asadmin create-domain --help` と入力するか、『Sun Java System Application Server 9.1 Reference Manual』の `create-domain(1)` を参照してください。

マシンが属する各 Application Server の管理ドメイン用として、1 台のマシンに1つのノードエージェントが必要です。Sun Java System Application Server 9.1 (Enterprise バンドル) を使用している場合は、インストール時にノードエージェントコンポーネントを選択すれば、ホスト名でデフォルトのノードエージェントが作成されます。Glassfish をインストールした場合、ノードエージェントは自動的に作成されません。ノードエージェントを作成する場合は、『Sun Java System Application Server 9.1 高可用性 (HA) 管理ガイド』の「ノードエージェントを作成するには」の手順を参照してください。

- 1 管理コンソールでノードエージェントが起動していないことを確認し、端末ウィンドウで、次のコマンドを入力します。

```
asadmin start-node-agent nodeagentname
```

nodeagentname 変数を、作成したノードエージェントの名前に置き換えます。Sun Java System Application Server 9.1 によって作成されるデフォルトのノードエージェントを使用する場合は、*nodeagentname* を、Application Server の実行されているホスト名に置き換えます。ポート 4848 のデフォルトドメイン (`domain1`) がクラスタプロファイルまたはエンタープライズプロファイルで実行されていない場合は、`--port` オプションを使用してクラスタまたはエンタープライズドメインのポートを指定します。

- 2 パスワードの入力を求められた場合は、マスターパスワードを入力します。ノードエージェントが起動し、ドメイン管理サーバーに接続します。

注- ノードエージェントを起動する前に、ドメイン管理サーバーが実行されていることを確認します。クラスタインスタンスの `server.log` ファイルを確認して、ノードエージェントやクラスタインスタンスの起動に関連する問題を特定します。

Windows 上でノードエージェントまたはクラスタを起動するときに問題が発生した場合は、『Sun Java System Application Server 9.1 リリースノート』を参照して、問題に対する解決策を検討してください。

ノードエージェントが起動したら、管理コンソールに表示することができます。

次の手順 ここでは、ドメイン管理サーバーを起動し、サーバーが正常に実行されていることを確認しました。また、管理コンソールにログインし、ログビューアを使用しました。ノードエージェントを起動しました。ここでクイックスタートを終了してもかまいませんし、次のセクションに進むこともできます。

▼ ドメインにクラスタサポートを追加する

管理コンソールを使用して、開発者ドメインにクラスタサポートを追加することができます。

- 1 管理コンソールにログオンします。ログオンの方法については、[18 ページの「管理コンソールにログオンするには」](#)を参照してください。
- 2 左側の区画で **Application Server** をクリックします。
- 3 「一般」タブで「クラスタサポートを追加」をクリックします。
- 4 「クラスタサポートを追加」ページが表示されます。このページの情報を確認して「OK」をクリックします。
- 5 「再起動が必要です」ページで「インスタンスの停止」ボタンをクリックして、ドメインを停止します。
- 6 ドメインのインストールされているマシンにログインして、次のコマンドを実行します。`install-dir/bin/asadmin list-domains`。ドメインが停止したことを確認します。
- 7 次のコマンドを実行します。`install-dir/bin/asadmin start-domain <domain-name>`再起動されたドメインはクラスタ化をサポートしています。

参考 障害追跡

再起動されたドメインにクラスタ化機能が表示されない場合は、ドメインを停止して、ドメインのインストールされたマシンを再起動し、手順7に示されたようにドメインを再起動します。

▼ バンドルされた Java DB サーバーを起動するには

Application Server には、バンドル版の Java DB サーバーが含まれています。起動するには次の手順に従います。

- 1 データベースをインストールしたユーザーとしてログインしていることを確認します。
- 2 **asadmin start-database** を実行します。
start-database コマンドの詳細については、コマンドのヘルプまたは『Sun Java System Application Server 9.1 Reference Manual』にある start-database(1) の情報を参照してください。

次の学習ステップ

『クイックスタートガイド』の次の学習ステップは、次の章に示されています。

- **第2章 クラスタの設定**
サンプルのクラスタを作成するための手順が示されています。開発者プロファイルでドメインを実行している場合は、この章を飛ばして次の章に進んでください。
- **第3章 アプリケーションの配備**
サンプルの Web アプリケーションを配備するための手順が示されています。クラスタ機能を備えたドメインを使用している場合は、エンタープライズアプリケーションをクラスタに配備するための手順を実行できます。
- **第4章 ロードバランサの操作**
ロードバランサを作成して使用するための手順が示されています。ロードバランサプラグインと Web サーバーのインストールが必要です。

Application Server の学習や使用に他のリソースも利用できます。以下にその内容を示します。

- *install-dir/docs-ee/about.html* にある製品詳細情報
- *install-dir/samples* にあるサンプルアプリケーション
- 『Sun Java System Application Server 9.1 リリースノート』
- 『Sun Java System Application Server 9.1 管理ガイド』

- 『Sun Java System Application Server 9.1 Reference Manual』 (マニュアルページ)
- 『Sun Java System Application Server 9.1 Application Deployment Guide』
- 『Sun Java System Application Server 9.1 高可用性 (HA) 管理ガイド』
- 『The Java EE 5 Tutorial』
- [Java BluePrints \(http://java.sun.com/blueprints\)](http://java.sun.com/blueprints) ガイドライン (Enterprise 向け)

クラスタの設定

この章では、クラスタの作成および設定方法について説明します。「クラスタ」とは、同じ設定、リソース、アプリケーションを共有するサーバーインスタンス (通常は複数のホスト上にある) をグループ化したものです。クラスタにより、サーバーインスタンス間のロードバランスが保たれ、フェイルオーバーを通じて処理が継続されることで、高可用性を実現します。複数のマシンにまたがるクラスタを作成し、各マシン上で、ノードエージェントを用いてこれらのクラスタを管理することができます。

