



Red Hat JBoss Enterprise Application Platform 7.3

Microsoft Azure で JBoss EAP を使用する

高可用性の設定など、Microsoft Azure で Red Hat JBoss Enterprise Application Platform を使用する手順

Red Hat JBoss Enterprise Application Platform 7.3 Microsoft Azure で JBoss EAP を使用する

高可用性の設定など、Microsoft Azure で Red Hat JBoss Enterprise Application Platform を使用する手順

法律上の通知

Copyright © 2023 Red Hat, Inc.

The text of and illustrations in this document are licensed by Red Hat under a Creative Commons Attribution–Share Alike 3.0 Unported license ("CC-BY-SA"). An explanation of CC-BY-SA is available at

<http://creativecommons.org/licenses/by-sa/3.0/>

. In accordance with CC-BY-SA, if you distribute this document or an adaptation of it, you must provide the URL for the original version.

Red Hat, as the licensor of this document, waives the right to enforce, and agrees not to assert, Section 4d of CC-BY-SA to the fullest extent permitted by applicable law.

Red Hat, Red Hat Enterprise Linux, the Shadowman logo, the Red Hat logo, JBoss, OpenShift, Fedora, the Infinity logo, and RHCE are trademarks of Red Hat, Inc., registered in the United States and other countries.

Linux[®] is the registered trademark of Linus Torvalds in the United States and other countries.

Java[®] is a registered trademark of Oracle and/or its affiliates.

XFS[®] is a trademark of Silicon Graphics International Corp. or its subsidiaries in the United States and/or other countries.

MySQL[®] is a registered trademark of MySQL AB in the United States, the European Union and other countries.

Node.js[®] is an official trademark of Joyent. Red Hat is not formally related to or endorsed by the official Joyent Node.js open source or commercial project.

The OpenStack[®] Word Mark and OpenStack logo are either registered trademarks/service marks or trademarks/service marks of the OpenStack Foundation, in the United States and other countries and are used with the OpenStack Foundation's permission. We are not affiliated with, endorsed or sponsored by the OpenStack Foundation, or the OpenStack community.

All other trademarks are the property of their respective owners.

概要

本書は、高可用性設定を含む、Microsoft Azure で Red Hat JBoss Enterprise Application Platform を使用するためのガイドです。

目次

第1章 はじめに	3
1.1. RED HAT CLOUD ACCESS 製品ページ	3
第2章 サポートされる設定	4
2.1. サポートされない機能	4
第3章 MICROSOFT AZURE 環境の作成	5
第4章 JBOSS EAP のインストール	6
第5章 JBOSS EAP サブシステムを CLOUD PLATFORM で機能するように設定する	7
5.1. WEB サービス	7
5.2. MESSAGING	7
5.3. 高可用性のリモータリング設定	7
第6章 JBOSS EAP での MICROSOFT AZURE の設定	9
第7章 MICROSOFT AZURE での JBOSS EAP 高可用性の使用	10
7.1. JBOSS EAP の高可用性のための AZURE_PING の設定	10
7.2. JBOSS EAP 高可用性の開始	12
7.3. JBOSS EAP の高可用性のトラブルシューティング	13

第1章 はじめに

JBoss EAP 7 は、Azure で JBoss EAP を実行するためにサポートされる特定の設定内で使用する限り、Microsoft Azure プラットフォームで使用できます。クラスター化された JBoss EAP 環境を設定する場合、Azure で JBoss EAP クラスターリング機能を使用するために必要な特定の設定を適用する必要があります。

本ガイドでは、Microsoft Azure で JBoss EAP を使用するためのサポートされる設定と、Azure で JBoss EAP クラスターリングを有効にするために必要な特定の JBoss EAP 設定について詳しく説明します。本ガイドに記載されていないその他すべての JBoss EAP 機能は、他の JBoss EAP インストールと同じように Azure で正常に動作します。Azure 固有の設定手順については、その他の [JBoss EAP ドキュメント](#) を参照してください。

1.1. RED HAT CLOUD ACCESS 製品ページ

Red Hat Cloud Access は、Amazon EC2 や Microsoft Azure などの Red Hat 認定クラウドプロバイダーで JBoss EAP をサポートする Red Hat サブスクリプション機能です。Red Hat Cloud Access では、従来のサーバーとパブリッククラウドベースのリソース間で、シンプルかつコスト効果の高い方法でサブスクリプションを移動することができます。

