

blueprism[®]

HubおよびInteract 4.7

高可用性構成ガイド

Document Revision: 2.0



商標および著作権

本文書に記載されている情報は、Blue Prism Limitedが独占的に所有する機密情報であり、権限を与えられたBlue Prism担当者の書面による同意なしに、第三者に開示してはなりません。本文書のいかなる部分も、複写機などの電子的あるいは機械的な形式や手段を問わず、Blue Prism Limitedの書面による許可を得ることなく、複製または送信してはなりません。

© 2023 Blue Prism Limited

Blue Prism、Blue Prismのロゴ、Prismデバイスは、Blue Prism Limitedおよびその関係会社の商標または登録商標です。All Rights Reserved.

すべての商標は本文書によって確認され、各所有者のために使用されています。
Blue Prismは、本文書で言及する外部Webサイトの内容に関して、責任を負いません。

Blue Prism Limited, 2 Cinnamon Park, Crab Lane, Warrington, WA2 0XP, United Kingdom。
英国で登録: 登録番号4260035。電話: +44 370 879 3000。Web: www.blueprism.com

内容


高可用性構成	4
対象者	4
前提条件	5
Rabbit MQクラスター	5
SQL高可用性グループ	5
ロードバランサー	6
Webサーバー	6
Blue Prismソフト ウェアとスクリプト	6
インフラストラクチャの例	7
HAProxyを使用した高可用性構成	7
IIS ARRおよびHAProxyを使用した高可用性構成	8
アプリケーションゲートウェイを使用した高可用性構成	9
HAProxyロードバランサー – 構成例	10
例 (HAProxy) ロードバランサーのスクリプト	10
初期インストールと構成	13
Blue Prism Hubをインストールする	13
Blue Prism Interactをインストールする	13
インストールを構成する	13
Webバインディング設定を更新する	14
スクリプト化されたソリューション	15
重要情報	15
Functions.ps1スクリプト	16
Prepare.ps1スクリプト	17
Setup.ps1スクリプト	20
ポストスクリプトの構成	21
高可用性環境をテストする	22
ロギング	24

高可用性構成


高可用性により、システムは複数のサーバーを使用して常に使用可能になります。複数のサーバーを構成することで、組み込みの耐障害性が得られます。1台のサーバーに障害が発生しても、可用性は失われません。

Blue PrismはBlue Prism® HubとBlue Prism® Interact Webサーバーを別のWebサーバーホストに複製するスクリプト化されたソリューションを提供します。

このガイドでは、Blue Prismが提供するスクリプトを使用して高可用性構成用にWebサーバーホストを準備する方法について説明します。この情報は、あくまでも概要として提供されています。業界標準のベストプラクティスに従い、経験豊かな専門家にアドバイスを求めることをお勧めします。

 複数のノードで高可用性環境を使用している場合、HubとInteractは現在、サーバーで一度に250件のリクエストをサポートできます。たとえば、同時に250人のユーザー全員がフォームで **送信** をクリックしてもサポートします。サーバーにリクエスト(テキストフィールドへの入力など)を送信することなく、情報を表示したり入力したりしているユーザーがシステム上に大勢同時にいることも考えられます。Blue Prismは、今後のリリースでこの制限を増やすことを目指しています。

高可用性のデプロイメントを開始する前にIT組織に確認して、ネットワークインフラストラクチャが意図したデプロイメントをサポートできることを確認してください。

 高可用性のインストールと構成プロセスを視聴するには、[Blue Prism HubとInteractの高可用性インストールビデオ](#)を参照してください。

対象者

このガイドは、次の分野のITプロフェッショナルを対象としています。

- Webサーバーホストの構成
- PowerShellスクリプトの使用

前提条件

このガイドでは、高可用性構成でのBlue Prismソフトウェアの構成のみについて説明します。必要なサードパーティ製品の構成の詳細は扱いません。

Hub(およびオプションでInteract) を設定する前に、次の操作を行う必要があります。

- **RabbitMQクラスター** – 3台 (またはそれ以上) のホストでインストールおよび構成されます。
- **SQL高可用性グループ** – 2 ~ 3台 のホストでインストールおよび構成されます。
- **ロードバランサー** – 1 ~ 2台 のホストでインストールおよび構成されます。
- **Webサーバー** – Blue Prism HubとBlue Prism Interactをインストールする準備が整った前提条件のソフトウェアとともにインストールされます。
- **Blue Prismソフトウェアとスクリプト** – Blue Prismの高可用性環境を構成するインストーラーとスクリプト。

Rabbit MQクラスター

RabbitMQ クラスターでは、3台以上のRabbitMQサーバーとミラーキューを使用することが推奨されます。サーバーはすべて、同じバージョンのRabbitMQとErlangを実行している必要があります。必要なバージョンについては、「[「メッセージブローカーサーバー」](#)」を参照してください。

クラスターの作成に関する情報は、RabbitMQのウェブサイト (<https://www.rabbitmq.com/clustering.html>) で確認できます。

SQL高可用性グループ

Always-On可用性グループでは、2台以上のSQL Serverを使用することが推奨されます。Azureを使用している場合は、Azureロードバランサーが必要です。


Blue Prism HubやBlue Prism Interactが標準構成(単一のWebサーバー)でインストールされている場合、インストーラーがデータベースを作成します。ただし高可用性構成では、ソフトウェアをインストールする前に必要なデータベースを手動で作成する必要があります。主要サービスのキャッシュとして使用される追加のデータベースも作成します。必要なデータベースは次のとおりです。

- AuditDB
- AuthenticationServerDB
- EmailServiceDB
- FileServiceDB
- HubDB
- LicenseManagerDB
- NotificationCenterDB
- AuthenticationServerCache
- HubCache

Interactもインストールする場合は、次の追加データベースが必要です。

- IadaDB
- InteractDB
- InteractCache

Always-On可用性グループの詳細については、Microsoftの文書を参照してください: <https://docs.microsoft.com/en-us/sql/database-engine/availability-groups/windows/overview-of-always-on-availability-groups-sql-server?view=sql-server-ver15>

 データベース管理者と協力してデータベースを作成し、Always-On可用性グループに追加します。スクリプト化されたプロセスがあり、組織に必要な追加の構成設定がある可能性があります。

ロードバランサー

高可用性環境のサーバーにタスクを頒布するには、ロードバランサーが必要です。組織に最適なロードバランサーを使用してください。監査イベントに適切なIPアドレスを保存するには、ロードバランサーでIP透過を設定する必要があります。

このガイドでは高可用性構成のロードバランサーの例として、HAProxyを使用します。この構成の詳細については、「[インフラストラクチャの例 次のページ](#)」および「[HAProxyロードバランサー – 構成例 ページ10](#)」を参照してください。

