HP IceWall技術レポート:

動作検証レポート:IceWall SSO +Amazon Web Services™

1. はじめに

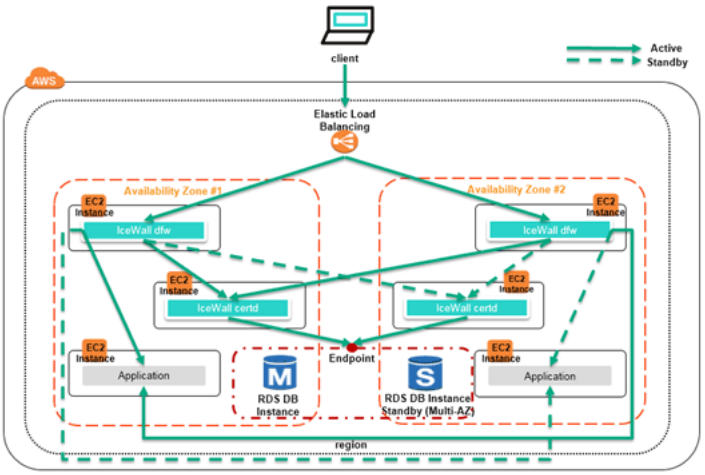
近年、導入・運用コストの削減などの理由から、業務システムのプラットフォームとしてIaaSやPaaSを活用する企業が増えています。
Amazon Web Services™ (以下、AWS)はIaaSやPaaSの分野で圧倒的なシェアを持つクラウドサービスであり、仮想サーバーやデータベース、コンテンツ配信などの幅広いサービスを提供しています。

IceWall SSOによる認証基盤も、AWSの仮想サーバー上で稼働させることが可能です。
本資料では、AWSの各種サービス(CloudFront, Amazon RDS等)を利用し構成した環境で、IceWallの動作確認を行った検証結果及びパラメーター設定時の注意点について紹介します。

2.構成

2.1 基本構成(冗長化構成)

AWS上にIceWall環境を構築する場合の基本構成は下図のようになります。
各コンポーネントを異なるアベィラビリティゾーンに配置することで、片方のアベィラビリティゾーンのサービスが使用できなくなった場合にもSSOのサービスを継続することが可能になります。



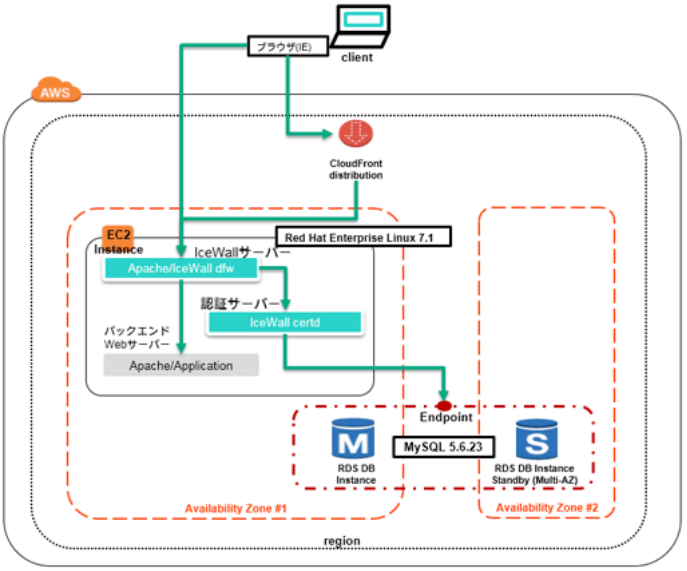
※IceWall dfwが稼働するEC2インスタンスへの負荷分散はAWSで提供される「Elastic Load Balancing」を使用します。

2.2 今回の検証で利用したAWSのサービスとSW構成

今回の検証では、AWSで提供されるサービスの内、特に以下のサービスとの連携を確認しました。

AWSサービス	説明	本検証での用途
Amazon EC2™	AWS上で提供される仮想サーバ OSはLinuxやWindowsから選択可能 OS、アプリケーションサーバー、アプリケーションを含むテンプレートから作成することが可能	IceWallサーバー 認証サーバー バックエンドWebサーバー
Amazon RDS™	AWS上で提供されるリレーショナルデータベースサービス 簡単な操作でデータベースを設定、運用、拡張することができる。 データベースエンジンはOracleなどのよく知られたものから選択可能	認証データベース
CloudFront™	AWSのコンテンツ配信サービス 配下のサーバー(オリジンサーバー)のコンテンツをキャッシュし、クライアントに配信する。コンテンツ取得のトラフィックを抑えることができ、コンテンツ表示のパフォーマンス向上が期待される。	キャッシュサーバー (クライアントとIceWallサーバーの間に配置)