この章では、クラスタまたはエンタープライズプロファイルでドメインを実行していることを前提とします。この章で説明する内容は、次のとおりです。

- [25 ページの「クラスタの作成」](#)
- [27 ページの「クラスタ化されたサーバーインスタンスのポートの表示」](#)

クラスタの作成

このセクションでは、2つの Application Server インスタンスを含むクラスタの作成方法について説明します。

▼ 管理コンソールを使用してクラスタを作成するには

説明を簡単にするために、クラスタは完全に1つのマシン内で実行されているものとして示す手順では、DASが稼動しているのと同じホスト上にサンプルのクラスタを作成します。

始める前に [21 ページの「ノードエージェントを起動するには」](#) の説明に従って、あらかじめノードエージェントを起動しておく必要があります。クラスタの作成時にインスタンスを指定する場合は、インスタンスを実行するマシンで、実行しているノード

エージェントと指定するインスタンスを関連付けます。ノードエージェントが実行されていない場合、インスタンスは起動しません。ノードエージェントとインスタンスの名前は、1つのドメインで作成されるクラスタ間で一意のものとする必要があります。

- 1 まだログインしていない場合は、管理コンソール(<http://localhost:portnumber>)にログインします。
localhost 変数を、ドメイン管理サーバーを実行するシステム名に置き換えます。
portnumber を、クラスタまたはエンタープライズドメインのポート番号に置き換えます。
- 2 「共通操作」で、「新しいクラスタを作成します」をクリックし、「新しいクラスタ」ページを表示します。
- 3 新しいクラスタの名前として、`cluster1`と入力します。
- 4 利用可能な設定テンプレートのドロップダウンリストから **default-config** 設定を選択して、「選択している設定のコピーを作成します」を選択します。
- 5 「作成するサーバーインスタンス」の下の「新規」ボタンを2回クリックして、クラスタ用の2つのインスタンスを指定する2つのエントリを作成します。
- 6 インスタンス名として「**instance1**」および「**instance2**」を入力します。
または、作成して起動したノードエージェントの名前を指定することもできます。
ノードエージェントを作成して起動するには、`asadmin create-node-agent(1)` コマンドを使用します。
- 7 「了解」をクリックします。
「クラスタが正しく作成されました」ページが表示され、左側の区画のツリーに `cluster1` が表示されます。このクラスタ用に、設定テンプレート `default-config` のコピーが作成され、`cluster1-config` という名前が割り当てられます。
- 8 `cluster1` クラスタを起動します。

▼ CLI コマンドを使用してクラスタを作成するには

- 1 次の `asadmin` コマンドを実行することで、`cluster1` という名前のクラスタを作成します。
`asadmin create-cluster --user admin --passwordfile adminpassword.txt --host localhost -port 4848 cluster1`

- 2 次のコマンドを実行することで、cluster1-nodeagent という名前のノードエージェントを作成します。`asadmin create-node-agent --user admin --passwordfile adminpassword.txt --host localhost --port 4848 cluster1-nodeagent`
- 3 次のコマンドを実行することで、このノードエージェントを起動します。`asadmin start-node-agent --user admin --passwordfile adminpassword.txt --host localhost --port 4848 cluster1-nodeagent`
- 4 次のコマンドを実行することで、クラスタの下に2つのインスタンスを作成します。
 - `asadmin create-instance --user admin --passwordfile adminpassword.txt --host localhost -port 4848 --cluster cluster1 --nodeagent cluster1-nodeagent --systemproperties "JMX_SYSTEM_CONNECTOR_PORT=8687:IIOP_LISTENER_PORT=3330:IIOP_SSL_LISTENER_PORT=4444:MUTUALAUTH_PORT=5550:HTTP_LISTENER_PORT=1110:HTTP_SSL_LISTENER_PORT=2220" instance1`
 - `asadmin create-instance --user admin --passwordfile adminpassword.txt --host localhost -port 4848 --cluster cluster1 --nodeagent cluster1-nodeagent --systemproperties "JMX_SYSTEM_CONNECTOR_PORT=8688:IIOP_LISTENER_PORT=3331:IIOP_SSL_LISTENER_PORT=4444:MUTUALAUTH_PORT=5551:HTTP_LISTENER_PORT=1111:HTTP_SSL_LISTENER_PORT=2221" instance2`
- 5 次のコマンドを実行することで、クラスタを起動します。`asadmin start-cluster --user admin --passwordfile adminpassword.txt --host localhost --port 4848 cluster1`

クラスタの作成および設定方法については、『Sun Java System Application Server 9.1 高可用性 (HA) 管理ガイド』の第6章「Application Server クラスタの使用」を参照してください。

ノードエージェントの作成および設定方法については、『Sun Java System Application Server 9.1 高可用性 (HA) 管理ガイド』の第8章「ノードエージェントの設定」を参照してください。

クラスタ化されたサーバーインスタンスのポートの表示

▼ クラスタ化されたサーバーインスタンスのポートを表示するには

前のセクション(25 ページの「クラスタの作成」)で作成したクラスタ化されたサーバーインスタンスのポート番号を表示して、デフォルト値を上書きすることができます

ます。デフォルトでは、instance1 の HTTP ポートは 38081 で、instance2 の HTTP ポートは 38080 です。クラスタの作成時に、これらのポートがビジー状態の場合、あるいはこれらのポートが他のインスタンスまたはクラスタに割り当てられている場合は、異なるポート番号が割り当てられます。