Webサーバー


高可用性構成で使用するWebサーバーホストを準備する必要があります。HubやInteractの初回インストールと構成にはWebサーバーが必要です。設定を複製する追加のWebサーバーホストも準備します。

サーバーのハードウェアおよびソフトウェア要件については、「[Hubインストールガイド](#)」および「[Interactインストールガイド](#)」を参照してください。

ホストを準備するには、次の手順を実行します。

- 各ホストにIISをインストールします。詳細については、「[IISをインストールする](#)」を参照してください。
- Microsoft .NET Core Runtime(6.0.9または6.0.10) およびMicrosoft Windows Desktop Runtime (6.0.9または6.0.10) を各ホストにインストールします。追加ホストでは、最初のWebサーバーと同じバージョンを使用する必要があります。インストールの詳細については、「[.NET Coreコンポーネントをインストールする](#)」を参照してください。

ソフトウェアのダウンロードの詳細については、「[Hubインストールガイド](#)」を参照してください。

 ソフトウェアをインストールする前に、Windows認証またはSQL認証のどちらを使用するかを決定する必要があります。Windows認証を使用する場合、サービスアカウントが適切なアプリケーションプールと証明書にアクセスできるように構成されていることを確認します。これは、最初のWebサーバーをインストールした後、HAスクリプトを実行する前に行う必要があります。詳細については、「[Windows認証を使用してHubをインストールする](#)」および「[Windows認証を使用してInteractをインストールする](#)」を参照してください。

Blue Prismソフトウェアとスクリプト

Blue Prismポータルから以下をダウンロードする必要があります。

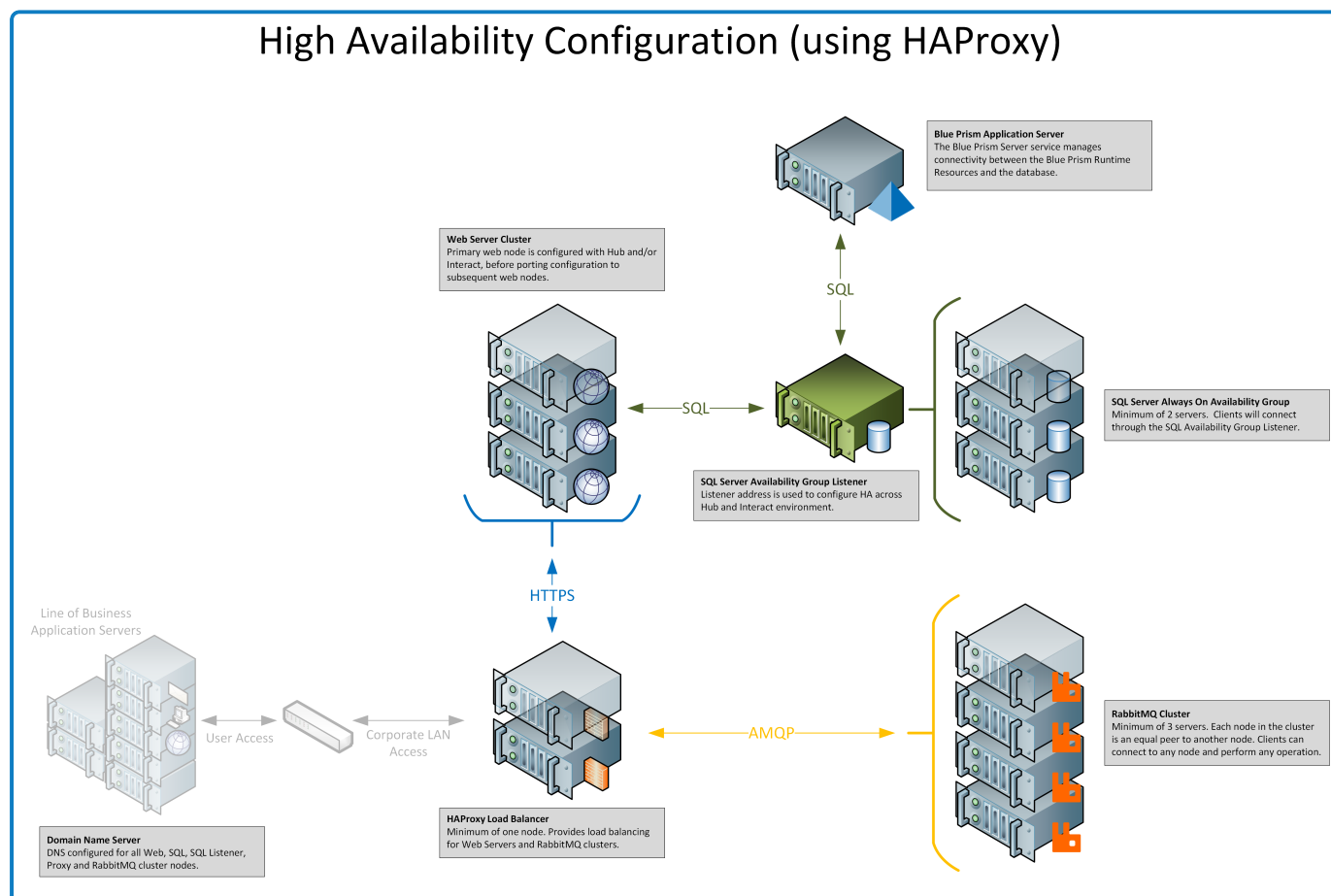
- Blue Prism Hubインストーラー
- Blue Prism Interactインストーラー
- 高可用性スクリプト
- Blue Prism Data Protector