本検証を実施した環境の構成は下図の通りです。



AWSの提供するサービスに関する詳細は、AWSのウェブサイト(https://aws.amazon.com/jp/)をご参照ください。
本検証に使用したSW構成は以下の通りです。

要素		SW	備考
クライアント	Windows 8.1	Internet Explorer 11	
キャッシュサーバー	–	CloudFront	
IceWallサーバー 兼 認証サーバー 兼 バックエンドWebサーバー	RedHat Enterprise Linux 7.1	Apache 2.4.6 IceWall SSO 10 dfw (Patch Release 8)	OSバンドル版を使用
		IceWall SSO 10 Certd (Patch Release 7) DBクライアントライブラリ ODBC Driver Manager: unixODBC 2.3.1 ODBC Driver: MySQL Connector/ODBC 5.2.5	OSバンドル版を使用
認証DB	–	MySQL(GPL) 5.6.23	マルチAZ配置

3. Amazon RDS(MySQL)との連携

3.1 Amazon RDS

本検証ではIceWallの認証DBとしてサポートされているMySQLを使用しました。

商用環境では、認証DBは冗長化されるのが一般的です。オンプレミス環境でMySQLを冗長化する場合は、2ノード間でレプリケーション構成をとり、IceWallのフェイルオーバー機能でプライマリ・セカンダリの切替を行います。
それに対し、本検証はAmazon RDS独自の冗長化方式であるMulti -AZ配置方式で認証DBの冗長化を行いました。Multi-AZ 配置では、プライマリのDBインスタンスを作成すると自動的に異なる「アベイラビリティゾーン」にスタンバイインスタンスが作成され、データが複製されます。「アベイラビリティゾーン」とは、物理的に独立したインフラストラクチャ上で稼働する領域の概念です。なお、AWSは世界各地に運用拠点があり、それぞれのデータセンター群を「リージョン」と呼びます。今回は一つの「リージョン」上の異なる「アベイラビリティゾーン」にプライマリ・セカンダリのDBインスタンスを配置しました。

Amazon RDS上に作成したDBインスタンスに接続する際は、インスタンス作成時に自動的に割り当てられた「エンドポイント」(FQDN:リッスンポートの形式)を使用します。この「エンドポイント」のFQDN部分は、AWS上のDNSでIPアドレスに解決され、フェイルオーバー時には、FQDNに紐づくIPアドレスがセカンダリDBインスタンスのものに替わります。

3.2 IceWallの設定

IceWallの認証モジュールからMySQLへの接続は、通常のMySQLと同様にODBC Driverを使用します。
Amazon RDS上のDBインスタンスに接続する際は、ODBC DriverでDSN(データソース名)を登録する際に、「エンドポイント」の情報を指定します。(なお、前述の通り、フェイルオーバー発生時にはエンドポイントに紐づくIPアドレスが替わるため、接続先のサーバーを指定する際は、IPアドレスではなくFQDNを使用します。)

IceWallの認証モジュール設定ファイルcert.confでは、DBHOST項目に登録したDSNを指定しますが、今回はIceWallのDBに対するフェイルオーバー機能は使用しないため、DBHOSTに指定するDSNは一つになります。

3.3 動作検証

以下の検証を実施し、Amazon RDS上に構成したDBインスタンスが、IceWall SSOの認証DBとして正常時及びフェイルオーバー後に問題なく動作することを確認しました。

1. ログイン処理・ログアウト処理・パスワード変更処理の確認
2. フェイルオーバー所要時間の測定
3. フェイルオーバー後の動作(ログイン処理)の確認

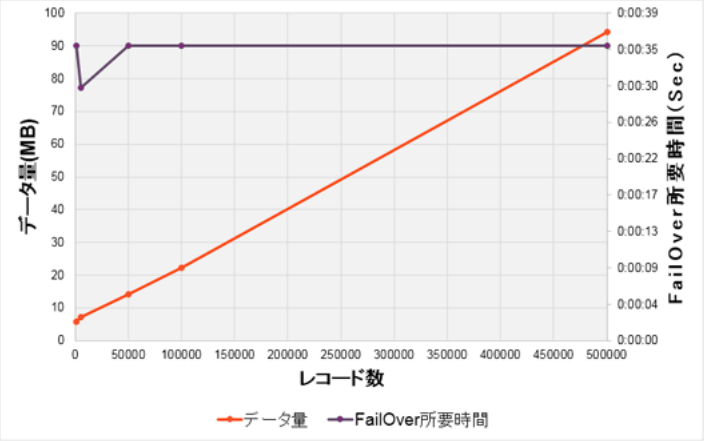
3.3.1 動作検証:ログイン処理・ログアウト処理・パスワード変更処理の確認

IceWall SSOの処理でDBへのアクセスが発生するのは「ログイン」「ログアウト」「パスワード変更」の3つのタイミングです。本検証では、上記3処理を行い問題なく処理が行えること(DB上の認証テーブルのデータが更新されること)を確認しました。

