



# 目次

セルフサービス分析はミッションクリティカル .....	3
高可用性を理解する .....	3
Tableau Server のスケーラビリティ .....	4
すぐに使える高可用性.....	4
Tableau Server の高可用性を理解する .....	5
汎用 .....	5
ゲートウェイ .....	5
アプリケーションサーバー.....	6
調整サービス .....	6
クラスタコントローラ.....	7
リポジトリ.....	7
バックグラウンダー .....	9
Data Server.....	9
キャッシュサーバー .....	9
データエンジン .....	9
ファイルストア .....	10
検索 & 参照.....	11
VizQL Server.....	11
サードパーティのモニタリングツールを統合する.....	11
プライマリサーバーノードのフェールオーバー .....	12
バックアッププライマリの構成 .....	13
クラスタのステータスをモニタリングする .....	14
アーキテクチャの考慮事項 .....	15
最適な構成を選択する.....	15
最小3ノードの高可用性の導入 .....	16
3ノード以上の導入 .....	17
Tableau Server を利用したディザスタリカバリ .....	18
最も簡単な DR 戦略 .....	18
サードパーティの DR ソリューション .....	18
Tableau の DR 機能 .....	18
高可用性のその先に .....	19

# セルフサービス分析はミッションクリティカル

現在、世界中の組織において、セルフサービス分析とデータドリブンな意思決定はますます標準となっています。ユーザーと意思決定者は、知りたい情報をリアルタイムで得るために、データへの迅速なアクセスとセルフサービスツールを必要としています。企業の経営陣はデータドリブンな意思決定の重要性を理解しており、毎日これらのシステムを活用しています。データを活用するには、基盤となるシステムに高い可用性が必要です。現在のチームと企業向けツールにとって、よりアクセスしやすく、より簡単に構成できるプラットフォームの機能が必要になっています。

Tableau Server 10.0は、将来のミッションクリティカルなセルフサービス分析を提供するプラットフォームです。セルフサービスのデータ探索を迅速に行えて、堅牢なガバナンスによってコンテンツとデータの信頼性が向上します。また、導入と管理が簡単で、あらゆる企業に合わせて拡張できます。このホワイトペーパーでは、Tableau Server 10.0が、高可用性を維持しながら規模に応じたセルフサービス分析を実行する仕組みをご紹介します。

## 高可用性を理解する

高可用性システムの目標は、システムのダウンタイムを最小限にすることです。可用性は、一般的に「9の数」で表されます。これは実際の稼働時間と期待される稼働時間の割合を効率的に測定した数値です。以下は、連続する9の数と年間ダウンタイムの対応表です。

連続する9の数	稼働率	年間ダウンタイムの合計
1	90%	36日と12時間
2	99%	3日と15時間
3	99.9%	8時間45分
4	99.99%	52分34秒
5	99.999%	5分15秒

図1: 典型的な可用性の指標と対応する年間ダウンタイム

通常、システム管理者は、許容されるダウンタイムのしきい値を決めるためにビジネスユーザーとサービスレベル契約 (SLA) を結びます。システム管理者は SLA に基づいて、その目標を満たすように導入するアーキテクチャを選択します。ほとんどのシステム管理者が、保守、アップグレード、パッチ適用のためのダウンタイムを計画します。さらに、計画外ダウンタイムと呼ばれる、予期しない障害が発生する可能性もあります。管理者は当然、ハードウェアまたはソフトウェアのアップデートのために計画的な保守を実施する必要があります。目標は、計画外ダウンタイムを最小限にすることです。

ユーザーがデータをすぐに表示して理解できることは非常に重要です。一方で、ビジネスインテリジェンスシステムの可用性を脅かす出来事はいつでも発生する可能性があります。それは、ハードウェア、ソフトウェア、ネットワークに関連するエラーの場合もあれば、ヒューマンエラーの場合もあります。コンポーネント障害の発生時に、少なくとも Tableau Server 10.0 のプロセスでは、自動的に再起動され、システムの稼働状態が保たれます。複数ノードを導入して適切に構成すれば、冗長プロセスを使用したサーバーの高可用性 (HA) を実現できます。Tableau が他のシステムよりも優れているのは、こうした高可用性のある分析環境の設定および構成を簡単に行えることです。

## Tableau Server のスケーラビリティ

Tableau Server は、スケールアップおよびスケールアウトするように設計されています。大規模な組織にエンタープライズクラスの導入における安定性を保証するとともに、小規模なチームにとって魅力的なシンプルさと使いやすさも兼ね備えています。Tableau Server は環境に応じて 1 台または複数のコンピューターで実行でき、同じノードで 1 つまたは複数のコンポーネントプロセスを実行できるため、ユーザーの需要と高可用性の要件を両方満たすことができます。

Tableau では、Tableau Public および Tableau Online といった、いくつかのクラウドスケールの Tableau Server を社内に導入して管理しています。Tableau Public はカスタマイズされた Tableau Server を導入したもので、毎週世界中の何百万もの一般の閲覧をサポートしています。Tableau では、エンジニアリングおよびリリースプロセスの一環として安定性と品質を向上するために、企業のお客様へのリリースに先駆けて Tableau Server ソフトウェアのベータバージョンを Tableau Public に導入しています。

## すぐに使える高可用性

Tableau Server では、簡単に高可用性を導入して構成できます。既定のインストールはわずか数分です。クラスタ内のプライマリコンピューターと追加のワーカーノードすべてに Tableau Server をインストールし、次に構成ユーティリティを使用して高可用性のあるクラスタを構成および形成できます。