ポート番号を表示して、オプションでデフォルト値を上書きするには、次の手順を実行します。

- 1 左側の区画でクラスタを展開し、cluster1 をクリックして、クラスタの「一般情報」ページを表示します。
- 2 「インスタンス」タブをクリックして、作成したインスタンス instance1 と instance2 を表示します。
 - a. instance1 をクリックして、このインスタンスを調べます。
「HTTP ポート」フィールドに、インスタンスに関連付けられた HTTP ポートと HTTPS ポートが表示されます。
 - b. instance2 に対して、同じ手順を繰り返します。

次の手順 ここでは、1つのマシンで、1つのクラスタを作成しました。各マシンにソフトウェアをインストールし、ノードエージェントを実行している限り、同じ基本手順を使用して、複数のマシンにまたがるクラスタを作成することもできます。

アプリケーションの配備

この章では、アプリケーションの配備方法について説明します。開発者プロファイルでドメインを実行している場合は、29ページの「サンプル Web アプリケーションの配備」を参照してください。クラスタプロファイルでドメインを実行している場合は、32ページの「サンプルのエンタープライズアプリケーションのクラスタへの配備」を参照してください。

この章で説明する内容は次のとおりです。

- 29ページの「サンプル Web アプリケーションの配備」
- 32ページの「サンプルのエンタープライズアプリケーションのクラスタへの配備」
- 33ページの「クラスタ内でインスタンスを起動するには」
- 34ページの「アプリケーションの配備を確認するには」

サンプル Web アプリケーションの配備

ここに示す手順は、開発者プロファイルでドメインを実行していることを前提としています。

▼ 自動配備を使用して Hello アプリケーションを配備するには

アプリケーションサーバーのインストールファイルには『The Java EE 5 Tutorial』で提供されている、「Hello」と表示するパッケージ化済みアプリケーションが含まれています。Hello アプリケーションは、samples/quickstart ディレクトリの WAR (Web ARchive) ファイルに格納されています。パッケージ化アプリケーションは、JAR (Java ARchive) または EAR (Enterprise ARchive) ファイルの形式にすることもできます。

あるいは、このアプリケーションを

<https://glassfish.dev.java.net/downloads/quickstart/hello.war> からダウンロードすることもできます。

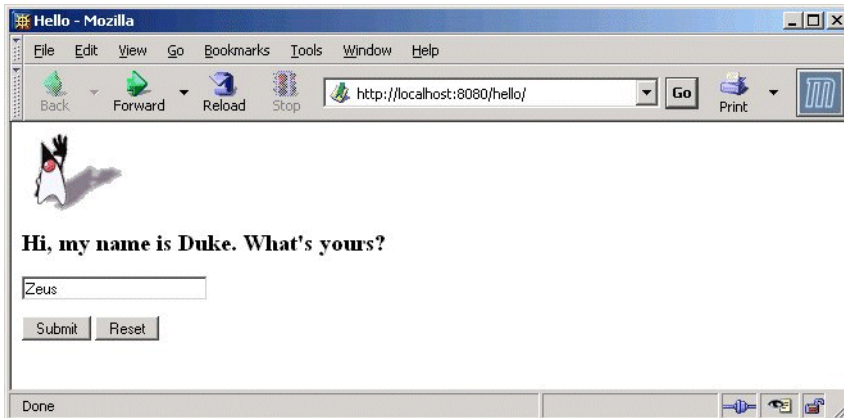
ドメインの `autodeploy` ディレクトリにコピーすると、パッケージ化済みアプリケーションを簡単な操作ですぐに使用できます。

- 1 `install-dir/samples/quickstart/hello.war` を探す
か、<https://glassfish.dev.java.net/downloads/quickstart/hello.war> からダウンロードします。
- 2 `hello.war` を `install-dir/domains/domain1/autodeploy/` ディレクトリにコピーします。

ヒント - Windows では、`autodeploy` ディレクトリへのショートカットをデスクトップに作成し、`hello.war` ファイルをそのショートカットにドラッグ&ドロップできます。

アプリケーションの配備が完了すると、サーバーは `autodeploy` ディレクトリに `hello.war_deployed` という名前のファイルを作成します。システムの性能によっては、この処理に数秒かかることがあります。このファイルが表示される前にアプリケーションページに移動しようとする、404-File Not Found エラーが発生します。

- 3 次の URL にアクセスしてアプリケーションページに移動します。 `http://localhost:8080/hello`
アプリケーションの最初のページで名前を入力を要求されます。



- 4 名前を入力して「Submit」をクリックします。
アプリケーションによって、入力内容に合わせてカスタマイズされた Hello という応答が表示されます。

▼ 管理コンソールを使用してHelloアプリケーションを配備するには

- 1 Webブラウザで `http://localhost:4848/asadmin` を開きます。
 - `localhost` は、ブラウザと Application Server が両方とも同じシステム上で実行されている場合に使用します。Application Server が別のシステムで実行されている場合は、そのシステムの名前を URL に代入します。
 - 4848 は管理コンソールのデフォルトのポート番号です。インストール時にポート番号を変更した場合は、その番号を使用してください。
- 2 管理ユーザー名とパスワードを入力します。
- 3 左の区画で「アプリケーション」ノードをクリックして展開します。
- 4 「Webアプリケーション」をクリックします。
配備済みのWebアプリケーションのリストが表示されます。
- 5 Helloアプリケーションをすでに配備している場合は、その横のチェックボックスを選択して「配備取消し」をクリックし、アプリケーションの配備を取り消します。
- 6 「配備」をクリックします。
- 7 Application Server にバンドルされているHelloアプリケーションを使用するか、<https://glassfish.dev.java.net/downloads/quickstart/hello.war> からダウンロードしたサンプルをアップロードすることができます。次のいずれかを行います。
 - 「Application Server からアクセス可能なローカルのパッケージファイルまたはディレクトリ」というオプションを選択して、「ファイルを参照」をクリックします。`install-dir /samples/quickstart` ディレクトリに移動して、`hello.war` を選択します。
- 8 Helloアプリケーションをダウンロードした場合は、「サーバーにアップロードされるパッケージファイル」オプションを選択し、「参照」をクリックします。アプリケーションをダウンロードしたディレクトリに移動します。
- 9 「了解」をクリックします。
アプリケーションが「Webアプリケーション」リストに表示されます。
- 10 アプリケーションが正しく配備されたことを確認するため、「Webアプリケーション」ページで、Helloアプリケーションを選択して「起動」をクリックします。