詳細は、[Red Hat Cloud Access on the Customer Portal](#) を参照してください。

第2章 サポートされる設定

以下は、Microsoft Azure で JBoss EAP を使用するための唯一の仮想マシンオペレーティングシステムです。

- Red Hat Enterprise Linux 7
- Red Hat Enterprise Linux 8
- Microsoft Windows Server 2012 R2
- Microsoft Windows Server 2016

[Red Hat Cloud Access](#) プログラムを使用して、JBoss EAP サブスクリプションを使用して独自の Azure 仮想マシン、または Microsoft Azure Marketplace から上記のいずれかの On-Demand オペレーティングシステムに JBoss EAP をインストールできます。仮想マシンのオペレーティングシステムサブスクリプションは JBoss EAP サブスクリプションとは異なることに注意してください。

上記のオペレーティングシステムの制限以外に、サポート対象の Java Development Kit (JDK) ベンダーおよびバージョンなどの [JBoss EAP](#) のサポートされる設定に関する詳細は、カスタマーポータルを参照してください。

2.1. サポートされない機能

Microsoft Azure 環境で JBoss EAP を使用する場合、一部のサポートされない機能はサポートされません。

管理対象ドメイン

JBoss EAP 管理対象ドメインは Microsoft Azure ではサポートされません。スタンドアロン JBoss EAP サーバーインスタンスのみがサポートされます。スタンドアロン JBoss EAP サーバーを使用した JBoss EAP クラスターの設定は Azure でサポートされていることに注意してください。

ActiveMQ Artemis 共有ストアを使用した高可用性

Artemis 共有ストアを使用した JBoss EAP メッセージング高可用性は、Microsoft Azure ではサポートされていません。Azure で JBoss EAP メッセージング高可用性を設定するには、[ActiveMQ Artemis High Availability](#) のセクションを参照してください。

mod_cluster Advertising

JBoss EAP を Undertow mod_cluster プロキシロードバランサーとして使用する場合は、Azure UDP マルチキャストの制限により mod_cluster アドバタイズ機能がサポートされません。[アドバタイズ](#)ではなく、[プロキシリスト](#)を使用するように mod_cluster 負荷分散を設定する必要があります。

第3章 MICROSOFT AZURE 環境の作成

Microsoft Azure 環境で JBoss EAP インスタンスをホストする仮想マシンを作成します。仮想マシンは、**Standard_A2** 以上の Azure サイズを使用する必要があります。

Azure On-Demand Premium イメージを使用して仮想マシンを作成したり、仮想マシンを手動で作成したりできます。

- Red Hat Enterprise Linux 仮想マシンの場合は、カスタマーポータルの手順を参照してください。
 - [Using the On-Demand Marketplace Red Hat Enterprise Linux 7 image in Azure .](#)
 - [Manually creating and provisioning a Red Hat Enterprise Linux 7 image for Azure .](#)
- Microsoft Windows Server 仮想マシンの場合は、Microsoft Azure で Windows Server 仮想マシンを作成する方法の説明である [Microsoft Azure documentation](#) を参照してください。



重要

JBoss EAP の高可用性を設定する場合は、同じ仮想ネットワーク内に仮想マシンを作成する必要があります。

第4章 JBOSS EAP のインストール



重要

Microsoft Azure Marketplace の Red Hat Enterprise Linux On-Demand 仮想マシンを使用している場合は、ZIP またはインストーラーの方法を使用して JBoss EAP をインストールする必要があります。Red Hat サブスクリプション管理には、Red Hat Enterprise Linux On-Demand 仮想マシンを登録しないでください。その仮想マシンへの請求が重複してしまいます。

Microsoft Azure 環境の仮想マシンへの JBoss EAP のインストールは、通常の JBoss EAP インストールと同じです。詳細は、[JBoss EAPインストールガイド](#) を参照してください。

第5章 JBOSS EAP サブシステムを CLOUD PLATFORM で機能するように設定する

Amazon EC2 や Microsoft Azure などのクラウドプラットフォームで適切に動作するように、一部の JBoss EAP サブシステムを設定する必要があります。これは、JBoss EAP サーバーは通常、クラウドプラットフォームからのみ確認可能な **10.x.x.x** などのクラウド仮想マシンのプライベート IP アドレスにバインドされているために必要です。特定のサブシステムでは、このアドレスをサーバーのパブリック IP アドレスにマッピングする必要もあります。これは、クラウド外から見えるものです。