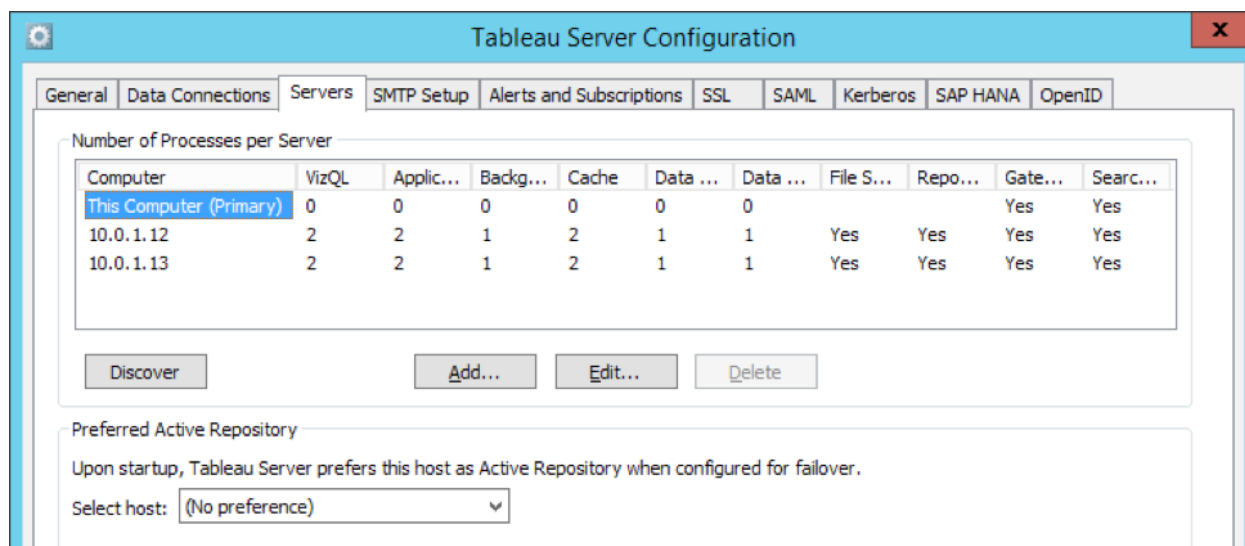


図 2: Tableau Server の構成ユーティリティを使用して、クラスタ内のノードの数と構成、ノードごとのプロセスを柔軟に決められる

# Tableau Server の高可用性を理解する

Tableau Server には、エンドユーザーがシステム全体に確実にアクセスできるようにするための専用プロセスが複数あります。このセクションは、Tableau Server のコンポーネントとその機能について精通していることを前提としています。まだよく知らない場合は、最初に『[Tableau サーバー管理者ガイド](#)』をお読みになることをお勧めします。

Tableau Server で高可用性を実現させる方法を理解するには、Tableau Server の各コンポーネントにおいて高可用性を実現させる方法を理解する必要があります。言い換えれば、Tableau Server クラスタ全体で高可用性を保って単一障害点に対する冗長性を持たせるには、すべてのコンポーネントにおいて高可用性を実現させる必要があります。それでは、コンポーネントを1つずつ確認していきましょう。

## 全般

Tableau Server には、すべてのサーバープロセスを再起動する自動化機能が組み込まれています。この自動化機能によって、失敗したサーバープロセスは自動的に再起動され管理者に通知されるため、高可用性を維持できます。この自動化が実行されるには、Tableau Server を構築しているハードウェアまたは仮想マシンが正常状態である必要があります。

ノード全体の障害から保護するには、固有のサーバープロセスを構成して、それらがクラスタ内の異なるノード間で冗長性を持つようすることが重要です。この冗長性は、プライマリノードでしか実行できないライセンス発行サービスを除いて、他のすべてのプロセスにおいて確保できます。このプロセス制限の対応方法については、「[プライマリサーバーノードのフェールオーバー](#)」セクションで詳しく説明します。

## ゲートウェイ

Tableau Server 8.1 以降、ゲートウェイプロセスは、プライマリ Tableau Server ノードだけでなく、Tableau Server クラスタ内のすべてのノードで実行できるようになりました。これは、HA にとって重要な意味があります。

Tableau Server 8.1 より前のバージョンでは、ゲートウェイプロセスを実行するように構成できるノードは1つだけでした。そして、その1つのプロセスが失敗した場合、Tableau Server ではクラスタの外部との通信ができなくなり、内部通信の多くにも混乱が生じていました。Tableau Server 10.0 では、複数のノードでゲートウェイプロセスを実行するように構成している場合、少なくとも1つのゲートウェイプロセスが実行されていればサーバーとの正常な通信を維持できます。

## ゲートウェイ障害のリスクを軽減する

ゲートウェイの高可用性を達成する鍵は、Tableau Server クラスタに複数のノードを用意し、複数のノードでゲートウェイプロセスを実行するように構成することです。Tableau では、すべてのノードでゲートウェイプロセスを構成することをお勧めしています。こうすることで、ゲートウェイプロセスが単一障害点となってサービスが利用できなくなるリスクを軽減できます。

ゲートウェイプロセスが失敗するとどうなりますか。前述のように、ゲートウェイプロセスが 1 つも実行されていなければ、Tableau Server クラスタ全体が利用できなくなります。他のゲートウェイプロセスが実行されていれば、機能しているゲートウェイに対するリクエストは正常に処理されます。ただし、他に機能しているゲートウェイがある場合でも、障害のあるゲートウェイによって受け取られたリクエストはリダイレクトされず、失敗したままになります。失敗したゲートウェイプロセスは自動的に再起動されるため、コンピューター自体が機能している場合は、失敗したゲートウェイプロセスが再度実行されてリクエストの処理が再開されます。

ゲートウェイ障害に対してさらに堅牢となるように構成するには、Tableau Server クラスタを外部ロードバランサー (ELB) の後ろに配置して、リクエストが機能しているゲートウェイプロセスだけにルーティングされるようにすることをお勧めします。『Tableau サーバー管理者ガイド』には、この設定に関する詳細なガイドが記載されています。

## アプリケーションサーバー

アプリケーションサーバーで高可用性を確保することは簡単です。Tableau Server クラスタの各ノードにアプリケーションサーバーインスタンスを構成するだけです。

アプリケーションサーバープロセスが失敗するとどうなりますか。そのインスタンスによって処理されていたリクエストは失敗しますが、その後のリクエストは実行中の他のアプリケーションサーバープロセスにルーティングされます。障害のあるアプリケーションサーバーを含むノードがまだ実行中である場合、失敗したプロセスは数秒で再起動されます。

## 調整サービス

調整サービスは、Tableau Server の基本的なインストールの一部です。下の表のとおり、調整サービスがインストールされるノードの数は、クラスタ内のノードの数に応じて異なります。

クラスタ内のノード数	コーディネーションサービスがインストールされるノード数
1 - 2	1
3 - 4	3
5 以上	5

調整サービスプロセスは、クラスタの最初の「*n*」ノード (プライマリノードを含む) に連続してインストールされます。ここで「*n*」は、上記の表に基づく調整サービスがインストールされるノードの数を指します。

## クォーラムを理解する

実行中の調整サービスプロセスの数が、構成されている調整サービスプロセスの合計数に基づくクォーラムに達していない場合、Tableau Server が完全に機能停止します。クォーラムは、絶対多数と同じ意味です。つまり、3 つまたは 4 つのコンピューターのクラスタでは、最大で 1 つのノード (1 つの調整サービスインスタンス) の損失を許容できます。5 つまたは 6 つのコンピューターのクラスタでは、最大で 2 つの調整サービスプロセスの損失を許容できます。

2 つのノードのみのクラスタでは、1 つの調整サービスプロセスの損失も許容できないことに注意してください。自動フェールオーバーを含む完全な高可用性に最低 3 つのノードが必要なのはこのためです。