インフラストラクチャの例

次の図は、高可用性配置のインフラストラクチャ構成の例を示しています。

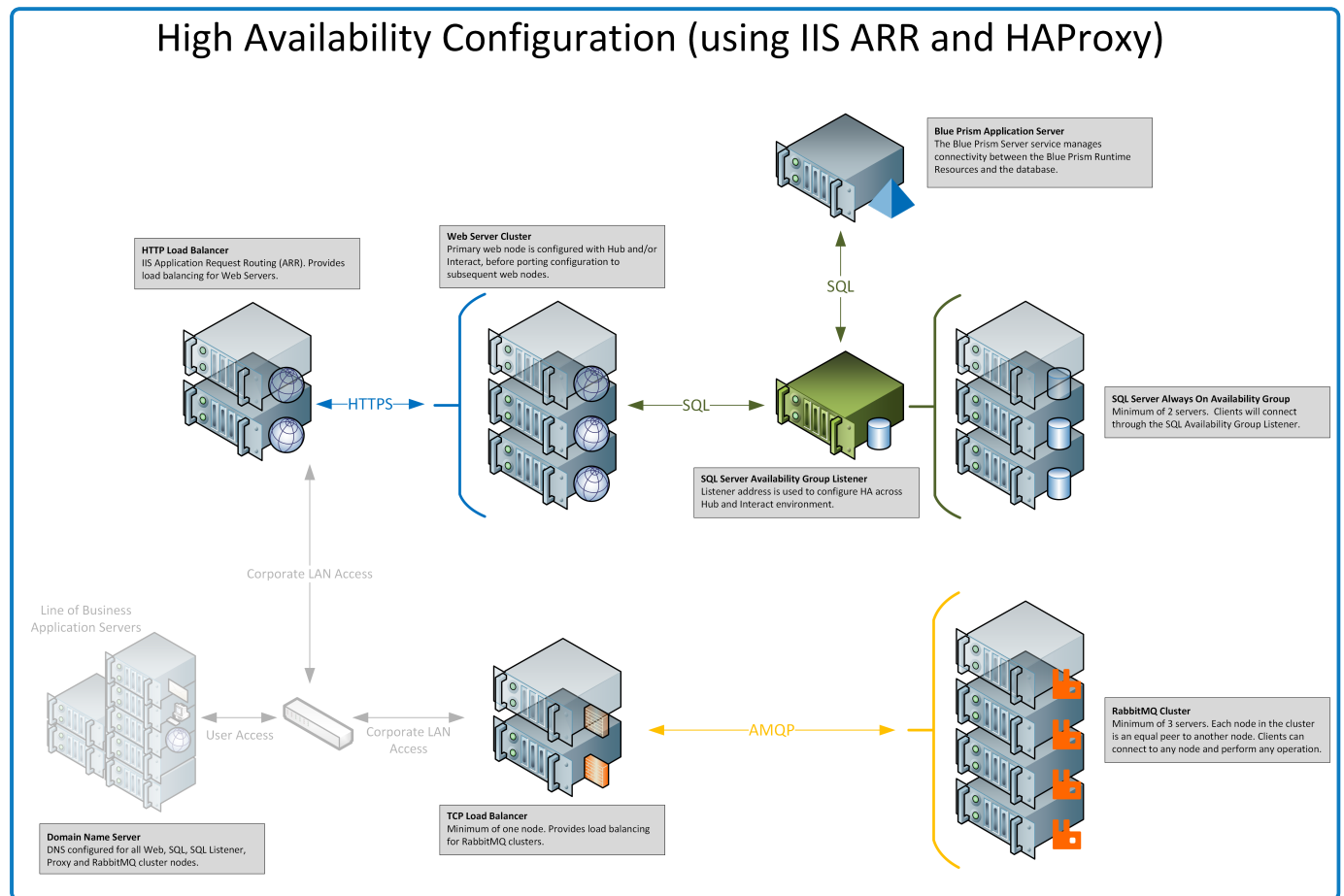
- HAProxyを使用 – このガイドでは例としてこの構成を使用します。
- IIS ARRおよびHAProxyを使用する
- アプリケーションゲートウェイを使用する

HAProxyを使用した高可用性構成



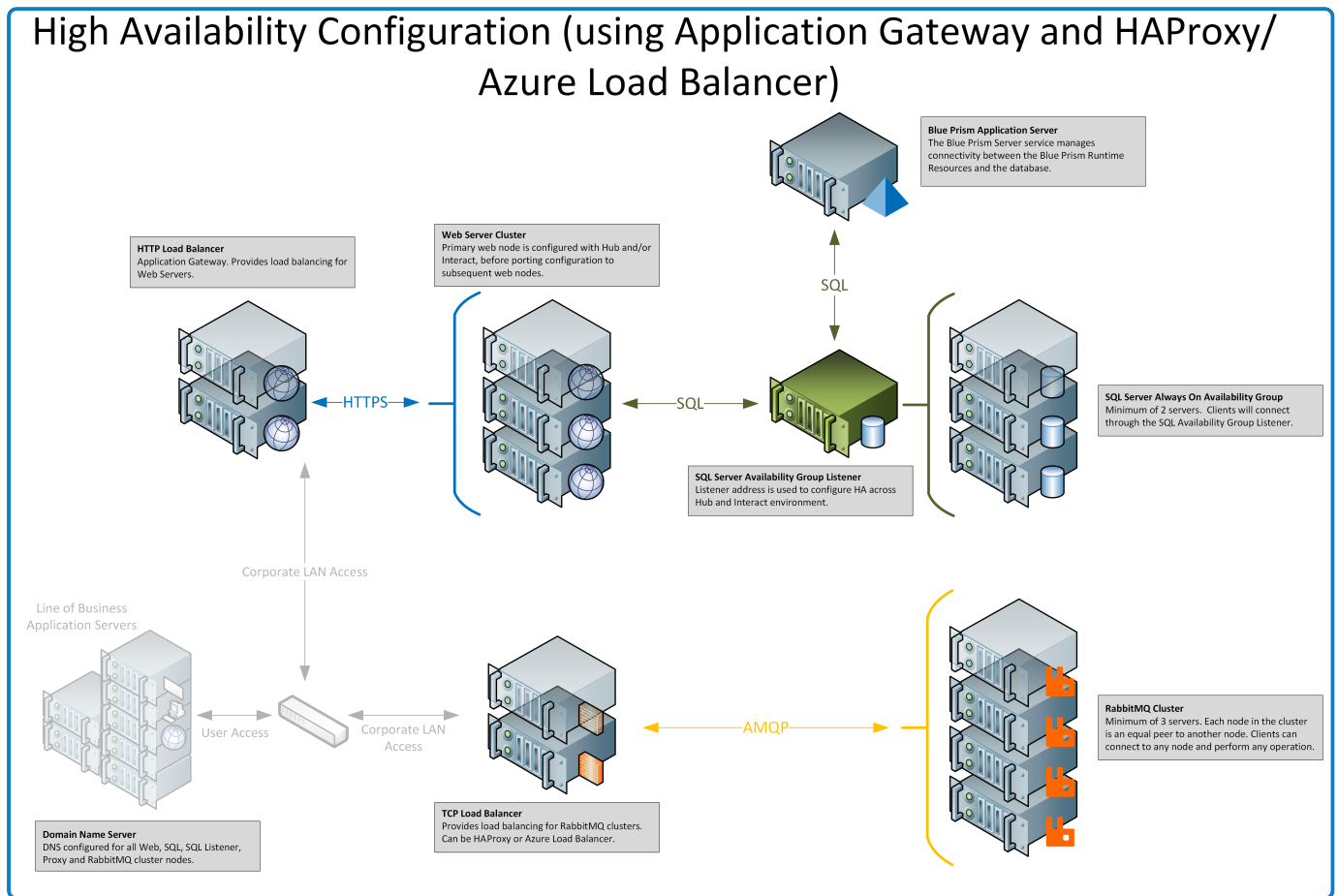
このガイドの情報では、この構成を例として使用します。

IIS ARRおよびHAProxyを使用した高可用性構成



アプリケーションゲートウェイを使用した高可用性構成

High Availability Configuration (using Application Gateway and HAProxy/
Azure Load Balancer)