3.3.2 動作検証：フェイルオーバー所要時間の測定

通常、DBのフェイルオーバー処理中は、DBのサービスを使用することができません。IceWall SSOでは認証処理に認証DBを使用しているため、DBサービスの中断中は新たな認証を行うことができませんが、認証済のリクエストについてはDBアクセスが発生しないため、DBサービスの中断中も問題なく処理を行うことが可能です。

フェイルオーバーに要する時間は、データ量やDBインスタンスの配置状態(リージョン、アベイラビリティゾーン)により異なると考えられますが、本検証環境でユーザーデータを50万レコードまで徐々に増加させフェイルオーバー所要時間を測定しました。その結果、本環境におけるフェイルオーバー処理は、データ量によらず約35秒程度で処理が完了しました。なお、ユーザーデータはIceWallで使用する標準的な情報を登録したもので、1レコード(=1ユーザー)あたり150バイト程度です。



※Amazon RDS上で意図的にフェイルオーバーを発生させる場合は、AWSの各種サービスの管理コンソールであるAWS Management Consoleから該当のDBインスタンスを選択し、DBインスタンスの操作メニューから再起動を選択します。その際「フェイルオーバーし再起動しますか?」というチェックボックスにチェックを入れます。また、AWS Management Consoleのログから、フェイルオーバー動作の開始と完了の時刻を確認することも可能です。

3.3.3 動作検証：フェイルオーバー後の動作(ログイン処理)の確認

DBインスタンスでフェイルオーバー完了後に、ログイン処理が正常に行えるかを確認しました。
フェイルオーバー後に正常に処理を再開するためには、TCPのタイムアウト値、IceWall認証モジュールがDBインスタンスとの間に確立するTCPコネクションの数、Webサーバーの画面タイムアウト値を含めた調整を行う必要があります。以下では本検証環境におけるパラメーター調整を例として紹介します。

IceWall認証モジュールはサービスの起動時にDBインスタンスの間にTCPのコネクションを確立し、ログイン処理などのDBアクセスに使用します。なお、起動時に確立するTCPコネクション数は認証モジュールの設定ファイルcert.confのMAXDBCONNECT項目で指定することが可能です。
Multi-AZ 配置による冗長化構成の場合、IceWall認証モジュールのサービス起動時には、プライマリのDBインスタンスとの間にTCPのコネクションが確立されます。フェイルオーバーが発生するとDBサービスがセカンダリのインスタンスに切り替わりますが、その際DB側からこのコネクションは切断されません。そのため、IceWall認証モジュール側から見ると、起動時に確立したTCPコネクションは確立済(ESTABLISHED)の状態のままです。

しかしこのコネクションはフェイルオーバー発生後は使用することができません。IceWall認証モジュールは、DBアクセスの失敗を検知すると自動的に再接続を試みますが、アクセス失敗を検知するまでの時間はOSのTCPパラメーター値に依存します。本検証を実施したRedHat 7.1ではTCPパラメーター「tcp_retries2」に制御されています。

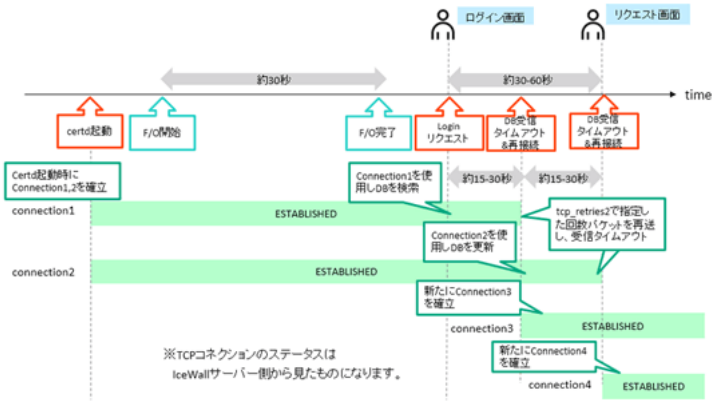
TCPパラメーター	説明	RHEL 7.1デフォルト値	備考
tcp_retries2	確立済のコネクションで応答がなかった場合のパケットの再送回数	6(回)	約130秒程度に相当