5.1. WEB サービス

クライアントが **Service.create(SELECTURL, serviceName)** を使用して Web サービスリクエストを行うと、ユーザーはサーバーのパブリック IP アドレスに接続しますが、その後、**webservices** サブシステムのサーバー設定ファイルに定義されたアドレスにリダイレクトされます。デフォルトでは、このアドレスは **/\${jboss.bind.address:127.0.0.1}** です。つまり、クラウドプラットフォームでは呼び出し元はサーバーのプライベート IP アドレスにリダイレクトされ、要求を解決できません。サーバーのパブリック IP アドレスは、以下のコマンドを使用して **ogg-host** 要素で設定する必要があります。

```
/subsystem=webservices:write-attribute(name=wsdl-host,value=PUBLIC_IP_ADDRESS)
```

5.2. MESSAGING

クラウドプラットフォームでメッセージングを使用する場合は、クライアントが使用する接続ファクトリーには、サーバーのパブリック IP アドレスを指すコネクタが必要です。

そのため、**完全な** プロファイルを実行している JBoss EAP サーバーに対して新しいコネクタとソケットバインディングを作成する必要があります。

1. 参照される **http-public** ソケットバインディングは **socket-binding-group** 内に作成する必要があります。

```
/socket-binding-group=standard-sockets/remote-destination-outbound-socket-binding=http-public:add(host=PUBLIC_IP_ADDRESS,port=${jboss.http.port:8080})
```

2. **messaging** サブシステムに新しい **http-connector** 要素を作成します。

```
/subsystem=messaging-activemq/server=default/http-connector=http-public-connector:add(endpoint=http-acceptor, socket-binding=http-public)
```

3. **connection-factory** に **connectors** を設定します。これは、クライアントによって使用されます。たとえば、デフォルトの接続としての **RemoteConnectionFactory** の設定は以下のようになります。

```
/subsystem=messaging-activemq/server=default/connection-factory=RemoteConnectionFactory:write-attribute(name=connectors, value=["http-public-connector"])
```

5.3. 高可用性のリモート設定

クラウドプラットフォーム上のクラスター化された EJB で JBoss EAP HA 機能を使用している場合は、EJB クライアントがクラスタービューの更新を受け取るために **remoting** サブシステムに追加の設定が必要になります。

これは **remoting** サブシステムのソケットバインディングに **client-mappings** を設定して行います。

```
/socket-binding-group=standard-sockets/socket-binding=http:write-attribute(name=client-mappings,value=[{ "destination-address" => "PUBLIC_IP_ADDRESS", "destination-port" => "8080" }])
```

第6章 JBOSS EAP での MICROSOFT AZURE の設定

Red Hat Enterprise Linux で起動時に JBoss EAP をサービスとして起動

JBoss EAP を Microsoft Azure の Red Hat Enterprise Linux でサービスとして起動する場合は、JBoss EAP サービスの以下の設定を変更して、Azure 仮想マシンのホスト名が初期化された後に開始されるようにする必要があります。

`/etc/systemd/system/multi-user.target.wants/eap7-standalone.service` を開き、`waagent.service` の依存関係を追加します。

以下の行を置き換えます。

```
After=syslog.target network.target
```

上記の行を、以下のように置き換えます。

```
After=syslog.target network.target waagent.service
Requires=waagent.service
```

Mod_cluster を使用した負荷分散

`mod_cluster` アドバタイズが Azure でサポートされていないため、JBoss EAP の負荷分散環境を設定する場合は、すべてのバランサーとワーカーが内部の Microsoft Azure 仮想ネットワークでアクセス可能な IP アドレスにバインドする必要があります。



注記

バランサーおよびワーカーは、**0.0.0.0** ではなく、実際の内部 Azure 仮想 IP アドレス (**172.28.0.2**) にバインドする必要があります。

また、`mod_cluster` アドバタイズが利用できないため、JBoss EAP を Undertow `mod_cluster` プロキシロードバランサーとして使用する場合は、以下のように各ワーカーノードが各バランサーの IP アドレスとポートが含まれるプロキシリストを使用するよう設定する必要があります。

```
/socket-binding-group=standard-sockets/remote-destination-outbound-socket-binding=proxy-one:add(host="BALANCER_IP_ADDRESS", port="BALANCER_PORT")
/subsystem=modcluster/mod-cluster-config=configuration:list-add(name=proxies,value=proxy-one)
```