調整サービスプロセスが失敗するとどうなりますか。残りの調整サービスプロセスがクォーラムに達している場合は何も起こりません。機能している調整サービスプロセスの数がクォーラムに達していない場合、基盤となる PostgreSQL データベースの参照整合性を保護するために、Tableau Server クラスタ全体が利用できなくなります。

コンピューター自体に問題がない場合、失敗した調整サービスプロセスは自動的に再起動されます。

## クラスタコントローラ

クラスタコントローラも Tableau Server 10.0 の基本的なインストールに含まれます。インストール中に、クラスタの各ノードに 1 つのクラスタコントローラプロセスが構成されます。明示的な設定は必要ありません。

クラスタコントローラプロセスが失敗するとどうなりますか。同じノードの他のすべての Tableau Server コンポーネントが利用できなくなり、Tableau Server のステータスページに「使用不可」と表示されます。また、そのノードで実行中のリポジトリプロセスも利用できなくなります (この影響について詳しく理解するには、下記の「リポジトリ」セクションをご覧ください)。クラスタコントローラプロセスの失敗から保護するために、個々のサーバーコンポーネントに冗長性を確保し、少なくともクラスタ内の 2 つの異なるノードで実行するようにしてください。

コンピューター自体に問題がない場合、失敗したクラスタコントローラプロセスは自動的に再起動されます。クラスタコントローラが再起動されると、そのノードに設定されているリポジトリプロセスも再起動されます。

## リポジトリ

リポジトリは、Tableau Server が機能するために不可欠なデータベースです。完全に機能しているリポジトリがない場合、Tableau Server クラスタ全体が利用できなくなります。どの時点でも、完全に機能しているリポジトリは 1 つしか使えません。このリポジトリは「アクティブ」リポジトリと呼ばれます。リポジトリが関係するすべての操作はこのアクティブリポジトリで実行されます。

可用性を向上させるために、クラスタ内の異なるノードに追加の「パッシブ」リポジトリを作成して Tableau Server を構成できます。アクティブリポジトリのコンテンツは、常にパッシブリポジトリに送られます。高可用性を考慮して構成されたクラスタでは、アクティブリポジトリに障害が発生した場合、サーバーの可用性を保つためにパッシブリポジトリのステータスが自動的にアクティブに変わります。高可用性を必要としているお客様には、パッシブリポジトリを構成することを強くお勧めします。クラスタ全体における最大のリポジトリ数は 2 つのみ (1 つのアクティブと 1 つのパッシブ) で、その 2 つを同じノードに配置することはできません。

クラスタコントローラは、リポジトリの起動、シャットダウン、アクティブからパッシブへのフェールオーバーを管理します。クラスタコントローラプロセスが失敗するとリポジトリにも問題が発生するのはこのためです。リポジトリを起動したクラスタコントローラプロセスが失敗すると、リポジトリも失敗します。

リポジトリプロセスが失敗するとどうなりますか。それは状況によって異なります。以下のリストにさまざまなケースをまとめたものです。

- パッシブリポジトリに障害が発生した場合、ユーザーに影響はありません。アクティブリポジトリが機能しているため、すべてが引き続き動作します。バックグラウンドでパッシブリポジトリが再起動され、データ複製が再開されます。ただし、パッシブリポジトリとアクティブリポジトリが完全にもう 1 度同期されるまで少し時間がかかる場合があります。

- アクティブリポジトリに障害が発生し、完全に同期されたパッシブリポジトリがない場合は、アクティブリポジトリが再起動されるまで Tableau Server が利用できなくなります。システムによって自動的に再起動が試行されますが、障害の原因によっては再起動できない場合もあります。

注: つまり、現在使えるのがアクティブリポジトリのみの場合は、Tableau Server に高可用性があると見なすことはできません。同期されたパッシブリポジトリがない場合は、アクティブリポジトリがシステム全体の単一障害点となります。

- アクティブリポジトリに障害が発生したときに、完全に同期されたパッシブリポジトリがあり、クラスタを HA の構成にしている場合は、パッシブリポジトリへのフェールオーバーが自動的に始まります。フェールオーバー後は、それまでのパッシブリポジトリが新しいアクティブリポジトリとなります。システムによって、障害のあったアクティブリポジトリは新しいパッシブリポジトリとして再起動され、同期が開始されます。また、その他の関連するプロセスも自動的に再起動され、新しく昇格したアクティブリポジトリを認識して再接続されます。この短い再起動期間は、サービスが一時的に中断されます。ただし、これは自動化された手順であるため、可用性を維持するために管理者が介入する必要はありません。手動でパッシブリポジトリをアクティブリポジトリに昇格させる場合は、`tabadmin failoverrepository` コマンドを使用して実行できます。



## バックグラウンダー

バックグラウンダーサービスの高可用性を実現するには、複数のバックグラウンダーサービスをクラスタ内の複数のノードで実行するように構成します。複数のバックグラウンダーを実行する場所と方法を決めるときは、他のサーバープロセスに基づいて各コンピューターの利用可能なキャパシティを考慮します。

バックグラウンダープロセスがダウンしたらどうなりますか。バックグラウンダーによって処理されていたジョブが失敗し、再試行されません。ただし、ほとんどのバックグラウンドジョブは定期的に行われるように計画されているため、機能しているバックグラウンドプロセスによって次の予定時に同じバックグラウンドタスクが開始され、正常に行われます。

コンピューターに問題がない場合、失敗したバックグラウンダープロセスは自動的に再起動されますが、失敗したジョブは再起動されません。

## Data Server

Data Server の高可用性を実現するには、1 つまたは複数の Data Server プロセスをクラスタ内の複数のノードで実行するように構成します。

Data Server プロセスが失敗するとどうなるでしょうか。プロキシを通して Data Server プロセスによって実行されているクエリが失敗し、ビューのレンダリングの失敗につながります。機能している Data Server があり、再ルーティングされたリクエストを受け取れる場合、失敗した操作の再試行など、その後のリクエストは成功します。

Tableau Server は Data Server がなくても機能します。ただし、実行中の Data Server がいない場合、クラスタはワークブックを外部データソースにプロキシ接続できなくなります。いずれかのデータソースに対して Data Server を使っていないビューは、引き続き正常に機能します。

## キャッシュサーバー

キャッシュサーバーは、共有の外部クエリキャッシュとして機能します。これはキー/バリューのペアのキャッシュで、以前のクエリの情報を保持し、今後のリクエストの速度を向上します。つまり、キャッシュサーバープロセスが 1 つ利用できなくなった場合、またはすべてが利用できなくなった場合でさえ、影響は比較的少なくて済みます。Tableau Server は引き続き機能しますが、以前キャッシュに保存した結果が利用できないため、アクションに比較的長い時間がかかる場合があります。クエリが再度実行されると、再起動されたキャッシュサーバーに再び保存されて、エンドユーザーの操作速度が向上します。実際には、キャッシュサーバーは可用性に影響しません。ただし、さまざまなエンドユーザーのパフォーマンスのシナリオに影響があります。他のすべてのプロセスと同じように、コンピューター自体に問題がない場合、キャッシュサーバーも自動的に再起動されます。