上記の表の通り、「tcp_retries2」の値により制御される確立済のコネクションの応答タイムアウト値は、デフォルトで実測したところ約15分程度と長く、DBアクセス失敗を検出する前にブラウザからの画面アクセスがタイムアウトしてしまいます。

本検証を行った環境では、Apacheのリクエストタイムアウト(Timeoutパラメーター)のデフォルト値は60秒でした。ユーザーのリクエスト画面がタイムアウトする前にDBコネクションの再接続を完了させるためには、再接続処理を60秒以内に完了させる必要があります。そのため、以下の表の通り関係パラメーターの調整を行いました。

区分	パラメーター	値	備考
OS (RedHat EL 7.1)	tcp_retries2	5(回)	
Webサーバー (Apache 2.4.6)	Timeout	60(秒)	デフォルト値
IceWall 認証モジュール	MAXDBCONNECT	2(本)	デフォルト値

「MAXDBCONNECT」が2の場合、IceWall認証モジュールの起動時にDBインスタンスの間に2本のコネクション(下図、Connection1及びConnection2)が確立されます。IceWall認証モジュールがログインリクエストを受け付けるとこれらのコネクションを使用してDBアクセスを試みますが、前述の通り、これらは使用できないため、再接続が試行されます。「tcp_retries2」を5回に設定するとIceWall認証モジュールがログインリクエストを受け付けてから、DBインスタンスとの間のコネクションを再確立試行するまでに約15秒から30秒かかりました。再確立試行する時点でDBインスタンスのフェイルオーバーは完了しているため、セカンダリのDBインスタンスとの間で瞬時に再接続が完了し、DB検索の処理が実施されます。その後ログイン時刻やログインステータス等の値を更新するためにDB更新の処理を試行しますが、この時先のDBアクセスと異なるコネクションが使用される場合は、さらにコネクションの再確立に約15秒から30秒かかりその後DB更新処理が実行されます。このケースでは、ユーザーがログインリクエストを送信してから60秒以内に処理が完了しているため、ユーザーには要求した画面が提示されます。



4. CloudFrontとの連携

4.1 CloudFront

本検証では、CloudFrontのオリジンサーバーとしてIceWallを導入したEC2インスタンスを指定し、IceWallの管理するバックエンドWebサーバーのコンテンツに対するキャッシュ動作を確認しました。

本検証で使用した設定は以下の通りです。

パラメーター		値	備考
Origin Settings	Origin Domain Name	EC2インスタンスに割り当てられたパブリック DNS(FQDN形式)	IceWallサーバー
	Origin Path		オリジンサーバーのドキュメントルートにマッピング
	HTTP Port	80	IceWallサーバーのdfw稼動ポート
Default Cache Behavior Settings	Allowed HTTP Methods	GET, HEAD, OPTIONS, PUT, POST, PATCH, DELETE	デフォルトはGET,HEAD
	Forward Headers	Whitelist/Host	デフォルトはNone
	Forward Cookies	Whitelist/IW_INFO	デフォルトはNone
	Object Caching	Use Origin Cache Headers	デフォルト値
	Default TTL	86400	デフォルト値

以下、CloudFrontのオリジンサーバーをIceWall SSOとする際にポイントとなるパラメーターについて説明します。

「Fowad Heders」に『whitelist/IW_INFO』を設定します。IceWall SSOによって認証を受けたアクセスリクエストには、認証済であることを示す認証Cookie(Cookie名:IW_INFO)が添付されます。この設定により、CloudFrontが認証CookieをIceWallサーバーへ送付するようになり、IceWall SSO側でのセッション管理が正常に行われるようになります。
またCloudFrontはIW_INFOの値が異なれば、それぞれの値に応じてコンテンツをキャッシュします。IceWallの認証Cookieは認証セッション毎に値が異なるため、既にCloudFront上にキャッシュされたコンテンツに対するリクエストでも認証セッションの異なるアクセスには、同じキャッシュ内容が使用されることはありません。

「Object Caching」に『Use Origin Cache Headers』を設定します。これで基本的にオリジンサーバーが返すコンテンツ内のCache-Controleヘッダーをはじめとするキャッシュに関連する情報に従いキャッシュが行われます。コンテンツにキャッシュ動作に関連する情報が含まれない場合、CloudFrontは内部に保持するデフォルト値に従いキャッシュを行います。