HAProxyロードバランサー – 構成例

組織に最適なロードバランサーを使用してください。以下の情報は、HAProxy(v2.4) ロードバランサーの設定例を示しています。

この例では、Blue PrismはLinuxマシンでHAProxy v2.4を使用しました(最小仕様: Ubuntu 20.04、1vcpu、2GB RAM)。

例 (HAProxy) ロードバランサーのスク립ト

次の例では、/etc/haproxy/haproxy.cfgファイルを使用するHAProxy(v2.4) ロードバランサーを使用します。

基本的なスク립ト構造の例

```
global
    # global settings here

defaults
    # defaults here


frontend
    # a frontend that accepts requests from clients

backend
    # servers that fulfill the requests
```

ここでは、

- **グローバル**見出しの下の設定は、低レベルでHAProxyに影響するプロセス全体のセキュリティおよびパフォーマンス構成を定義します。
- **デフォルト**セクションを使用すると、重複を減らせます。設定は、その後の**フロントエンド**と**バックエンド**のすべてのセクションに適用されます。その次のセクションで設定を上書きできます。
- **バックエンド**サーバーの前にリバースプロキシとしてHAProxyを配置すると、クライアントが接続できるIPアドレスとポートが**フロントエンド**セクションによって定義されます。
- **バックエンド**セクションは、ロードバランシングされ、リクエストを処理するために割り当てられるサーバーのグループを定義します。「web_servers」など、各**バックエンド**にラベルを追加できます。

構成例

 正しいフォーマットについては、以下の例のオンライン版を参照してください。

```
#Example of HAPROXY config
#ANMQP loadbalancer for 3 nodes with IP-addresses 10.30.0.10,10.30.0.20,10.30.0.30
#HTTPS loadbalancer without SSL termination for 2 nodes with IP-addresses 10.30.0.50,10.30.0.60
#statistics is available at https://haproxyname.yourdomainname.com:10001/stats with adminname:adminpassword credits
global
    log /dev/log      local0
    log /dev/log      local1 notice
    chroot /var/lib/haproxy
    stats socket /run/haproxy/admin.sock mode 660 level admin expose-fd listeners
    stats timeout 30s
    user haproxy
    group haproxy
    daemon

    # Default SSL material locations
    ca-base /etc/ssl/certs
    crt-base /etc/ssl/private

    # See: https://ssl-config.mozilla.org/#server=haproxy&server-version=2.0.3&config=intermediate
    ssl-default-bind-ciphers ECDHE-ECDSA-AES128-GCM-SHA256:ECDHE-RSA-AES128-GCM-SHA256:ECDHE-ECDSA-AES256-GCM-SHA384:ECDHE-
    RSA-AES256-GCM-SHA384:ECDHE-ECDSA-CHACHA20-POLY1305:ECDHE-RSA-CHACHA20-POLY1305:DHE-RSA-AES128-GCM-SHA256:DHE-RSA-AES256-GCM-
    SHA384
    ssl-default-bind-ciphersuites TLS_AES_128_GCM_SHA256:TLS_AES_256_GCM_SHA384:TLS_CHACHA20_POLY1305_SHA256
    ssl-default-bind-options ssl-min-ver TLSv1.2 no-tls-tickets
```

```
defaults
    log      global
    mode     http
    option   httplog
    option   dontlognull
    timeout  connect 5000
    timeout  client 50000
    timeout  server 50000
    errorfile 400 /etc/haproxy/errors/400.http
    errorfile 403 /etc/haproxy/errors/403.http
    errorfile 408 /etc/haproxy/errors/408.http
    errorfile 500 /etc/haproxy/errors/500.http
    errorfile 502 /etc/haproxy/errors/502.http
    errorfile 503 /etc/haproxy/errors/503.http
    errorfile 504 /etc/haproxy/errors/504.http

frontend stats
    bind *:10001 ssl crt /etc/haproxy/cert/yourdomainname.pem
    mode http
    stats enable
    stats hide-version
    stats refresh 10s
    stats show-node
    stats auth adminname:adminpassword
    stats uri /stats

frontend main_frontend
    bind *:443 ssl crt /etc/haproxy/cert/yourdomainname.pem
    acl ims_acl hdr(host) -i ims.yourdomainname.com
    acl hub_acl hdr(host) -i hub.yourdomainname.com
    acl interact_acl hdr(host) -i interact.yourdomainname.com
    acl audit_acl hdr(host) -i audit.yourdomainname.com
    acl emailqueue_acl hdr(host) -i emailqueue.yourdomainname.com
    acl fileserver_acl hdr(host) -i fileserver.yourdomainname.com
    acl iada_acl hdr(host) -i iada.yourdomainname.com
    acl interactremoteapi_acl hdr(host) -i interactremoteapi.yourdomainname.com
    acl licensemanager_acl hdr(host) -i licensemanager.yourdomainname.com
    acl notificationcenter_acl hdr(host) -i notificationcenter.yourdomainname.com
    acl signalr_acl hdr(host) -i signalr.yourdomainname.com
    use_backend ims_backend if ims_acl
    use_backend hub_backend if hub_acl
    use_backend interact_backend if interact_acl
    use_backend audit_backend if audit_acl
    use_backend emailqueue_backend if emailqueue_acl
    use_backend fileserver_backend if fileserver_acl
    use_backend iada_backend if iada_acl
    use_backend interactremoteapi_backend if interactremoteapi_acl
    use_backend licensemanager_backend if licensemanager_acl
    use_backend notificationcenter_backend if notificationcenter_acl
    use_backend signalr_backend if signalr_acl

frontend amqp_frontend
    bind *:5672
    mode tcp
    option tcplog
    use_backend amqp_backend

backend amqp_backend
    mode tcp
    balance roundrobin
    server rabbit1 10.30.0.10:5672 check inter 5s
    server rabbit2 10.30.0.20:5672 check inter 5s
    server rabbit3 10.30.0.30:5672 check inter 5s

backend ims_backend
    balance roundrobin
    option httpchk
    http-check send meth GET uri /health ver HTTP/1.1 hdr host ims.yourdomainname.com
    http-check expect string Healthy
    server web1 10.30.0.50:443 ssl ca-file /etc/haproxy/cert/root.ca check inter 5s
    server web2 10.30.0.60:443 ssl ca-file /etc/haproxy/cert/root.ca check inter 5s

backend hub_backend
    balance roundrobin
    option httpchk
    http-check send meth GET uri /health ver HTTP/1.1 hdr host hub.yourdomainname.com
    http-check expect string Healthy
    server web1 10.30.0.50:443 ssl ca-file /etc/haproxy/cert/root.ca check inter 5s
    server web2 10.30.0.60:443 ssl ca-file /etc/haproxy/cert/root.ca check inter 5s

backend interact_backend
    balance roundrobin
```

```
option httpchk
http-check send meth GET uri /health ver HTTP/1.1 hdr host interact.yourdomainname.com
http-check expect string Healthy
server web1 10.30.0.50:443 ssl ca-file /etc/haproxy/cert/root.ca check inter 5s
server web2 10.30.0.60:443 ssl ca-file /etc/haproxy/cert/root.ca check inter 5s

backend audit_backend
balance roundrobin
option httpchk
http-check send meth GET uri /health ver HTTP/1.1 hdr host audit.yourdomainname.com
http-check expect string Healthy
server web1 10.30.0.50:443 ssl ca-file /etc/haproxy/cert/root.ca check inter 5s
server web2 10.30.0.60:443 ssl ca-file /etc/haproxy/cert/root.ca check inter 5s

backend emailqueue_backend
balance roundrobin
option httpchk
http-check send meth GET uri /health ver HTTP/1.1 hdr host emailqueue.yourdomainname.com
http-check expect string Healthy
server web1 10.30.0.50:443 ssl ca-file /etc/haproxy/cert/root.ca check inter 5s
server web2 10.30.0.60:443 ssl ca-file /etc/haproxy/cert/root.ca check inter 5s

backend fileserver_backend
balance roundrobin
option httpchk
http-check send meth GET uri /health ver HTTP/1.1 hdr host fileserver.yourdomainname.com
http-check expect string Healthy
server web1 10.30.0.50:443 ssl ca-file /etc/haproxy/cert/root.ca check inter 5s
server web2 10.30.0.60:443 ssl ca-file /etc/haproxy/cert/root.ca check inter 5s

backend iada_backend
balance roundrobin
option httpchk
http-check send meth GET uri /health ver HTTP/1.1 hdr host iada.yourdomainname.com
http-check expect string Healthy
server web1 10.30.0.50:443 ssl ca-file /etc/haproxy/cert/root.ca check inter 5s
server web2 10.30.0.60:443 ssl ca-file /etc/haproxy/cert/root.ca check inter 5s

backend interactremoteapi_backend
balance roundrobin
option httpchk
http-check send meth GET uri /health ver HTTP/1.1 hdr host interactremoteapi.yourdomainname.com
http-check expect string Healthy
server web1 10.30.0.50:443 ssl ca-file /etc/haproxy/cert/root.ca check inter 5s
server web2 10.30.0.60:443 ssl ca-file /etc/haproxy/cert/root.ca check inter 5s

backend licensemanager_backend
balance roundrobin
option httpchk
http-check send meth GET uri /health ver HTTP/1.1 hdr host licensemanager.yourdomainname.com
http-check expect string Healthy
server web1 10.30.0.50:443 ssl ca-file /etc/haproxy/cert/root.ca check inter 5s
server web2 10.30.0.60:443 ssl ca-file /etc/haproxy/cert/root.ca check inter 5s

backend notificationcenter_backend
balance roundrobin
option httpchk
http-check send meth GET uri /health ver HTTP/1.1 hdr host notificationcenter.yourdomainname.com
http-check expect string Healthy
server web1 10.30.0.50:443 ssl ca-file /etc/haproxy/cert/root.ca check inter 5s
server web2 10.30.0.60:443 ssl ca-file /etc/haproxy/cert/root.ca check inter 5s

backend signalr_backend
balance roundrobin
option httpchk
cookie SERVERID insert indirect nocache
http-check send meth GET uri /health ver HTTP/1.1 hdr host signalr.yourdomainname.com
http-check expect string Healthy
server web1 10.30.0.50:443 ssl ca-file /etc/haproxy/cert/root.ca check inter 5s cookie web1
server web2 10.30.0.60:443 ssl ca-file /etc/haproxy/cert/root.ca check inter 5s cookie web2
```