## データエンジン

データエンジンコンポーネントは、インメモリ分析を使用するときにデータ抽出のロードおよびクエリを実行します。データエンジンの高可用性は、1 つまたは複数のデータエンジンプロセスをクラスタ内の複数のノードで実行するように構成するだけで実現できます。すべてのデータエンジンはアクティブ/アクティブモードで実行され、まったく同じ機能を実行します。データエンジンプロセスを実行するように構成されているノードは、必ずファイルストアプロセスも実行するように構成されることに注意してください。ファイルストアプロセスは、抽出ファイルの保存および複製を管理します。詳しくは次のセクションで説明します。

データエンジンプロセスがダウンしたらどうなるでしょうか。現在そのデータエンジンプロセスで実行されているクエリが失敗し、結果としてビューのレンダリングまたは抽出の更新が失敗します。同じ操作を再度実行すると、機能している他のデータエンジンに自動的に再割り当てされます。

コンピューター自体に問題がない場合、失敗したデータエンジンプロセスが自動的に再起動され、再開されます。

## ファイルストア

前述したように、1 つまたは複数のデータエンジンプロセスを実行しているノードでは、ファイルストアプロセスが自動的に開始されます。ファイルストアプロセスは、ノード間の抽出ファイルの保存および複製を管理します。

## ファイルストアの仕組み

ユーザーが初めて Tableau Server に抽出ファイルをパブリッシュするか、抽出の更新が行われると、システムに抽出ファイルが作成されます。このようなイベントの直後に、単一のノードにある単一のファイルストアに抽出が保存されます。特定の抽出ファイルには冗長性がなく、単一障害点となるため、そのままでは高可用性があるとはいえません。ファイルストアプロセスは相互に通信し、ローカルの抽出をクラスタ内の他のすべてのファイルストアノードにすばやく複製します。ファイルストアプロセスは、クラスタネットワークのリソースに応じて可能な限りすばやくファイルをコピーするように設計されていますが、抽出のサイズによって必要な時間は異なります。クラスタ内の複数のノードでコピーが利用可能になれば、その抽出ファイルは高可用性があると考えられます。

ファイルストアプロセスが失敗するとどうなりますか。2 種類の結果があります。

- 関連するノード間の抽出ファイルのコピーが停止します。
- 関連するノードにおける不要になった抽出ファイルの削除が一時停止します。通常、この削除プロセスは「抽出の刈り取り」と呼ばれます。

抽出の刈り取りが一時停止しても、すぐに影響はありません。不要な抽出ファイルが蓄積することにより、そのノードのディスク領域が消費されるだけです。最終的にはこれが問題につながりますが、適切なサイズのノードにはディスク領域に十分なバッファがあります。

ファイルが複製されないということは、障害のあるファイルストアノードに追加された新しい抽出ファイルをクラスタ内の他のファイルストアノードから利用できず、また他のノードの抽出ファイルを障害のあるファイルストアノードから利用できないことを意味します。ファイルストアプロセスが再起動されると、すべてのノードにあるすべてのファイルストアを同期することでシステムが自動的に修正されます。

コンピューター自体に問題がない場合、失敗したファイルストアプロセスは自動的に再起動されます。ファイルストアプロセスはすぐに機能を再開し、障害発生中に追加されたかその後に追加されたかにかかわらず、すべてのファイルが同期されます。

## 検索 & 参照

検索 & 参照プロセスで高可用性を実現することは簡単です。検索 & 参照プロセスを複数のコンピューターで実行するようにシステムを構成することです。

検索 & 参照プロセスが失敗するとどうなりますか。Tableau Server の多くの機能が利用できなくなり、ユーザーはシステムにログインできますが、ワークブックコンテンツが不足しているように見えます。実際は、コンテンツは不足しておらず、検索結果に返されなくなるだけです。検索 & 参照プロセスが再起動されるとコンテンツが表示されるようになります。複数の検索 & 参照プロセスが複数のノードで構成および実行されている状態で障害が発生した場合、失敗した検索 & 参照プロセスに対するリクエストも失敗しますが、その後のリクエストは機能している検索 & 参照プロセスにルーティングされます。各検索 & 参照プロセスで、クラスタ内のすべてのノードにわたるインデックスが作成されるため、1つを除いてすべての検索 & 参照プロセスが失敗した場合でもすべてのノードに結果が返されます。

## VizQL Server

VizQL Server プロセスの高可用性は、1つまたは複数のインスタンスを複数のコンピューターで実行するように構成するだけで実現できます。

VizQL Server プロセスが失敗するとどうなりますか。VizQL Server プロセスが1つのみで、それが失敗した場合、Tableau Server はビューのレンダリングができなくなります。高可用性を達成するには、冗長性のある VizQL プロセスを構成する必要があります。ほとんどの一般的な構成では、各ノードで2つから4つの VizQL Server プロセスを実行します。これにより、高可用性とスケーラビリティのニーズを同時に満たすことができます。複数の VizQL Server プロセスを実行していて1つのプロセスが失敗した場合、そのプロセス失敗時のリクエストが失敗し、セッションデータが損失します。その後のリクエストは、Tableau Server クラスタ内で機能している他の VizQL Server プロセスにルーティングされます。

ここまで、各サーバープロセスの失敗時の動作と、失敗のリスクを軽減して Tableau Server クラスタ全体の高可用性を維持する方法を確認しました。失敗の各シナリオに対して計画を立てるだけでなく、今までに起こったプロセスの失敗を知るためにクラスタを積極的にモニタリングする必要があります。

## サードパーティのモニタリングツールを統合する

Tableau Server に組み込みのメカニズムを使ってシステムの状態をモニタリングするだけでなく、各プロセスのステータスをコンピューターで読み込める XML 形式でリモートで受け取ることもできます。サーバーへのリモートアクセスを有効にし、インスタンスに対する適切な URL ([http://<my\\_tableau\\_server>/admin/systeminfo.xml](http://<my_tableau_server>/admin/systeminfo.xml)) を使う必要があります。

これによって `status.xml` ファイルが返されます。このファイルを解析してサーバーのステータスを特定したり、他のシステムまたはモニタリングツールに統合したりできます。