参考 CLI コマンドを使用したアプリケーションの配備

asadmin deploy コマンドを使用してアプリケーションを配備することができます。

『Sun Java System Application Server 9.1 Reference Manual』の deploy(1) を参照してください。asant コマンドを使用して配備することもできます。アプリケーションの配備方法については、『Sun Java System Application Server 9.1 Application Deployment Guide』の「Deploying Modules and Applications」を参照してください。

サンプルのエンタープライズアプリケーションのクラスタへの配備

ここに示す手順は、クラスタプロファイルでドメインを実行していることを前提としています。この章で示す手順は、[第2章クラスタの設定](#)で説明したとおりに cluster1 という名前のクラスタが作成済みであることを前提としています。このセクションでは、clusterjsp というサンプルアプリケーションを使用して HTTP のロードバランス機能を説明します。

▼ 管理コンソールを使用してサンプルのエンタープライズアプリケーションを配備するには

- 1 「共通操作」ページに移動していない場合は、「ホーム」ボタンをクリックするか、左側の区画の「共通操作」をクリックして、このページに移動します。
- 2 右側の区画の「共通操作」の下で、「エンタープライズアプリケーション(.ear)を配備」をクリックします。
- 3 ローカルマシンに **Application Server** がインストールされている場合は、「サーバーにアップロードされるパッケージファイル」テキストボックスで「参照」をクリックして、`install-dir`
`/samples/ee-samples/highavailability/apps/clusterjsp/clusterjsp.ear` に移動します。別のマシンにインストールされている **Application Server** を使用して、管理コンソールにリモートでアクセスしている場合は、「**Application Server** からアクセス可能なローカルのパッケージファイルまたはディレクトリ」テキストボックスで「参照」をクリックします。`install-dir /samples/quickstart/clusterjsp/clusterjsp.ear` に移動します。
- 4 ページ内の「ターゲット」領域で、「利用可能なターゲット」リストから cluster1 を選択して、「追加」をクリックして「選択されたターゲット」リストに移動します。

- 5 「了解」をクリックします。

`clusterjsp` アプリケーションが `cluster1` に配備されました。このアプリケーションを起動するには、まずクラスタ内でインスタンスを起動する必要があります。[33 ページの「クラスタ内でインスタンスを起動するには」](#) を参照してください。

参考 CLI コマンドを使用したアプリケーションの配備

`asadmin deploy` コマンドを使用してアプリケーションを配備することができます。

『Sun Java System Application Server 9.1 Reference Manual』の `deploy(1)` を参照してください。`asant` コマンドを使用して配備することもできます。アプリケーションの配備方法については、『Sun Java System Application Server 9.1 Application Deployment Guide』の「Deploying Modules and Applications」を参照してください。

▼ クラスタ内でインスタンスを起動するには

配備されたアプリケーションを起動する前に、クラスタ内でインスタンスを起動する必要があります。

- 1 左側の区画のツリーで「クラスタ」をクリックして、`cluster1` ノードをクリックします。
- 2 右側の区画で、「一般」タブがアクティブになっていない場合は、このタブに移動します。
- 3 インスタンスがまだ稼働中でない場合は、「クラスタの起動」ボタンがアクティブになっています。このボタンをクリックして、クラスタのインスタンスを起動します。
- 4 インスタンスが起動しているかどうかは「状態」フィールドで確認します。このフィールドには、稼働中のインスタンスの数が表示されます。
これでアプリケーションを起動することができます。[34 ページの「アプリケーションの配備を確認するには」](#) を参照してください。

参考 CLI コマンドを使用したインスタンスの起動

`asadmin start-instance` コマンドを使用してインスタンスを起動することができます。『Sun Java System Application Server 9.1 Reference Manual』を参照してください。

▼ アプリケーションの配備を確認するには

- 1 次の URL をブラウザに入力します。

`http://localhost:port/clusterjsp`

`localhost` 変数を、ドメイン管理サーバーを実行するシステム名に置き換えます。

`port` 変数を、`instance1` の HTTP-LISTENER-PORT の値に置き換えます。この例では、`http://localhost:38081/clusterjsp` を使用します。

- 2 セッションの属性データのいくつかを追加します。
- 3 セッションおよびホスト情報が次のとおり表示されるかどうかを確認します。次に例を示します。
 - Executed From Server: `localhost`
 - Server Port Number: `38081`
 - Executed Server IP Address: `198.19.255.255`
 - Session Created: `Tue Aug 23 15:26:07 PDT 2005`
- 4 セッションデータのいくつかを追加して、「セッションに追加」ボタンをクリックします。
- 5 インスタンス `instance2` に対して、この手順を繰り返します。このときブラウザには次の URL を入力します。