ActiveMQ Artemis 高可用性

Artemis 共有ストアを使用した JBoss EAP メッセージング高可用性は、Microsoft Azure ではサポートされていません。Azure で JBoss EAP メッセージング高可用性を設定するには、TCP JGroups スタックを使用して `AZURE_PING JGroups 検索プロトコル` でレプリケートされたジャーナルを使用する必要があります。レプリケートされたジャーナルの使用に関する詳細は、JBoss EAP `Configuring Messaging` の `Data Replication` を参照してください。

`AZURE_PING` で TCP JGroups スタックを設定した後、JGroups スタックを `messaging-activemq` サブシステムの `discovery-group` および `broadcast-group jgroups-cluster` 設定に追加する必要があります。詳細は、JBoss EAP `Configuring Messaging` の `Server Discovery` を参照してください。

第7章 MICROSOFT AZURE での JBOSS EAP 高可用性の使用

Microsoft Azure は、UDP マルチキャストを基にした JGroups 検出プロトコルをサポートしていません。他の JGroups 検出プロトコル (静的設定 (**TCPPING**))、共有データベース (**JDBC_PING**)、共有ファイルシステムベースの ping (**FILE_PING**)、または **TCPGOSSIP** などを使用することは可能ですが、**AZURE_PING** は Azure 用に特別に開発した共有ファイル検出プロトコルを使用することを強くお勧めします。

7.1. JBOSS EAP の高可用性のための AZURE_PING の設定

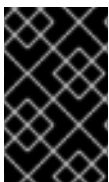
ここでは、AZURE_PING JGroups 検出プロトコルを使用するよう JBoss EAP クラスターを設定する方法を説明します。[仮想マシンの作成時に前提条件](#)を満たしていることを確認します。

AZURE_PING は、Microsoft Azure ストレージアカウント内で共通の Blob コンテナを使用します。AZURE_PING が使用できる Blob コンテナがない場合は、仮想マシンがアクセスできる Blob コンテナを作成します。

Blob コンテナの作成後に、AZURE_PING を設定するために以下の情報が必要になります。

- **storage_account_name**: blob コンテナが含まれる Microsoft Azure ストレージアカウントの名前。
- **storage_access_key**: ストレージアカウントのシークレットアクセスキー。
- **container**: **PING** データに使用する blob コンテナの名前。

AZURE_PING を JGroups 検出プロトコルとして使用するよう JBoss EAP を設定するには、[事前設定された JBoss EAP 設定ファイルのサンプルを使用するか](#)、[既存の設定を変更する](#) ことができます。



重要

以下の手順では、UDP JGroups スタックを使用して AZURE_PING を設定します。[Azure で JBoss EAP メッセージング高可用性](#)を設定する場合は、代わりに TCP JGroups スタックで AZURE_PING を設定する必要があります。

7.1.1. サンプル設定ファイルの使用

JBoss EAP には、Microsoft Azure のスタンドアロンサーバーのクラスターリングを設定するための設定ファイルのサンプルが含まれています。これらのファイルは **EAP_HOME/docs/examples/configs/** にあり、**standalone-azure-ha.xml** と **standalone-azure-full-ha.xml** です。



注記

サーバープロファイル間の相違点は、JBoss EAP [設定ガイド](#) の [相違点の説明](#) を参照してください。

これらの設定ファイルのサンプルは、Microsoft Azure でクラスターリングを使用するために事前設定されており、必要なのは Azure ストレージアカウントおよび blob コンテナの値を指定することのみです。

必要な設定ファイルのサンプルを **EAP_HOME/standalone/configuration/** にコピーします。

7.1.2. 既存設定の変更

既存の JBoss EAP の高可用性設定を変更する場合は、**jgroups** サブシステムに以下の変更を加える必要があります。

1. 管理 CLI を起動し、サーバーを埋め込み、選択した設定ファイルにオフライン変更を行います。以下に例を示します。

```
$ EAP_HOME/bin/jboss-cli.sh
[disconnected /] embed-server --server-config=standalone-ha.xml
```

2. デフォルトでは、JGroups は UDP スタックを使用します。別のスタックを使用していた場合は、UDP スタックの使用に戻ります。

```
[standalone@embedded /] /subsystem=jgroups/channel=ee:write-
attribute(name=stack,value=udp)
```