以下は、サーバーステータス URL へのリモートアクセスによって取得した `status.xml` ファイルをコンピューターに読み込んだ XML のサンプルです。

```
<systeminfo xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <machines>
    <machine name="my_tableau_server">
      <repository worker="my_tableau_server:8060" status="Active"
preferred="false"/>
      <dataengine worker="my_tableau_server:27042" status="Active"/>
      <applicationserver worker="my_tableau_server:8600" status="Active"/>
      <apiserver worker="my_tableau_server:8000" status="Active"/>
      <vizqlserver worker="my_tableau_server:9100" status="Active"/>
      <dataserver worker="my_tableau_server:9700" status="Active"/>
      <backgrounder worker="my_tableau_server:8250" status="Active"/>
      <gateway worker="my_tableau_server:80" status="Active"/>
      <searchandbrowse worker="my_tableau_server:11000" status="Active"/>
      <cacheserver worker="my_tableau_server:6379" status="Active"/>
      <filestore worker="my_tableau_server:9345" status="Active"
pendingTransfers="0" failedTransfers="0" syncTimestamp="2015-02-
27T20:30:48.564Z"/>
      <clustercontroller worker="my_tableau_server:12012"
status="Active"/>
      <coordination worker="my_tableau_server:12000" status="Active"/>
    </machine>
  </machines>
  <service status="Active"/>
</systeminfo>
```

## プライマリサーバーノードのフェールオーバー

プライマリ Tableau Server ノードとは、最初に Tableau Server がインストールされたサーバーを指します。そのプライマリサーバーは、すべての機能が揃ったサーバーインストールを実行できるだけでなく、固有のライセンス管理プロセスおよび他の管理機能があるという点で独特です。大規模な導入では、プライマリサーバーに特有の必要最小限の管理機能のみをインストールし、ワーカーノードと重複するプロセスを除外することもできます。Tableau Server では非常に柔軟な導入が可能です。Tableau Server のプロセスは 72 時間ごとにプライマリサーバーをポーリングしてライセンスを確認することに注意してください。ライセンスの確認時にプライマリサーバーにアクセスできない場合、ライセンス確認が失敗し、Tableau Server の導入が「無許可」状態となって無効になります。

ただし、実際には、ハードウェア、OS、ネットワーク障害など、プライマリサーバーは Tableau Server ソフトウェアと関連のないさまざまな理由で利用できなくなる可能性があります。そのような場合でも、クラスタを外部ロードバランサーの後ろに配置し、クラスタ内のすべてのノードにゲートウェイを作成することで、残りのクラスタは引き続きサービスリクエストに対応できます。ただし、ライセンスの確認期限が切れてもプライマリノードが無効な場合は、クラスタ全体のライセンスがなくなります。このような問題に備えるために、プライマリノードのウォームバックアップを準備してください。

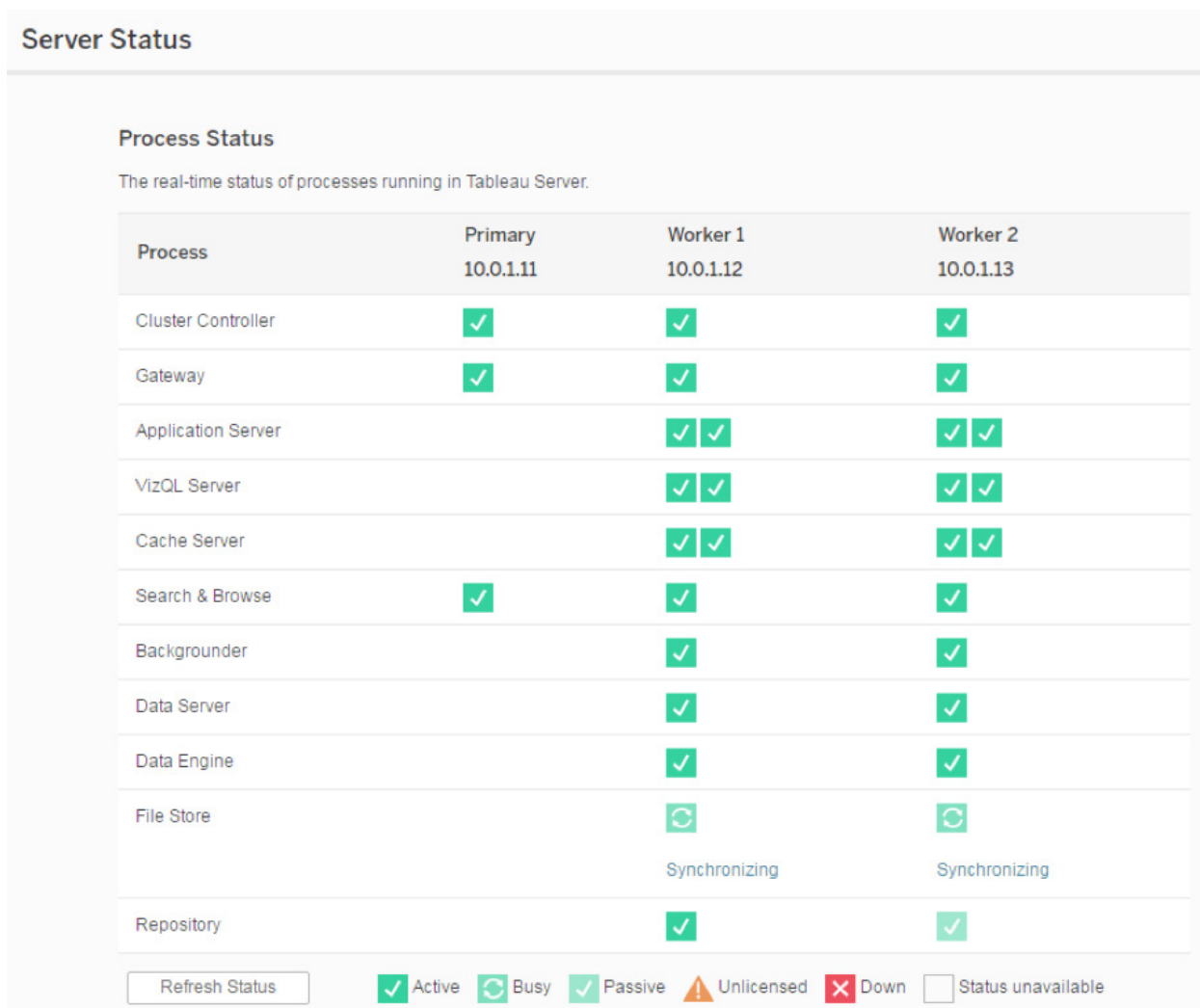
## バックアッププライマリの構成

プライマリサーバーに障害が発生した場合、サーバーに障害を起こしている問題が解決されるまで管理タスクにダウンタイムが発生します。このため、プライマリサーバーのノードがダウンした場合に備えて「バックアップ」プライマリとして、専用のバックアップマシンを割り当てておく必要があります。バックアッププライマリを構成して準備ができて、バックアッププライマリの電源を入れたりクラスタに接続したりしないでください。これは、ライセンス発行と管理機能を引き続きプライマリサーバーのノードに対して動作させるためです。バックアッププライマリサーバーの設定および準備方法については、『[Tableau サーバー管理者ガイド](#)』をご覧ください。