`http://localhost:38080/clusterjsp`

次の手順 おめでとうございます。これで『クイックスタートガイド』のこのセクションを修了しました。ここでクイックスタートを終了してもかまいませんし、次のロードバランスのセクションに進むこともできます。

ロードバランサの操作

ここでは、Web Server ソフトウェアを Application Server のクラスタに対するロードバランサとして機能するように設定する方法の手順について説明します。また、ロードバランサの設定手順や Web Server にエクスポートする手順についても説明します。ロードバランサ機能は、エンタープライズプロファイルかクラスタプロファイルでドメインを実行している場合のみ使用できます。

ロードバランサはクラスタとともに配備されます。ロードバランサには次のような機能があります。

- アプリケーションまたはサービスを、複数の物理 (または論理) ホストにまたがって水平方向にスケーリングすることに加え、単一の URL でユーザーを表すことができます。
- セッションの持続性を使用することにより、ホストの障害またはサーバーのクラッシュからユーザーを保護します。
- ユーザーから内部ネットワークを見えないようにして、セキュリティを強化します。

Application Server には、Sun Java™ System Web Server、Apache、および Microsoft Windows IIS などの一般的な Web サーバー用のロードバランサプラグインが含まれています。

このセクションを完了するには、このマニュアルの手順に従って今までに作成したドメイン管理サーバーと 2 つの Application Server のインスタンスに加え、Web Server を実行するための十分なメモリーがシステムに必要です。512M バイト～1024M バイトのメモリーをシステムに装備することをお勧めします。

このトピックでは、次の手順について説明します。

- [36 ページの「ロードバランスの設定」](#)
- [37 ページの「ロードバランスの起動」](#)
- [37 ページの「ロードバランスの検証」](#)

- 39 ページの「インメモリーレプリケーション機能を使用した高可用性とフェイルオーバー」

ロードバランスの設定

ロードバランスを設定する前に、ロードバランサプラグインをインストールする必要があります。ロードバランサプラグインのインストール手順については、『Sun Java System Application Server 9.1 Installation Guide』の「To Install the Load Balancing Plug-in (standalone)」を参照してください。ここでは、クラスタまたはエンタープライズプロファイルでドメインを実行していることを前提とします。

▼ ロードバランスを設定するには

- 1 管理コンソールを使用してロードバランサを作成します。または、`asadmin create-http-lb(1)` コマンドを使用することもできます。
 - a. 管理コンソールで「HTTP ロードバランサ」ノードをクリックします。
 - b. 「新規」をクリックします。
 - c. ロードバランサの名前として `lb1` と入力し、「変更を自動的に適用」チェックボックスをオンにします。このオプションを選択した場合は、ロードバランサの設定をエクスポートする必要がありません。ロードバランサの設定に対するすべての変更が、自動的に伝達されます。
 - d. **Web Server** のインストールされているホスト、**Web Server** インスタンスのポートを入力します。このサンプルシナリオでは、**Web Server** ホストは `localhost` で、ポートは **38000** です。
 - e. ターゲットとして `cluster1` を選択します。
サンプルのクラスタ (`cluster1`) の作成方法は、25 ページの「クラスタの作成」を参照してください。
 - f. 「了解」をクリックします。
- 2 `cluster1` のロードバランスを有効にします。

```
asadmin enable-http-lb-server cluster1
```
- 3 `clusterjsp` アプリケーションのロードバランスを有効にします。

```
asadmin enable-http-lb-application -name clusterjsp cluster1
```

参照 ロードバランサ設定の変更方法や診断プログラムの作成方法といった詳細なトピックについては、『Sun Java System Application Server 9.1 高可用性 (HA) 管理ガイド』の第5章「HTTP 負荷分散の設定」を参照してください。

ロードバランスの起動

Web Server を起動または再起動することでロードバランスを起動します。

- ロードバランサとして機能する Web Server インスタンスがまだ実行されていない場合は、Web Server を起動します。

Web Server 7.0 を使用している場合は、`wadm start-instance` コマンドを使用します。

Web Server 6.1 を使用している場合は、`<websvr-instance-dir>` ディレクトリで `start` スクリプトを実行します。

- ロードバランサとして機能する Web Server インスタンスがすでに実行されている場合は、Web Server を停止して再起動します。

Web Server 6.1 の場合は、`web_server_install_dir /https-hostname` で停止プログラムを使用し、起動プログラムを実行することでサーバーを再起動します。

Web Server 7.0 の場合は、`wadm stop-instance` を使用して、次に `wadm start-instance` コマンドを実行します。

ロードバランスの検証

アプリケーションを配備してロードバランサが実行されたら、ロードバランスが機能していることを検証します。

▼ ロードバランスを検証するには

- 1 `clusterjsp` アプリケーションの最初のページを表示するには、ブラウザで次の URL を入力します。

```
http://localhost:web_server_port/clusterjsp
```

`localhost` 変数を、Web Server を実行するシステム名に置き換えます。

`web_server_port` 変数を、`web_server_install_dir /https-hostname /config/server.xml` の LS 要素のポート属性の値に置き換えます。この例では、ポート 38000 を使用しています。

34 ページの「アプリケーションの配備を確認するには」で表示されたのと同じようなページが表示されます。

- 2 セッションおよびホスト情報が次のとおり表示されるかどうかを確認します。次に例を示します。
 - Executed From Server: *localhost*
 - Server Port Number: *38000*
 - Executed Server IP Address: *192.18.145.133*
 - Session Created: *Day Mon 05 14:55:34 PDT 2005*
- 3 **Server Port Number** は **38000**、つまり **Web Server** のポートです。ロードバランサは、クラスタの2つのインスタンスに要求を転送しました。
- 4 別のブラウザソフトウェア、または別のマシン上のブラウザを使用して、新しいセッションを作成します。同じブラウザからの要求が何度も続いて、同じインスタンスに進みます。