ここでは、

- ロード バランサーは、各 サービス、RabbitMQ クラスター、および統計のあるページに対して個別のフロントエンドセクションを使用します。
- SSL サポートを有効にするには、HAProxyは/etc/haproxy/cert/にあるcertフォルダーに証明書を持っている必要があります。
- HAProxyは5秒間隔で「/health」ページにリクエストを送信し、「Healthy」を返信として期待します。
- SignalR サービスはスティッキーセッション(クライアントが単一のサーバーにこだわる)を使用します。


初期インストールと構成

Blue Prism高可用性スクリプトを実行する前に、Hubを備えた初期Webサーバーをインストールし、構成する必要があります。必要に応じて、Interactを実行します。このWebサーバーを使用して、構成を高可用性構成の追加のWebサーバーに複製します。

Blue Prism Hubをインストールする

スクリプトを使用する前に、Hubを1台のWebサーバーにインストールする必要があります。標準インストールプロセスに従いますが、次の主な注意点があります。

- Hubインストーラーの前提条件2 – RabbitMQ]画面で、**サーバー名**]フィールドに、個々のRabbitMQノードではなくクラスター全体で使用されているロードバランサーのアドレスを入力します。

 デフォルトで設定された、RabbitMQのゲストアカウント認証情報は使用しないでください。Hub用にRabbitMQで作成したアカウントの認証情報を使用します。

- Hubインストーラーの各種 **SQL接続**]画面の **SQL Serverを選択してください**]フィールドで、個々のSQL Serverではなく可用性グループリスナーの詳細を入力します。また、**データベース名**]フィールドの情報が、**前提条件 ページ5**で、手動で作成した適切なデータベース名と一致していることを確認します。
- Hubインストーラーの各種 **IS設定**]画面で、**ホスト名**]フィールドにロードバランサーの構成で指定したホスト名を入力し、適切な証明書を選択します。

Windows認証を使用している場合は、サービスアカウントが適切なアプリケーションプールと証明書にアクセスできるように構成されていることを確認します。詳細については、「[Windows認証を使用してHubをインストールする](#)」を参照してください。

Blue Prism Interactをインストールする

Interactが必要な場合は、スクリプトを使用する前にInteractをインストールする必要があります。標準インストールプロセスに従いますが、次の主な注意点があります。

- Interactインストーラーの各種 **SQL接続**]画面で、**SQL Serverを選択してください**]フィールドに、個々のSQL Serverではなく可用性グループリスナーの詳細を入力します。また、**データベース名**]フィールドの情報が、**前提条件 ページ5**で、手動で作成した適切なデータベース名と一致していることを確認します。
- Interactインストーラーの各種 **IS設定**]画面で、**ホスト名**]フィールドにロードバランサーの構成で指定したホスト名を入力し、適切な証明書を選択します。

Windows認証を使用している場合は、サービスアカウントが適切なアプリケーションプールと証明書にアクセスできるように構成されていることを確認します。詳細については、「[Windows認証を使用してInteractをインストールする](#)」を参照してください。