プライマリサーバーに障害が発生した場合に備え、バックアッププライマリサーバーへの切り替え手順についても『[Tableau サーバー管理者ガイド](#)』に記載されています。この手順を実行することで、残りのクラスタに変更に関するアラートが送信され、管理リクエストとライセンス発行リクエストが新しいプライマリサーバーに自動的に再ルーティングされます。また、これらの手順を自動化することもできます。

## クラスタのステータスをモニタリングする

システム管理者は、Tableau Server のステータスページで Tableau Server 10.0 のプロセスのステータスをモニタリングできます。システム管理者はすべてのノードのサーバープロセスを表示することでクラスタ全体の状態を把握し、アクティブおよびパッシブリポジトリのステータスも確認できます。



The screenshot displays the 'Server Status' page with a 'Process Status' section. Below the title, it states 'The real-time status of processes running in Tableau Server.' The main content is a table with columns for 'Process', 'Primary 10.0.1.11', 'Worker 1 10.0.1.12', and 'Worker 2 10.0.1.13'. The 'Process' column lists various services like Cluster Controller, Gateway, Application Server, etc. The status is indicated by green checkmarks (Active/Passive), a green refresh icon (Synchronizing), or a red X (Down). A legend at the bottom explains these icons: Active (green check), Busy (green refresh), Passive (green check with vertical line), Unlicensed (orange triangle), Down (red X), and Status unavailable (grey square). A 'Refresh Status' button is also present.

Process	Primary 10.0.1.11	Worker 1 10.0.1.12	Worker 2 10.0.1.13
Cluster Controller	✓	✓	✓
Gateway	✓	✓	✓
Application Server		✓ ✓	✓ ✓
VizQL Server		✓ ✓	✓ ✓
Cache Server		✓ ✓	✓ ✓
Search & Browse	✓	✓	✓
Backgrounder		✓	✓
Data Server		✓	✓
Data Engine		✓	✓
File Store		🔄 Synchronizing	🔄 Synchronizing
Repository		✓	✓

図 3: サーバーステータスの管理ページには、各ノードの各プロセスのステータスが表示される

Tableau Server では、Tableau Server 構成ユーティリティを使って、システムに障害が発生した場合にシステム管理者にメールアラートを送信するように設定することもできます。また、このユーティリティを使って、ディスク領域の容量問題について事前に警告を送信することもできます。

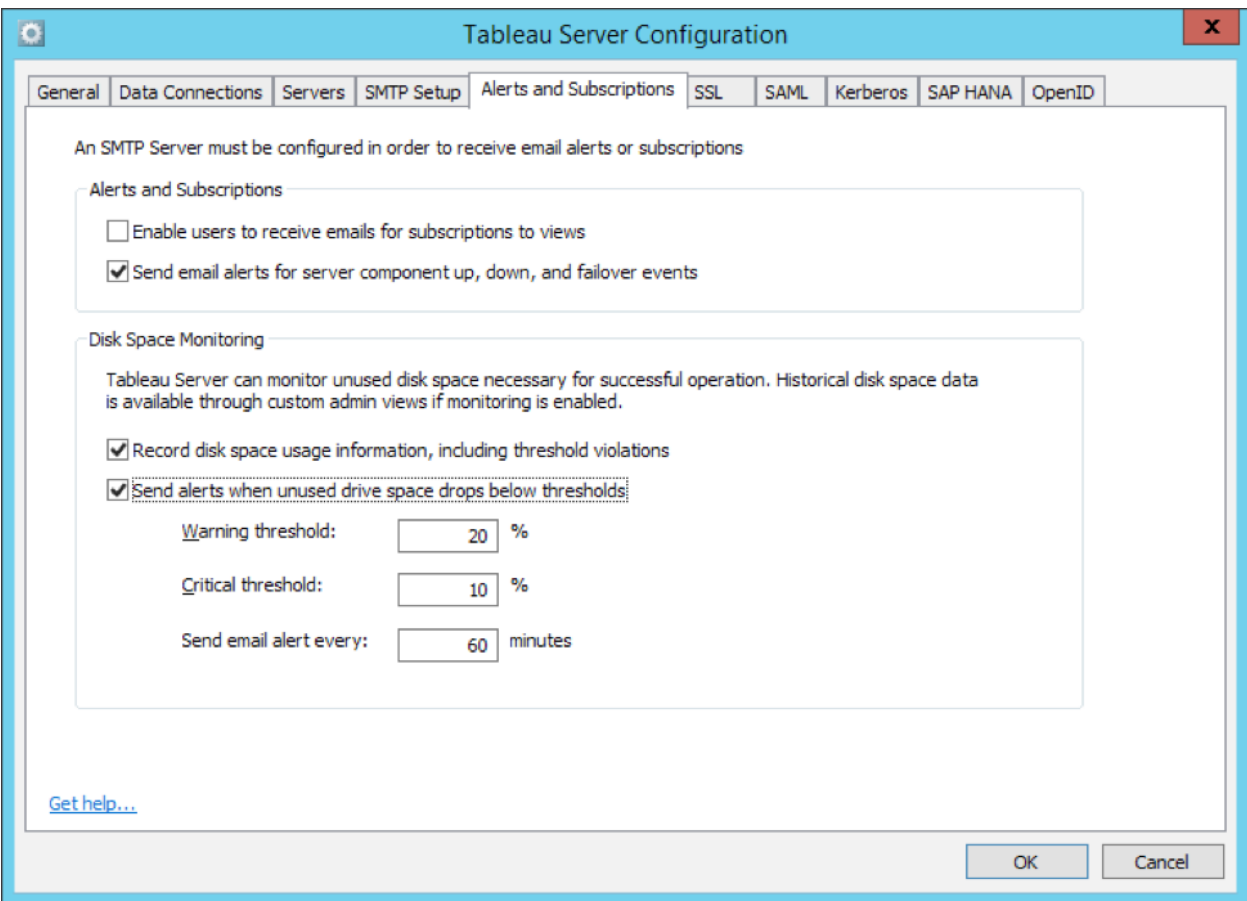


図 4: Tableau Server 構成ユーティリティでは、システム障害に関するアラートを設定できる

## アーキテクチャの考慮事項

高可用性を適切に実装するには、組織のアップタイムの目標と、必要としているサービスレベルを理解し、それに沿って冗長性を計画する必要があります。

### 最適な構成を選択する

最適なクラスタ構成は、ビジネスのニーズと、組織で利用できるリソースを考えてあわせて決める必要があります。高可用性を実現する構成はいくつか考えられますが、さまざまなサイズのクラスタのトレードオフを理解して、現在の環境に最適な構成を選択することが重要です。さらに、対称性のあるコンピューター構成（各ワーカーノードに同じコンポーネントをすべて含める）または非対称のコンピューター構成（各ノードにあるコンポーネントの数と種類が異なる）のいずれかを選択できます。通常、ワーカーノードを対称性のある構成にすると、ノードを複製してクラスタに追加することが簡単になります。ただし、構成を計画する際は、1つのアクティブリポジトリと1つのパッシブリポジトリという制限があることに留意してください。