これらのセッションは、クラスタの2つのインスタンスに分散されている必要があります。次の場所にあるサーバーアクセスログファイルを調べれば確認できます。

 - Solaris Java Enterprise System インストールの場合

```
/var/opt/SUNWappserver/nodeagents/nodeagent_name
/instance1/logs/access/server_access_log

/var/opt/SUNWappserver/nodeagents/nodeagent_name
/instance2/logs/access/server_access_log
```
 - Linux Java Enterprise System インストールの場合

```
/var/opt/sun/appserver/nodeagents/
nodeagent_name/instance1/logs/access/server_access_log

/var/opt/sun/appserver/nodeagents/
nodeagent_name/instance2/logs/access/server_access_log
```
 - Windows Java Enterprise System インストールの場合

```
install-dir \nodeagents\nodeagent_name \instance1\logs\access\server_access_log
install-dir\nodeagents\nodeagent_name \instance1\logs\access\server_access_log
```
 - スタンドアロンの Application Server インストールの場合

```
install-dir /nodeagents/nodeagent_name/instance1/logs/access/server_access_log
install-dir/nodeagents/nodeagent_name /instance2/logs/access/server_access_log
```
- 5 HttpSession に格納する名前と値のペア (Name=Name Value=Duke) を追加します。
- 6 「セッションデータに追加」 ボタンをクリックします。
- 7 セッションデータが追加されたかどうかを確認します。

インメモリーレプリケーション機能を使用した高可用性とフェイルオーバー

GlassFish v2はHADBを提供していません。高可用性とフェイルオーバーのために、GlassFishではインメモリーレプリケーション機能が提供されています。この機能を実行するための手順を次に示します。

1. アプリケーションを配備する前にロードバランサプラグインをインストールしたWebサーバーを再起動します。これによって、`loadbalancer.xml`ファイルに設定されている順序で、インスタンスによって要求が処理されます。この章で提供される`loadbalancer.xml`ファイルを使用する場合、`instance1`が最初の要求を処理します。
2. すでに`clusterjsp` Webアプリケーションが配備済みなので、ここにセッションデータが格納されます。最初の要求を処理したのと同じインスタンスによって後続の要求が処理され、セッションデータが要求間で維持されていることを確認できるようにする必要があります。
3. いくつかの要求を送信し、これらの要求を処理したインスタンスをメモしてから、その特定のインスタンスをシャットダウンします。インスタンスを停止するには次のコマンドを使用します。

```
asadmin stop-instance --user adminuser --password adminpassword instance1
```
4. 次の要求を送信し、新しいデータが格納されていることと、これまでに追加されたデータもまだセッション内に存在することを確認します。要求を処理するサーバーのどれかが使用できない場合は、同じクラスタ内の別のサーバーが、それ以前のすべてのセッションとともにその要求を引き継ぎ、要求を完了させます。

高可用性フェイルオーバーの設定

前の章で使用した設定では、サーバーインスタンスがダウンした場合、ユーザーはセッション状態を失います。ここでは、2つの高度なトピックの2番目のものとして、高可用性データベース (HADB) のインストール、高可用性クラスタの作成、および HTTP セッション持続性のテストの手順について説明します。

GlassFish v2 は HADB を提供していません。高可用性とフェイルオーバーのために、GlassFish ではインメモリーレプリケーション機能が提供されています。詳細については、を参照してください。

Application Server は、HTTP セッション持続性とステートフルセッション Bean の持続性の両方をサポートしています。この章の手順では、インメモリーレプリケーションまたは HADB を使用して高可用性を実現しています。

次の手順では、すでにこの『クイックスタートガイド』の前の項に記載されている手順を実行済みであると仮定しています。手順は、それらを完了させるために必要な順序で示されます。HADB 機能を使用するには、エンタープライズプロファイルでドメインを実行する必要があります。

注 - このセクションの手順を完了するには、追加のハードウェアリソースが必要な場合があります。

このトピックは、次のセクションで構成されます。

- 42 ページの「高可用性クラスタと HADB」
- 42 ページの「HADB インストール前の手順」
- 44 ページの「HADB のインストール」
- 45 ページの「HADB の起動」
- 46 ページの「高可用性のためのクラスタおよびアプリケーションの設定」
- 46 ページの「クラスタの再起動」
- 47 ページの「HTTP セッションフェイルオーバーの検証」

高可用性クラスタと HADB

Sun Java System Application Server の高可用性クラスタは、状態の複製サービスを以前に作成したクラスタおよびロードバランサと統合し、その結果 HTTP セッションのフェイルオーバーが有効になります。

HttpSession オブジェクトとステートフルセッション Bean の状態は、HADB に格納されます。HADB は、セッション状態を格納するための高可用性データベースです。この水平的で拡張可能な状態管理サービスは、アプリケーションサーバー層とは無関係に管理できます。これは、ロードバランス、フェイルオーバー、および状態復元機能を含む、最大 99.999% のサービスとデータ可用性をサポートするように設計されています。

状態管理の機能を Application Server と切り離しておくことには、大きな利点があります。Application Server インスタンスは、状態レプリケーションを外部の高可用性状態サービスに委任する、スケーラブルで高性能な Java™ Platform, Enterprise Edition 5 (Java EE™ 5 プラットフォーム) コンテナとしての動作に CPU サイクルを消費します。この疎結合アーキテクチャーのため、アプリケーションサーバーインスタンスはクラスタとの間で簡単に追加または削除することができます。HADB の状態レプリケーションサービスを独立に拡張して、最適な可用性とパフォーマンスを得ることができます。アプリケーションサーバーインスタンスがレプリケーションも実行していると、Java EE アプリケーションのパフォーマンスが低下したり、ガベージコレクションの一時停止時間が長くなったりすることがあります。