インストールを構成する

- 標準的な手順に従って初めてインストールを構成します。詳細については、「[Hubの初期構成](#)」および「[Interactプラグインをインストールする](#)」を参照してください。

 データベース接続を構成する際、可用性グループリスナーの詳細を入力する必要があります。

- Automation Lifecycle Management(ALM) など、組織が使用するその他のプラグインをインストールしてライセンスを付与します。
- 環境の構成が完了したら、Hubからログアウトします。

Webバインディング設定を更新する

HubまたはInteractをインストールした後、初期Webサーバーの各サイトのWebバインディング設定で、**サーバー名表示要求**オプションがデフォルトで有効になります。組織のドメイン構造によっては、このオプションの変更が必要になる場合があります。最も一般的な構成では、このオプションを無効にします。

 設定の詳細については、「[重要情報 次のページ](#)」を参照してください。

サーバー名表示要求オプションを無効にするには、次を実行します。

1. 最初のWebサーバーでInternet Information Services (IIS) マネージャーを開きます。
2. **接続**ペインでサーバー名を展開し、**サイト**を展開してから、必要なWebサイトを選択します。
3. **アクション**パネルで、**バインディング**をクリックします。
サイトバインディング]ダイアログが表示されます。
4. バインディングを選択し、**編集...**をクリックします。
編集サイトバインディング]ダイアログが表示されます。
5. **サーバー名表示要求**オプションをオフにし、**OK**をクリックします。
6. **閉じる**をクリックします。
7. すべてのHub/Interactサイトでこのプロセスを繰り返します。


スクリプト化されたソリューション

初期のWebサーバー構成が完了したら、以下のスクリプトを使用して、HubとInteractを含む初期Webサーバーホストを高可用性用に準備したり、この構成を高可用性構成の新しいWebサーバーホストに複製したりできます。

次の順序で実行する必要がある3つのPowerShellスクリプトがあります。


	スクリプト	説明
1.	functions.ps1	外部PowerShellの機能を含むスクリプト(他の2つのスクリプトで使用)。
2.	prepare.ps1	最初のWebサーバーホストを準備するためのスクリプト。
3.	setup.ps1	追加のWebサーバーホストを構成するためのスクリプト。

これらのスクリプトとBlue Prism Data Protectorを、最初のWebサーバーホストのフォルダー(C:\Scriptsなど)にコピーします。

 常に管理者としてPowerShellを実行します。

重要情報


- prepare.ps1スクリプトのパラメーターとして接続文字列を渡す場合は、可用性グループリスナーを指定する必要があります。
- 推奨されるDNSのメソッドを使用する場合、各アプリケーションのホスト名はロードバランサーのプライベートIPを指定する必要があります。

 DNSが使用できず、ホストファイルが使用されている場合は、各Webサーバーホストのホストファイルを更新する必要があります。

- Blue Prism HubまたはInteractをインストールした後、初期Webサーバーの各サイトのWebバインディング設定で、**サーバー名表示要求**オプションがデフォルトで有効になります。このオプションは、次にする必要があります。
 - 無効 – 同じドメインを共有する複数のサブドメインWebサイトがあり、すべてのサブドメインにワイルドカード証明書または単一の証明書を使用している場合。

 これは、HubとInteract環境の最も一般的なセットアップです。

- 有効 – 同じドメインを共有しない複数のサブドメインWebサイトがあり、各ドメインに異なる証明書がある場合。
- 有効 – 関連するサブドメインWebサイトで、同じドメインを共有するサブドメインと、ドメインを共有しないサブドメインが複数ある場合。

 **サーバー名表示要求**オプションは、Internet Information Services (IIS) マネージャーで変更できます。詳細については、「[Webバインディング設定を更新する前のページ](#)」を参照してください。

Functions.ps1スクリプト

このスクリプトには、prepare.ps1スクリプトおよびsetup.ps1スクリプトで使用されるPowerShellの機能がすべて格納されます。

PowerShellの機能

functions.ps1スクリプトには、PowerShellの次の機能が含まれます。

PowerShellの機能	説明
Convert-Guid	GUIDをBlue Prism Hub/Interactインストール情報が保存されているレジストリパスのIDに変換します。
Install-Dependencies	PowerShellモジュールのインストールを簡素化し、パッケージプロバイダーを選択します。
Install-WinFeature	Windowsの機能をインストールします。
New-Site	新規IISサイトおよびアプリケーションプールを作成し、証明書を生成またはインポートして、拡張アプリケーションプールオプションを設定します。
New-HostedService	Windowsサービスを作成します。
New-Password	事前定義された条件でランダムなパスワードを生成します。
Remove-HostedService	Windowsサービスを削除します。
Remove-Site	IISサイトおよびアプリケーションプールを削除します。
Set-CertificatePrivateKeyAcl	ユーザーおよびグループにインストールされている証明書のプライベートキーに対する許可を設定します。
Set-FolderPermissions	フォルダーとファイルに対する許可を設定します。
Set-Logging	スクリプトのログレベルを設定します。


Functionsスクリプトを実行する


Functions.ps1スクリプトはPrepare.ps1スクリプトによって呼び出され、個別に実行する必要はありません。スクリプトの実行については、[Prepare.ps1スクリプト 次のページ](#)を参照してください。

Prepare.ps1スクリプト

prepare.ps1スクリプトは、Authentication Server、Hub、Interactなどのアプリケーションがある最初のWebサーバーホスト上ですべての準備を実行します。

まずはprepare.ps1スクリプトを、最初のWebサーバーホスト (HubとInteractがBlue Prismインストーラーからインストールされた場所) で実行する必要があります。

 また、C:\ScriptsフォルダーにBlue Prism Data Protectorも必要です。このツールの詳細については、「Blue Prism Data Protectorツール」を参照してください。

 prepare.ps1スクリプトは、追加のWebサーバーの作成に使用するファイルを生成します。出力ディレクトリのサイズは約800MBです。このスクリプトを実行する前に、1GB以上の空きディスク容量があることを確認してください。

機能

prepare.ps1スクリプトは、次の機能を提供します。

- レジストリからデータを収集し、レジストリファイルを作成します。
- 証明書をエクスポートします。
- アプリケーションとサービスを準備します。
- データベースの移行を実行します。
- appsettings.jsonファイルのプロパティを変更します。
- アプリケーションとサービスのルートフォルダーを圧縮します。
- variable.jsonファイルを生成し、それをデータとして取り込みます。