## 最小3ノードの高可用性の導入

前述したように、完全な HA モードで実行するには、クラスタ内に最低3ノードが必要です。3ノードのクラスタは、大規模な導入を始める方法としても最適です。1つのノードに障害が発生した場合でも、残りの2つのアクティブなノードによってクォーラムを達成できます。下の図5は、管理およびライセンス発行機能に関するプライマリノードの役割を示しています。また、Tableau Server のデータおよびビジュアライゼーション機能を提供する追加の2つのワーカーノードを示しています。このようなコンポーネントの実際の組み合わせは、HA のニーズに加え、スケーラビリティのニーズによって異なる場合があります。たとえば、図では各ワーカーノードにつき VizQL プロセスは1つのみですが、実際は1つのワーカーノードで2つ以上の VizQL Server プロセスを実行して HA とスケーラビリティのニーズをサポートできます。さらに、プライマリノードにより多くのコア容量がある場合は、VizQL Server またはデータエンジンとファイルストアの組み合わせなどのサーバープロセスを追加してスケーラビリティを向上させることもできます。

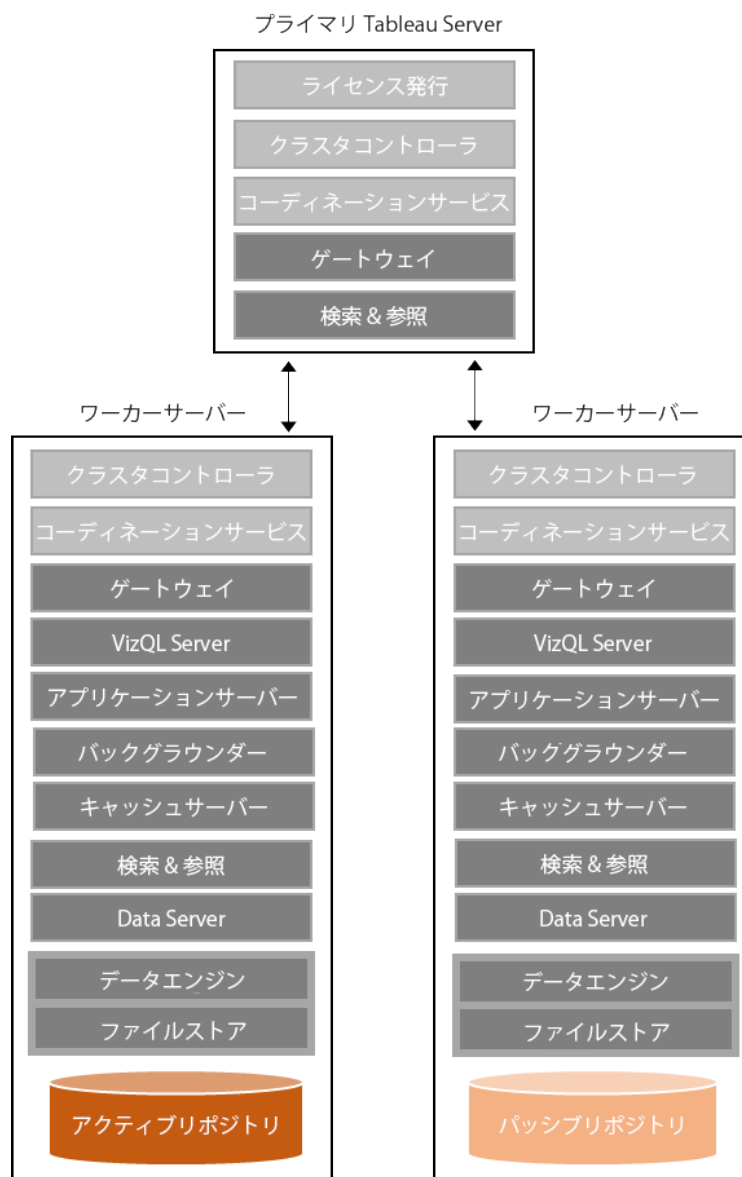


図5: 最小3ノードのHA導入の例



複数のコンピューターにデータエンジンをインストールすることで、ハードウェアに障害が発生した場合に備えて冗長性を向上し、バックアップによるすばやい復旧が可能になります。さらに冗長性を高める必要がある場合は、各ワーカーノードに追加のデータエンジンを導入できます。

プライマリサーバーでのプロセスは、いずれかのワーカーノードがダウンしても影響を受けません。ただし、ワーカーはプライマリに依存します。この構成では、アクティブリポジトリであるワーカーに障害が発生した場合、パッシブリポジトリであるワーカーが自動的にアクティブになります。ただし、前述のように、バックアッププライマリを構成し、プライマリサーバーに障害が発生した場合でもすばやく復旧できるように準備しておくことが最善です。さらに、組織で外部ロードバランサーにアクセスできる場合は、そのELBの後ろにクラスタを導入してエンドユーザーの可用性を向上させることをお勧めします。高可用性のある導入を始めたばかりなら、この構成から始めるのは良い方法です。

### 3 ノード以上の導入

3 ノード以上を使用する場合、クラスタに奇数のノードを導入することが、高可用性を向上させるために適切なトポロジです。偶数のノードのクラスタにおけるクォーラムに必要なノード数は、1 つノードが少ないクラスタと同じです。たとえば、2 つのノードが機能停止した場合、3 ノードのクラスタと 4 ノードのクラスタは同じように影響を受け、クォーラムを満たすことができずシャットダウンされます。

3 ノードのクラスタに 4 番目のノードを加えて偶数にすることでワークロードをより多くのノードに分散して、スケーラビリティを向上できるだけでなく、冗長プロセスが追加されることで完全なクラスタ停止のリスクを軽減できます。

他の考慮事項は関係なくダウンタイムを最小限にすることだけを考えているなら、5 つ以上のノードの導入アーキテクチャをお勧めします。このホワイトペーパーでは、初めての導入に適している 3 ノードのアーキテクチャをお勧めしていますが、ミッションクリティカルな高可用性要件のある全社的な大規模導入の場合は、基準として 5 つ以上のノードを使用することを検討してください。