3. 以下のバッチコマンドを実行して既存の UDP スタックを削除し、Microsoft Azure 用に設定された新しい UDP スタックを挿入します。

```
batch
/subsystem=jgroups/stack=udp:remove
/subsystem=jgroups/stack=udp:add()
/subsystem=jgroups/stack=udp/transport=UDP:add(socket-binding=jgroups-udp,properties=
{ip_mcast=false})
/subsystem=jgroups/stack=udp/protocol=azure.AZURE_PING:add(properties=
{storage_account_name="${jboss.jgroups.azure_ping.storage_account_name}",
storage_access_key="${jboss.jgroups.azure_ping.storage_access_key}",
container="${jboss.jgroups.azure_ping.container}")})
/subsystem=jgroups/stack=udp/protocol=MERGE3:add
/subsystem=jgroups/stack=udp/protocol=FD_SOCKET:add(socket-binding=jgroups-udp-fd)
/subsystem=jgroups/stack=udp/protocol=FD:add
/subsystem=jgroups/stack=udp/protocol=VERIFY_SUSPECT:add
/subsystem=jgroups/stack=udp/protocol=pbcast.NAKACK2:add(properties=
{use_mcast_xmit=false,use_mcast_xmit_req=false})
/subsystem=jgroups/stack=udp/protocol=UNICAST3:add
/subsystem=jgroups/stack=udp/protocol=pbcast.STABLE:add
/subsystem=jgroups/stack=udp/protocol=pbcast.GMS:add
/subsystem=jgroups/stack=udp/protocol=UFC:add
/subsystem=jgroups/stack=udp/protocol=FRAG2:add
run-batch
```



注記

設定ファイルで Microsoft Azure ストレージアカウントおよび blob コンテナの値を保存する場合は、上記の設定のシステムプロパティの参照を Azure 環境の値に置き換えます。以下のコマンドは、JBoss EAP を起動する例で、システムプロパティを使用します。

設定ファイルのスタック XML は以下のようになります。

```
<stack name="udp">
  <transport type="UDP" socket-binding="jgroups-udp">
    <property name="ip_mcast">
      false
    </property>
```

```

</transport>
<protocol type="azure.AZURE_PING">
  <property name="storage_account_name">
    ${jboss.jgroups.azure_ping.storage_account_name}
  </property>
  <property name="storage_access_key">
    ${jboss.jgroups.azure_ping.storage_access_key}
  </property>
  <property name="container">
    ${jboss.jgroups.azure_ping.container}
  </property>
</protocol>
<protocol type="MERGE3"/>
<protocol type="FD SOCK" socket-binding="jgroups-udp-fd"/>
<protocol type="FD"/>
<protocol type="VERIFY_SUSPECT"/>
<protocol type="pbcast.NAKACK2">
  <property name="use_mcast_xmit">
    false
  </property>
  <property name="use_mcast_xmit_req">
    false
  </property>
</protocol>
<protocol type="UNICAST3"/>
<protocol type="pbcast.STABLE"/>
<protocol type="pbcast.GMS"/>
<protocol type="UFC"/>
<protocol type="FRAG2"/>
</stack>

```

4. 埋め込みサーバーを停止し、管理 CLI を終了します。

```

[standalone@embedded /] stop-embedded-server
[disconnected /] exit

```

7.2. JBOSS EAP 高可用性の開始

Microsoft Azure で高可用性を使用して JBoss EAP を起動するには、以下を行う必要があります。

- [AZURE_PING 検索プロトコル](#)で設定された設定ファイルを使用し、Microsoft Azure ストレージアカウントおよび blob コンテナに必要な値を指定。
- プライベートインターフェイスを、トラフィックのクラスターリングに使用される **Microsoft Azure** 内部 IP アドレスにバインド。これは、以下に示すように起動時に実行することができます。あるいは、[JBoss EAP設定ガイド](#)で示されている設定に示されるように設定することもできます。



警告

セキュリティ上の理由から、クラスターリングトラフィックは意図しないネットワークに公開しないようにする必要があります。

これは、エンドポイントを Microsoft Azure 仮想ネットワークに制限するか、トラフィックをクラスターリングするための専用の仮想ネットワークと専用の仮想マシン NIC を作成して実行できます。

以下のコマンドを使用して JBoss EAP の高可用性インスタンスを起動します。Microsoft Azure ストレージアカウントおよび blob コンテナの値を設定ファイルに保存した場合、-