4.2 IceWallの設定

パラメーター		値	備考
dfw.conf	HOST	LOCALHOST=localhost:81	バックエンドサーバー
	SVRFILE	LOCALHOST,./sample.conf	上記バックエンドサーバーのホスト設定ファイル
	REV_PATH	REV_PATH=http://localhost:81/image/	認証対象外パス
sample.conf	RES_HEADER	#RES_HEADER=Pragma,NOTSEND	インストール初期状態からコメントアウト
		#RES_HEADER=Pragma,ADD,no-cache	
		#RES_HEADER=Cache-Control,NOTSEND	
		#RES_HEADER=Cache-Control,ADD,no-cache, no-store, max-age=0, private, must-revalidate	
		#RES_HEADER=Expires,NOTSEND	
		#RES_HEADER=Expires,ADD,Thu, 01 Jan 1970 00:00:00 GMT	
cert.conf	COOKIEEXP	1	認証Cookie(IW_INFO)に有効期限(Expores)を追加する
	LOMETHOD	0	COOKIEEXPに1を設定するために必須の設定

「dfw.conf」では、同一サーバー内の別ポート(81番)で稼働するWebサーバーをバックエンドWebサーバーとして設定しています(上記、HOSTパラメーター)。
また、バックエンドWebサーバー内の一部のパス以下に画像ファイルを配置し認証対象外パス(認証を必要とせずにアクセスを許可するコンテンツ)に設定しています(上記、REV_PATHパラメーター)。ユーザーはIceWallサーバー経由で認証対象外パス以下の画像ファイルにアクセスしても認証は求められません。

「sample.conf」では、バックエンドWebサーバーから受け取ったキャッシュ関連のヘッダーについて、IceWallサーバーが操作を行わずにそのままCloudFrontへ転送するよう、インストール時の初期値から設定を変更しました（上記、RES_HEADERパラメーター）。

「certd.conf」では、認証Cookie(IW_INFO)に有効期限の情報(Expires)が付加されるよう設定をしています（上記、COOKIEEXPパラメーター）。この設定により、CloudFront内のキャッシュの有効期限とIceWallの認証セッションの有効期限を一致させることができます。認証Cookieに有効期限の情報が付加されていないと、先にIceWallの認証セッションが期限切れで無効になった場合でもCloudFront側が有効期限内であればCloudFrontのキャッシュ内容がクライアントに返答される事になってしまいます。

4.3 動作検証

以下の検証を実施し、IceWall SSOの認証の状態に合わせたキャッシングが行われる事を確認しました。

1. 認証対象コンテンツへのアクセス時のキャッシング動作確認
2. 認証対象外コンテンツへのアクセスのキャッシング動作確認

4.3.1 動作検証：認証対象コンテンツへのアクセス

本検証の設定では、認証Cookieはファイルとして保存されるため、ブラウザセッション終了後も認証セッションの有効期限内であればリクエストに添付されます。また、有効期限が過ぎるとリクエストに添付されなくなるため、有効期限切れのリクエストに対しキャッシュが使用されることはありません。そのため、認証セッションが有効な間のみCloudFrontのキャッシュが利用されるようIceWallを構成することができます。

4.3.2 動作検証：認証対象外コンテンツへのアクセス

フォワーダー設定ファイルdfw.confのREV_PATH項目で指定したパス以下のコンテンツへのアクセスはIceWallを経由する際に認証を求められません。そのため認証Cookieが付加されないリクエストについては、コンテンツの返すキャッシュが使用され、キャッシュの有効期限の間はキャッシュが使用されることを確認しました。
ただし、一度認証対象のコンテンツにアクセスし、認証Cookieが発行されると、認証対象外のパスに対してもリクエスト先のURLが認証Cookieの送付範囲内であれば認証Cookieがリクエストに添付されます。IceWallでは認証対象外のリクエストと共に送付された認証Cookieの有効性の確認は行いませんが、この場合は認証対象コンテンツのキャッシングと同様の動作となります。

5. まとめ

IceWall認証基盤のプラットフォームとして、AWSを活用出来る事が確認出来ました。
ただし、Amazon RDS(MySQL/Multi AZ配置)やCloudFrontとIceWallを組み合わせて利用する場合は、AWSの仕様に合わせて、IceWall及び関連製品のパラメーター設計を行う必要があります。

ここで述べた内容な技術的観点に基づいて検証した結果を示したものでAWS環境での動作や性能を保証するものではありません。実際の構築に関しては、日本ヒューレット・パッカード株式会社またはIceWall販売パートナーへご相談ください。

2016.1.22　新規掲載

執筆者　日本ヒューレット・パッカード　テクノロジーコンサルティング事業統括
テクニカルコンサルタント　土居 恭子