各 HADB ノードは 512M バイトのメモリーを必要とするため、同じマシン上で 2 つの HADB ノードを実行するには、1G バイトのメモリーが必要です。メモリーが少ない場合は、各ノードを別のマシン上に設定してください。耐障害性がないため、配備の際に 1 台のホストのみで 2 ノードのデータベースを実行することはお勧めできません。

HADB インストール前の手順

この手順は、最も一般的なインストール前のタスクを取り扱います。HADB をインストールするための前提条件、ネットワーク冗長性の設定、ファイルシステムのサポートなど、その他のインストール前のトピックについては、『Sun Java System Application Server 9.1 高可用性 (HA) 管理ガイド』の第 2 章「高可用性 (HA) データベースのインストールと設定」を参照してください。

このセクションで推奨されるシステム設定値は、最大 6 個の HADB ノードを実行するために十分なものであり、共有メモリーを使用するシステム上のその他のアプリケーションについては考慮されていません。

▼ HADB に対応するようにシステムを設定するには

- 1 root ユーザーとしてアクセスします。
- 2 共有メモリーおよびセマフォーに関連する変数を定義します。

■ Solaris の場合:

- a. /etc/system ファイルに次の行を追加します。次の行がコメントとしてファイル内にある場合は、それらのコメントを解除し、値が次の値と一致していることを確認します。

```
set shmsys:shminfo_shmmax=0x80000000
set shmsys:shminfo_shmseg=36
set semsys:seminfo_semmnu=600
```

shminfo_shmmax をシステムの合計メモリーに設定します (16 進数の表記では、示されている値 0x80000000 が 2G バイトのメモリーを表す)。

seminfo_* 変数がすでに定義されている場合は、示されている容量でそれらを増分します。seminfo_semmni と seminfo_semmns のデフォルト値は変更する必要はありません。shminfo_shmseg 変数は Solaris 8 以降は使用されなくなりました。

- b. 次のコマンドを使用して再起動します。

```
sync; sync; reboot
```

■ Linux の場合:

- a. /etc/sysctl.conf ファイルに次の行を追加します。次の行がコメントとしてファイル内にある場合は、それらのコメントを解除します。マシン上の物理メモリー量に値を設定します。この値は、10 進数のバイト数で指定します。たとえば、物理メモリーが 2G バイトのマシンの場合は、次のように指定します。

```
echo 2147483648 > /proc/sys/shmmax
echo 2147483648 > /proc/sys/shmall
```

- b. 次のコマンドを使用して再起動します。

```
sync; sync; reboot
```

- Windows の場合: システム設定は特に必要ありません。

- 3 スタンドアロンの **Application Server** をインストールするときに既存の **JDK** ソフトウェアを使用した場合は、**JDKバージョン**を確認します。
HADB では Sun JDK 1.4.1_03 以上が必要です (JDKバージョンの最新情報については、『Sun Java System Application Server 9.1 リリースノート』を参照)。インストールされているバージョンをチェックし、まだ行なっていない場合は、JDKがインストールされているディレクトリに `JAVA_HOME` 環境変数を設定します。
- 4 再起動後、必要に応じてドメイン、**Web Server**、およびノードエージェントを再起動します。
ドメインを再起動するには、`asadmin start-domain domain1` コマンドを使用します。
Web サーバーを再起動するには、`web_server_install_dir/https-hostname` で起動プログラムを実行します。
ノードエージェントを再起動するには、`asadmin start-node-agent hostname` コマンドを使用します。`hostname` 変数を、Application Server を実行するホスト名に置き換えます。

HADBのインストール

ここでは、高可用性データベース (HADB) のインストールの手順を説明します。

注 - Application Server マシンで高可用性データベースを実行する場合と、Application Server のインストール時に HADB をインストールした場合は、[45 ページの「HADB の起動」](#)に進みます。

2G バイトのメモリーと 1～2 個の CPU を持つマシンがある場合には、HADB コンポーネントを Application Server システムと同じマシン上にインストールできます。それ以外の場合は、追加のハードウェアを使用します。次に例を示します。

- それぞれが 512M バイトから 1G バイトのメモリーを持つ 2 つの 1 CPU システム
- 1G バイトから 2G バイトのメモリーを持つ 1 つの 1～2 CPU システム

▼ HADB をインストールするには

- 1 **Application Server** または **Java Enterprise System** インストーラを実行します。
- 2 **HADB** をインストールするためのオプションを選択します。
- 3 ホストでのインストールを完了します。

HADBの起動

ここでは、主に `ma-initd` スクリプトの実行による、HADB 管理エージェントの起動について説明します。本稼働配備の場合は、管理エージェントをサービスとして起動し、その可用性を保証します。詳細については、『Sun Java System Application Server 9.1 高可用性 (HA) 管理ガイド』の「HADB 管理エージェントの起動」を参照してください。

複数のホスト上で HADB ノードを含むデータベースを起動する場合は、それぞれのホスト上の管理エージェントを起動します。

▼ Solaris または Linux 上の Java Enterprise System インストールで HADB を起動するには

- 1 `/etc/init.d` ディレクトリに移ります。
`cd /etc/init.d`
- 2 次のコマンドを実行して、エージェントを起動します。
`./ma-initd start`