Djboss.jgroups.azure_ping システムプロパティーの定義を省略することができます。

```
EAP_HOME/bin/standalone.sh -b IP_ADDRESS -bprivate IP_ADDRESS --server-  
config=EAP_CONFIG_FILE.xml -  
Djboss.jgroups.azure_ping.storage_account_name=STORAGE_ACCOUNT_NAME -  
Djboss.jgroups.azure_ping.storage_access_key=STORAGE_ACCESS_KEY -  
Djboss.jgroups.azure_ping.container=CONTAINER_NAME
```

以下に例を示します。

```
EAP_HOME/bin/standalone.sh -b 172.28.0.2 -bprivate 172.28.0.2 --server-config=standalone-azure-  
ha.xml -Djboss.jgroups.azure_ping.storage_account_name=my_storage_account -  
Djboss.jgroups.azure_ping.storage_access_key=y7+2x7P68pQse9MNH58Bkk5po9OGzeJc+0IRqYcQ9  
Cr/Sp4xiUFJVlBY+MGXJRNx3syksikwm4tOYIFgjvoCmw== -  
Djboss.jgroups.azure_ping.container=my_blob_container
```



注記

JBoss EAP サブシステムは必要とときにのみ起動されるため、分散可能なアプリケーションを JBoss EAP サーバーにデプロイして、高可用性 JBoss EAP サブシステムを起動する必要があります。

クラスター内で別の JBoss EAP インスタンスを起動した後、クラスター内の最初のサーバーのコンソールログに以下のようなログが表示されるはずですが、

```
INFO [org.infinispan.remoting.transport.jgroups.JGroupsTransport] (thread-2,ee,eap-server-1)  
ISPN000094: Received new cluster view for channel server: [eap-server-1|1] (2) [eap-server-1, eap-  
server-2]
```

7.3. JBOSS EAP の高可用性のトラブルシューティング

Microsoft Azure でクラスターリングが機能しない場合は、以下のリストにある要件をすべて完了していることを確認してください。

- JBoss EAP サーバーをホストする Microsoft Azure 仮想マシンが同じ仮想ネットワークを使用していることを確認します。

- AZURE_PING で使用する Blob コンテナがあることを確認します。
- **Jgroups** サブシステムに AZURE_PING 検出プロトコルが設定された JBoss EAP 設定ファイルを使用していることを確認します。
- パブリックインターフェイスとプライベートインターフェイスの両方を適切な Microsoft Azure IP アドレスにバインドしていることを確認します。
- Microsoft Azure ストレージアカウントおよび Blob コンテナに正しい値があり、以下のいずれかを行うようにします。
 - 設定ファイルへの保存
 - JBoss EAP の起動時に適切なシステムプロパティを設定。
- 配布可能な Java アプリケーションがクラスター内のすべての JBoss EAP サーバーにデプロイされていることを確認。

7.3.1. Blob コンテナでの古い検出ファイルのクリーニング

AZURE_PING を使用する JBoss EAP クラスターが異常終了した場合 (たとえば、**kill -9** を使用して JBoss EAP プロセスを終了する場合など)、一部の古い検出ファイルは Blob コンテナに残される可能性があります。

これらのファイルは、通常、正常なクラスターのシャットダウン時にクリーンアップされます。ただし、正常でないシャットダウンから残ると、オンラインではなくなったノードとの通信を試行するクラスターメンバーの起動パフォーマンスに影響する可能性があります。

これが問題となる場合は、以下の設定でクラスターの表示が変更されるたびに、クラスターコーディネーターがすべての検出ファイルを削除し、更新することができます。

```
/subsystem=jgroups/stack=udp/protocol=azure.AZURE_PING/property=remove_all_data_on_view_change:add(value=true)
```

注記

または、各表示の変更でコンテナをクリーンアップすることが理想的でない場合は、クラスターにジョインしようとするノードの結合試行回数を減らすことができます。デフォルトのジョイン試行数は **10** です。たとえば、結合試行の数を **3** に設定するには、次のコマンドを実行します。

```
/subsystem=jgroups/stack=udp/protocol=pbcast.GMS/property=max_join_attempts:add(value=3)
```

古い検出ファイルは引き続き存在しますが、クラスターにジョインしようとしているノードは、オンラインでなくなったノードとの通信を試行するのに多くの時間を費やしません。