抽出と抽出の更新を頻繁に行う組織では、一般的に、専用のワーカーノードにバックグラウンダープロセスを取り入れた導入アーキテクチャを採用します。抽出の更新のワークロードは VizQL Server プロセスがサポートするビジュアライゼーションのワークロードと干渉することがあるため、バックグラウンダープロセスを専用のワーカーマシンに割り当てることで、これらの異なる 2 つのワークロード間におけるリソースの競合を避けることができます。クラスタ内の複数のノードにわたりプロセスの冗長性を確保するために、バックグラウンダーのみのワーカーは一般的にペアで導入されます。

前述のように、高可用性は重大な障害によるダウンタイムを削減します。サービスレベルの要件を満たし、高可用性に対応するトポロジを導入しても、すばやい復旧が要求される障害に直面する可能性があります。災害やハードウェア障害に備えて、ディザスタリカバリの目標と目的に基づいてシステム復旧の計画を立てておく必要があります。

# Tableau Server を利用したディザスタリカバリ

Tableau 環境のディザスタリカバリ (DR) を計画する場合、主な考慮事項は 2 つあります。(1) ビジネスが許容できる復旧プロセスの一部としてのダウンタイムと、(2) ビジネスが許容できるデータ損失です。これは、復旧時間目標 (RTO) と復旧ポイント目標 (RPO) という観点で定義されます。RTO とは、システムが完全復旧するまでにどのぐらいの時間が経過するのかを測るもので、RPO とは、ビジネスにおけるデータ損失の許容度を測るものです。

RTO と RPO のしきい値を定義するときは、妥協点を理解することが重要で、この妥協点とはほとんどの場合が財務上のものです。詳細にかかわらず、システムをすばやくオンラインに復旧させるために必要なインフラストラクチャとスタッフを考慮すると、非常に低い復旧時間目標を実現するにはコストがかかります。インフラストラクチャと対応するスタッフにかかるコストを軽減するには、大幅に長い RTO (より長いダウンタイム) を許容する必要があります。このトレードオフは Tableau Server だけではなく、すべてのサーバーソフトウェアの導入に当てはまります。

データ損失への許容度は、特に個々のアプリケーションと、そのアプリケーションでのデータ変更率によって決まります。たとえば、24 時間の RPO とは、損失するのは最大でも過去 24 時間以内のデータであることを意味します。

## 最もシンプルな DR 戦略

多くの場合、最もシンプルで効果的な DR 戦略は、夜間にバックアップを実行し、ウォームまたはホットのスタンバイサーバーマシンを復旧データセンターに設定して準備しておくことです。従来の BI ツールと比較して、Tableau Server はすばやくインストール、構成、実行できます。また、多くの場合、スタンバイマシンにクリーンサーバーをインストールし、バックアップから復元することでビジネスの目的を達成できます。サーバーのインストールとバックアップの復元にかかる時間はバックアップのサイズとサーバーの複雑性に依りますが、ほんの数時間で簡単に完了できます。

## サードパーティの DR ソリューション

15 分またはそれ以上の RPO を提供するサードパーティのソリューションはたくさんあり、お客様の IT チームですらそれを利用しているかもしれません。サードパーティのバックアップ復旧と DR ソリューションはこのホワイトペーパーの範囲外ですが、サードパーティの復旧システムを使用する場合には、Tableau Server をシステム上にある他のミッションクリティカルなアプリケーションと同じように扱う必要があります。サードパーティのバックアップ復旧テクノロジーもそれに合わせて構成する必要があります。

## Tableau の DR 機能

サードパーティの DR テクノロジーを使用できない場合、Tableau Server にはネイティブのバックアップおよび復元機能が組み込まれています。Tableau Server では、実行中でもバックアップを作成できるので、完全なホットバックアップを作成できます。加えて、このホットバックアップは Tableau Server と同様にスマートで、抽出ファイルとリポジトリデータベース、および関連する構成のみをバックアップします。計画停止など、Tableau Server を実行していないときでもバックアップを実行できます。

24 時間の RPO の場合は、24 時間ごとのサーバーのバックアップを選択できます。また、RPO に応じてバックアップスケジュールを変更できます。復旧時間は、ソフトウェアの障害が発生したかハードウェアの障害が発生したかによって、また新しいハードウェアをサイトにプロビジョニングできるかどうかによって異なります。すでに Tableau Server の同じインスタンスをインストールして構成している他のノードがスタンバイになっている場合、Tableau Server を本番環境に復元する作業は、保存したバックアップファイルを使用して迅速にコマンドラインの管理操作をするだけです。

ほとんどの企業にとって、ディザスタリカバリポリシーには、主にサーバーが稼働しているデータセンターと、復旧のターゲットとなるデータセンター間の距離の要件が伴います (たとえば、アメリカ合衆国では、DR サイトは少なくとも 250 マイル以上離れている必要があります)。このような状況で、Tableau はディザスタリカバリのために管理される他のビジネスクリティカルなアプリケーションと同じように操作ことができます。Tableau の組み込みのバックアップおよび復旧テクノロジーを活用して DR のニーズに対応してください。

## 高可用性のその先に

高可用性を実現するためには、規模に応じたセルフサービス分析を実行するために Tableau Server に組み込まれている機能だけでは十分ではありません。ハードウェアやネットワークなどの障害は Tableau Server ソフトウェアの範囲外で発生する可能性もあります。パブリッククラウドまたはプライベートクラウドの仮想ハードウェアなど、この種類の障害に対してより復旧容易性の高いインフラストラクチャ上に Tableau Server を導入することで、高可用性を向上できます。VMware vMotion など、ライブマイグレーション機能を提供しているベンダーもあります。

実際の高可用性は、ユーザーのニーズを理解し、そのニーズを満たすためのベストプラクティスとプロセスに従うことで実現できます。Tableau の堅牢な機能を使うことで高可用性を達成することは簡単になりますが、定期的な保守や計画的バックアップなどのベストプラクティスに代わるものではありません。ビジネスで意思決定をするためには、ビジネスインテリジェンスアプリケーションが重要です。組織における最も重大な分析上の問題を解決するために、Tableau Server をご使用ください。

# Tableau について

Tableauは、インパクトを生み出すアクションにつながるインサイトを、お客様がデータから引き出せるように支援しています。どこにあるどのような形式のデータにでも、簡単にアクセスできます。隠れたビジネスチャンスを見つけ出すアドホック分析もすぐに行えます。ドラッグ & ドロップ操作で、高度なビジュアル分析を行えるインタラクティブなダッシュボードを作成できます。そして組織全体で共有すれば、チームメンバーが自分の視点からデータを分析できるようになります。グローバルな大企業から、中小企業やスタートアップまで、あらゆる場所で多くのお客様が Tableau の分析プラットフォームを使い、データを見て理解しています。

## リソース

[企業向け Tableau: IT の概要](#)

[Tableau Server のスケーラビリティ: サーバー管理者向けの技術的導入ガイド](#)

[サーバー管理者ガイド](#)

[Amazon Web Services における Tableau](#)