パラメーター

prepare.ps1スクリプトには、次のパラメーターが含まれます。

パラメーター	説明
-HubCacheConnectionString	Blue Prism Hub Distributed Cacheデータベースへの接続文字列を指定します。 これには可用性グループリスナーの詳細と、手動で作成したデータベース名 (HubCache) が必要です。
-AuthServerCacheConnectionString	Blue Prism Authentication Server Distributed Cacheデータベースへの接続文字列を指定します。 これには可用性グループリスナーの詳細と、手動で作成したデータベース名 (AuthenticationServerCache) が必要です。
-InteractCacheConnectionString	Blue Prism Interact Cacheデータベースへの接続文字列を指定します。 これには可用性グループリスナーの詳細と、手動で作成したデータベース名 (InteractCache) が必要です。 -IncludeInteract/パラメーターも含める場合、このパラメーターを含める必要があります。高可用性のHubのみを準備する場合は、必要ありません。

パラメーター	説明
-DataProtectorPath	実行可能なBlue Prism Data Protectorへのフルパスを指定します。
-IncludeInteract	他のWebサーバーホストにコピーするBlue Prism Interactのデータおよびファイルを収集する必要があることを指定します。 このパラメーターはオプションです。このパラメーターを含まない場合、スクリプトの影響を受けるのはHubのみです。

Prepareスクリプトを実行する

⚠ このスクリプトを実行する前に、1GB以上の空きディスク容量があることを確認してください。

- 最初のWebサーバーホストで、管理者としてPowerShellを実行し、スクリプトが保存されているフォルダーにディレクトリを変更します。例：

```
cd C:\Scripts
```

- Prepareスクリプトを実行します。[パラメーター 前のページ](#)にリスト表示されているパラメーターの詳細を指定する必要があります。スクリプトの例を以下に示します。

Blue Prism HubとBlue Prism Interactの両方をインストールし、SQL認証を使用している場合：

```
.\prepare.ps1 `
-HubCacheConnectionString 'Server=aglistener.domain.local;Database=HubCache;User
Id=sqladmin;Password=StR0nGP@ssw0rD;MultiSubnetFailover=True;' `
-AuthServerCacheConnectionString 'Server=aglistener.domain.local;Database=AuthenticationServerCache;User
Id=sqladmin;Password=StR0nGP@ssw0rD;MultiSubnetFailover=True;' `
-InteractCacheConnectionString 'Server=aglistener.domain.local;Database=InteractCache;User
Id=sqladmin;Password=StR0nGP@ssw0rD;MultiSubnetFailover=True;' `
-DataProtectorPath '.\BluePrismDataProtector.Console.exe' `
-IncludeInteract
```

Blue Prism Hubのみをインストールし、SQL認証を使用している場合：

```
.\prepare.ps1 `
-HubCacheConnectionString 'Server=aglistener.domain.local;Database=HubCache;User
Id=sqladmin;Password=StR0nGP@ssw0rD;MultiSubnetFailover=True;' `
-AuthServerCacheConnectionString 'Server=aglistener.domain.local;Database=AuthenticationServerCache;User
Id=sqladmin;Password=StR0nGP@ssw0rD;MultiSubnetFailover=True;' `
-DataProtectorPath '.\BluePrismDataProtector.Console.exe'
```

Blue Prism HubとBlue Prism Interactの両方をインストールし、Windows認証を使用している場合：

```
.\prepare.ps1 `
-HubCacheConnectionString 'Server=aglistener.domain.local;Database=HubCache;Integrated
Security=True;MultiSubnetFailover=True;' `
-AuthServerCacheConnectionString 'Server=aglistener.domain.local;Database=AuthenticationServerCache;Integrated
Security=True;MultiSubnetFailover=True;' `
-InteractCacheConnectionString 'Server=aglistener.domain.local;Database=InteractCache;Integrated
Security=True;MultiSubnetFailover=True;' `
-DataProtectorPath '.\BluePrismDataProtector.Console.exe' `
-IncludeInteract
```

Blue Prism Hubのみをインストールし、Windows認証を使用している場合：

```
.\prepare.ps1 `
-HubCacheConnectionString 'Server=aglistener.domain.local;Database=HubCache;Integrated
Security=True;MultiSubnetFailover=True;' `
```

```
-AuthServerCacheConnectionString 'Server=aglistener.domain.local;Database=AuthenticationServerCache;Integrated Security=True;MultiSubnetFailover=True;'`  
-DataProtectorPath '.\BluePrismDataProtector.Console.exe'
```

prepare.ps1スクリプトを実行すると、次が利用できるようになります。

- 圧縮されたアプリケーションコンテンツ、証明書、レジストリファイルを含むファイルフォルダーがスクリプトフォルダーに保存されました。
 - 各スクリプトに必要なすべての値を含む、スクリプトフォルダー内のvariables.jsonファイル。
3. 最初のWebサーバーホスト上のスクリプトフォルダー全体を、追加した各Webサーバーホストにコピーします。同じディレクトリ構造(たとえばC:\Scripts)を使用します。
- 次のステップについては、[Setup.ps1スクリプト](#) [次のページ](#)を参照してください。

Setup.ps1スクリプト

setup.ps1スクリプトは、prepare.ps1スクリプトによって準備されたアプリケーションのセットアップと構成を実行します。

setup.ps1スクリプトは、高可用性構成にある追加の各Webサーバーホストで実行する必要があります。Webサーバーのホストは、[前提条件 ページ5](#)に詳述されている前提条件のソフトウェアで設定する必要があります。

setup.ps1スクリプトを実行すると、次が使用できるようになります。

- インポートした証明書とレジストリファイル。
- インストールした必要なWindowsの機能と役割すべて。
- 最初のWebサーバーホストで作成および構成したすべてのアプリケーションとサービス。

 setup.ps1スクリプトは、[Prepare.ps1スクリプト ページ17](#)に必要な手順を完了後にのみ実行できます。

機能

setup.ps1スクリプトは、次の機能を提供します。


- レジストリファイルのインポート。
- 証明書のインポート。
- 証明書のプライベートキーの許可設定。
- Windowsの機能および役割のインストール。
- フォルダー構造の作成、およびアプリケーションコンテンツの抽出。
- 以前にエクスポートされたすべてのアプリケーションに対するIISサイトの作成。
- 以前にエクスポートされたすべてのサービスに対するWindowsサービスの作成。

パラメーター

setup.ps1スクリプトには、次のパラメーターが含まれます。

パラメーター	説明
-Force	すべてのサイトとサービスの再作成を強制します。

Setupスクリプトを実行する

 スクリプトを実行する前に、最初のWebサーバーからこのサーバーにC:\Scriptsをコピーしていることを確認してください。また、1GB以上の空きディスク容量があることを確認してください。

1. まだ実行していない場合は、スクリプトとファイルを含むフォルダー(C:\Scripts) を最初のWebサーバーからこのWebサーバーにコピーします。
2. PowerShellを管理者として実行し、スクリプトとファイルがコピーされたフォルダーにディレクトリを変更します。例：


```
cd C:\Scripts
```