▼ Windows 上の Java Enterprise System インストールで HADB を起動するには

Sun Java System を設定および実行すると、デフォルトで HADB が起動します。ただし、手動で起動する必要がある場合は、次の手順に従います。

- 1 「スタート」⇒「設定」⇒「コントロールパネル」の順に選択して、「管理ツール」をダブルクリックします。
- 2 「サービス」ショートカットをダブルクリックします。
- 3 「サービス」リストで **HADBmgmtAgent Service** を選択します。
- 4 「操作」メニューから「開始」を選択します。

▼ Solaris または Linux 上のスタンドアロンインストールで HADB を起動するには

- 1 **Application Server** インストールの HADB bin ディレクトリに移ります。 *install-dir* /hadb/4/bin
- 2 次のコマンドを実行して、エージェントを起動します。
`./ma-initd start`

▼ Windows 上のスタンドアロンインストールで HADB を起動するには

- 1 端末ウィンドウで、**Application Server** インストールの HADB bin ディレクトリに移ります。 *install-dir*\hadb\4. x\bin
x は HADB のリリース番号を表しています。
- 2 次のコマンドを実行して、エージェントを起動します。
`ma -i ma.cfg`

高可用性のためのクラスタおよびアプリケーションの設定

HADB を使用するには FirstCluster クラスタを設定する必要があり、HTTP セッションの持続性を確認できるようにするには clusterjsp アプリケーションの高可用性を有効にする必要があります。 `asadmin configure-ha-cluster` コマンドを使用して、既存のクラスタを高可用性のために設定します。このコマンドの使用方法については、`asadmin` コマンドプロンプトで `configure-ha-cluster --help` と入力するか、`configure-ha-cluster(1)` のマニュアルページを参照してください。

クラスタの再起動

前のセクションで行なった変更を有効にするには、クラスタのインスタンスを再起動する必要があります。

▼ クラスタを再起動するには

- 1 管理コンソールで、「クラスタ」ノードを展開します。
- 2 「FirstCluster」をクリックします。
- 3 右側の区画で、「インスタンスの停止」をクリックします。
- 4 インスタンスが停止したら、「インスタンスの起動」をクリックします。

HTTPセッションフェイルオーバーの検証

セッションデータのフェイルオーバーのテスト手順は、[37 ページの「ロードバランスの検証」](#)で説明したロードバランスのテスト手順と似ています。ここでは、セッションデータは障害のあと保持されます。サンプルアプリケーションは障害が発生したあとに自動的に再試行するように設定されているため、フェイルオーバーはユーザーには透過的に行われます。

▼ HTTPセッションフェイルオーバーを検証するには

- 1 **clusterjsp** アプリケーションを表示するには、ブラウザで、次の URL を入力します。

```
http://localhost :web_server_port /clusterjsp
```

`localhost` 変数を、Web サーバーを実行するシステム名に置き換えます。

`web_server_port` 変数を、`web_server_install_dir /https-hostname /config/server.xml` の LS 要素のポート属性の値に置き換えます。この例では、ポート 38000 を使用しています。

[34 ページの「アプリケーションの配備を確認するには」](#)で表示されたのと同じようなページが表示されます。

- 2 セッションおよびホスト情報が次のとおり表示されるかどうかを確認します。次に例を示します。

- Executed From Server: *localhost*
- Server Port Number: 38000
- Executed Server IP Address: 192.18.145.133
- Session ID: 41880f618e4593e14fb5d0ac434b1
- Session Created: Wed Feb 23 15:23:18 PST 2005

- 3 サーバーアクセスログファイルを表示して、アプリケーションを処理しているアプリケーションサーバーインスタンスを確認します。このログファイルは、次の場所にあります。
 - Solaris Java Enterprise System インストールの場合

```
/var/opt/SUNWappserver/nodeagents/nodeagent_name  
/i1/logs/access/server_access_log  
  
/var/opt/SUNWappserver/nodeagents/nodeagent_name  
/i2/logs/access/server_access_log
```
 - Linux Java Enterprise System インストールの場合

```
/var/opt/sun/appserver/nodeagents/  
nodeagent_name/i1/logs/access/server_access_log  
  
/var/opt/sun/appserver/nodeagents/  
nodeagent_name/i2/logs/access/server_access_log
```
 - Windows Java Enterprise System インストールの場合

```
install-dir \nodeagents\nodeagent_name\i1\logs\access\server_access_log  
install-dir\nodeagents\nodeagent_name\i2\logs\access\server_access_log
```
 - スタンドアロンの Application Server インストールの場合

```
install-dir /nodeagents/nodeagent_name/i1/logs/access/server_access_log  
install-dir/nodeagents/nodeagent_name/i2/logs/access/server_access_log
```
- 4 このページを処理している **Application Server** インスタンスを停止します。
 - a. 管理コンソールの左側の区画で、「クラスタ」を展開します。
 - b. 「FirstCluster」をクリックします。
 - c. 右側の区画で、「インスタンス」タブをクリックします。
 - d. 要求に対応したサーバーインスタンスの横にあるチェックボックスをクリックし、「停止」ボタンをクリックします。
- 5 `clusterjsp` サンプルアプリケーションページを読み込み直します。
セッション ID とセッション属性データは保持されます。
- 6 その他の **Application Server** インスタンスのアクセスログをチェックし、この時点でこれが要求を処理していることに注目します。
HTTPセッションは持続的に HADB に格納されるため、状態のフェイルオーバー機能が作動します。HTTPセッションの状態だけでなく、Application Server も EJB の状態を HADB に格納することができます。

次の手順 おめでとうございます。これで、Application Server のクイックスタートの手順をすべて修了しました。

ここでは、HADB をインストール、設定、および起動し、高可用性を使用するようクラスタとアプリケーションを設定しました。Application Server のその他の情報については、[23 ページの「次の学習ステップ」](#)を参照してください。