3. Setupスクリプトを実行する場合、追加のパラメーターを指定する必要はありません。例：

```
.\setup.ps1
```

このスクリプトがWebサーバーを構築するため、スクリプトの実行には時間がかかります。

4. 構成が必要な追加のWebサーバーで、このプロセスを繰り返します。

 setup.ps1スクリプトの実行中にエラーが発生した場合は、-Forceパラメーターを使用して再度実行します。

```
.\setup.ps1 -Force
```

ポストスクリプトの構成

Windows認証を使用する場合は、追加のWebサーバーが正しく設定されていることを確認する必要があります。Webサーバーで、Windowsサービスアカウントが次を持つことを確認します。

- 必要な証明書の許可。
- IISアプリケーションプール上の所有権。
- HubによってインストールされたWindowsサービスの所有権。

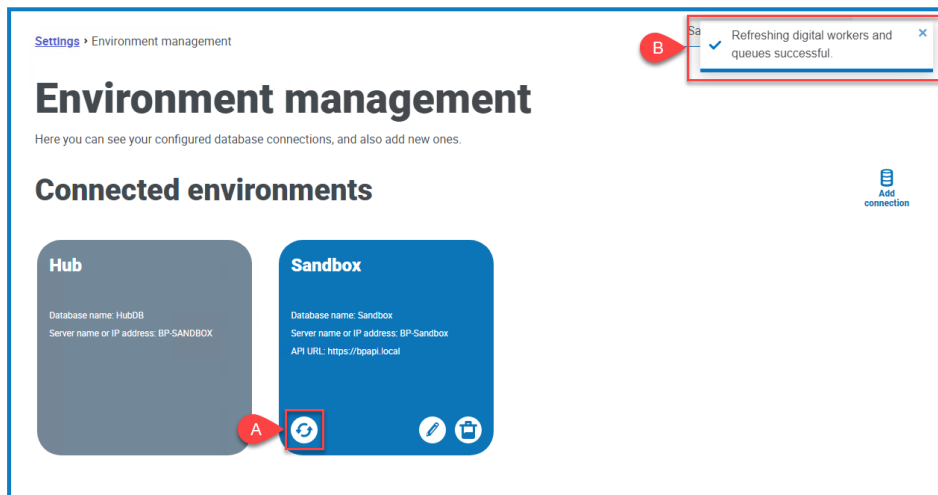
RabbitMQでAMQPS(Advanced Message Queuing Protocol - Secure)を使用している場合、アプリケーションプールにRabbitMQ証明書の許可を付与する必要があります。

詳細については、Hubインストールガイドの「[Windows認証を使用してHubをインストールする](#)」と「[RabbitMQをAMQPSと使用する](#)」を参照してください。さらに、Interactをインストールする場合は、Interactインストールガイドの「[Windows認証を使用してInteractをインストールする](#)」と「[RabbitMQをAMQPSと使用する](#)」を参照してください。

高可用性環境をテストする

高可用性環境の構成が完了したら、障害発生時に正常に動作することをテストする必要があります。
これには、以下の操作を行います。

1. 高可用性環境が最初から正常に動作していることを確認します。
 - a. Hub管理者としてBlue Prism Hubにログインし、[Prismプロファイル]アイコンをクリックして、[設定]ページの[環境管理]をクリックします。
[環境管理]ページが表示されます。
 - b. Blue Prismデータベーススタイルの[更新]アイコン(A)をクリックして、Hub環境のキューと情報を更新します。
[Digital Workersとキューを正常に更新しました]メッセージが表示されます(B)。



テストプロセス中は、このWebブラウザウィンドウを閉じないでください。

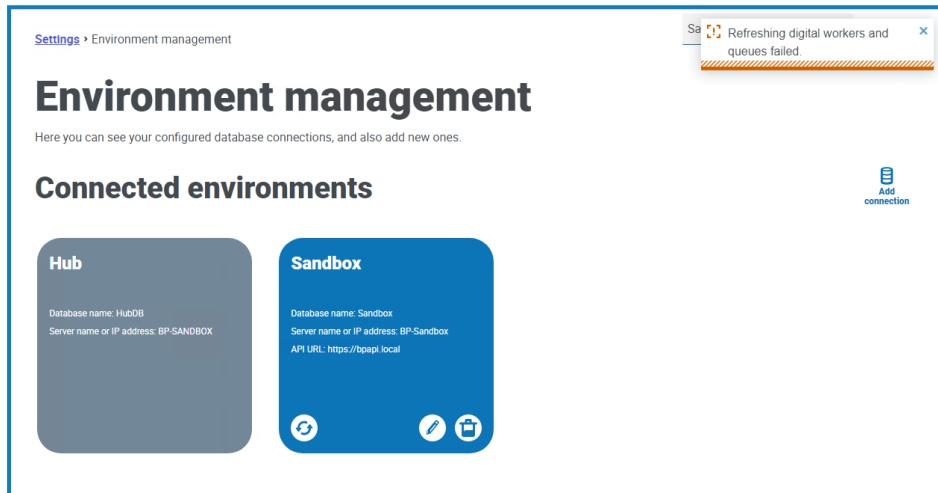
2. サーバー障害をシミュレーションする:
 - a. クラスタ内の追加Webサーバーのいずれかで、Internet Information Services(IIS) マネージャーを開き、サーバーを停止します。これでサーバー障害をシミュレーションします。
 - b. [Hub環境 マネージャー]ページを表示しているWebブラウザウィンドウに戻り、[更新]アイコンを再度クリックします。

通信を再度確立するため、[Digital Workersとキューを正常に更新しました]のメッセージが表示されるまで、情報の更新に若干時間がかかる場合があります。

3. サーバーの再起動と別のサーバーの障害をシミュレーションします。

- a. Internet Information Services(IIS) マネージャーで、ステップ2aで停止したWebサーバーを起動します。
- b. クラスタ内の最初のWebサーバー(最初のサーバー)で、Internet Information Services(IIS) マネージャーを開き、サーバーを停止します。
- c. [Hub環境 マネージャー] ページを表示しているWebブラウザウィンドウに戻り、[更新]アイコンを再度クリックします。

[更新に失敗しました]のメッセージが表示されます。障害は、サーバーを切り替えた後の負荷分散の遅延が原因です。



- d. 数秒待つてから、[更新]アイコンを再度クリックします。

[正常に更新されました]のメッセージが表示されます。それでも更新に失敗したメッセージが表示される場合は、この手順を繰り返します。

4. テストを完了します。

- a. Internet Information Services(IIS) マネージャーで、ステップ3bで停止したWebサーバーを起動します。

ロギング

ログファイルは、これらのスクリプトを実行した結果として作成されます。ログファイルは次の形式で作成されます。

<SYSTEM DRIVE>\<SERVER HOSTNAME>.log

例: C:\webserver.log。