



# High Availability for Tableau Server

Delivering Mission-Critical Analytics at Scale

By: Fatima Calcuttawala, Product Manager  
Kitty Chou, Product Manager

# Table of Contents

<b>Self-Service Analytics is Mission-Critical</b> .....	3
<b>Understanding High Availability</b> .....	3
<b>Minimum High Availability Deployment</b> .....	4
<b>External Load Balancer (ELB)</b> .....	5
<b>Understanding Tableau Server High Availability for Each Server Process</b> .....	6
General .....	6
TSM Services .....	6
Business Services .....	10
<b>Monitoring Cluster State</b> .....	16
Integrating with Third-party Monitoring Tools .....	18
<b>Recover from Initial Node Failure</b> .....	19
<b>Architectural Considerations</b> .....	19
Basic 3-Node HA Deployment .....	20
Deployments with More than Three Nodes .....	21
External Coordination Service Ensemble .....	22
<b>Beyond High Availability</b> .....	23
<b>About Tableau</b> .....	24
<b>Additional Resources</b> .....	24

# Self-Service Analytics is Mission-Critical

Today, self-service analytics and data-driven decision-making are the norm in leading organizations worldwide. Users and decision makers have come to depend on immediate access to data and self-service tools to answer their questions in real time. Executives understand the importance of data-driven decisions at their companies and rely on these systems daily. This reliance on data requires a high degree of availability to the underlying systems. A platform's capabilities need to be more accessible and easily configurable by existing teams and enterprise tools.

Tableau Server delivers the future of mission-critical self-service analytics. It enables rapid self-service data exploration, promotes trust in content and data through robust governance, and is easy to deploy, manage, and scale across any enterprise. In this paper we will explore how Tableau Server running Tableau Services Manager (TSM) delivers self-service analytics at scale with high availability (HA).

## Understanding High Availability

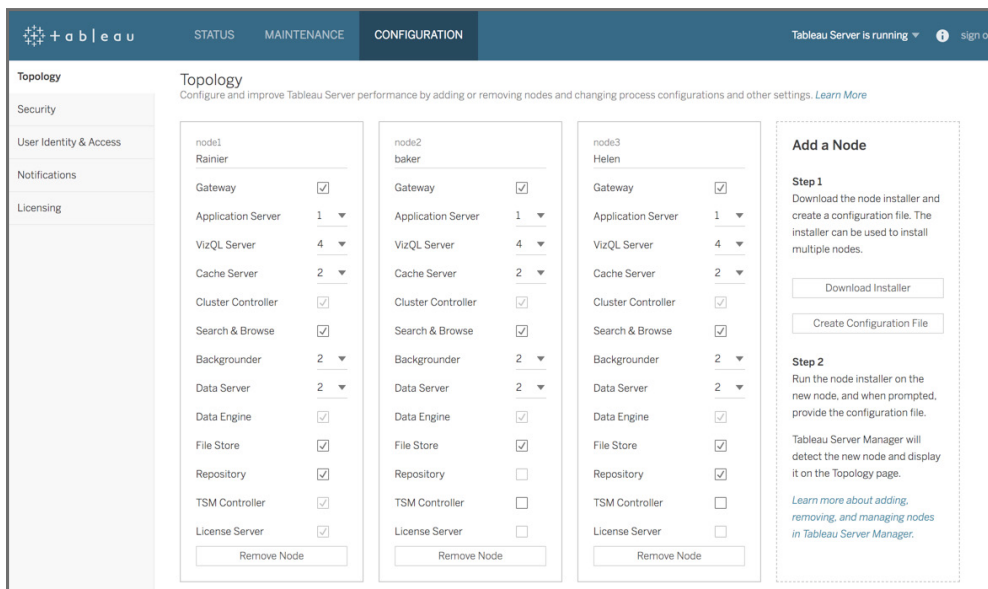
The goal of highly available systems is to minimize downtime of the system. Most system administrators plan downtime for maintenance, upgrades, and patching. In addition, there is some likelihood of unexpected failures, which is referred to as unplanned downtime. Of course, administrators need to conduct planned maintenance for hardware or software updates; the goal is to minimize unplanned downtime.

There are two common strategies for achieving HA. The first is eliminating single points of failure, so that the system can be robust to unexpected failures. We know that failures happen in real life and the best way to protect against these failures is to ensure redundancy in the system. The second is detecting when there are failures and triggering reliable failover mechanisms as necessary. Tableau Server employs both techniques to achieve HA.

We understand how important it is for users to readily see and understand their data. We also realize there will always be events that threaten the availability of business intelligence systems, whether related to hardware, software, networks, or even human error. That is why we have built Tableau Server with high availability out of the box and made it easy to configure and setup. At a minimum, Tableau Server processes will automatically restart to keep your system running in the event of component failure. A properly configured multi-node deployment also uses redundant processes to achieve server high availability.

# Minimum High Availability Deployment

The first step to becoming highly available is to create a distributed installation of Tableau Server, since you will need some redundancy in your cluster. If one of the nodes goes down, you want to make sure that you still have at least one of each of the core services for running Tableau Server configured on one of the other nodes in the cluster. However, two nodes are not sufficient for HA in a Tableau Server installation. In order to be HA, you must install Tableau Server on at least 3 nodes. The primary reason for this is that Tableau Server relies on the concept of a quorum to determine if the cluster is in a consistent state to protect against network partitioning problems. (See [CAP theorem](#) to learn more.) A quorum is just another way of saying an absolute majority. If a quorum (or majority) of the nodes in the cluster agree on the state of the system, then we can conclude that the system is consistent and therefore not susceptible to the network partition problem. With a two-node cluster, if one-node goes down, then there is no quorum that can be established, and Tableau Server must go into a failure mode. A cluster with three or four nodes can tolerate the loss of, at most, one node. A cluster with five or more nodes can tolerate losing up to two nodes. (See [Coordination Service](#) for more details.) You can easily add additional nodes and configure redundant services on the additional nodes using the TSM Web UI or command line interface (CLI).

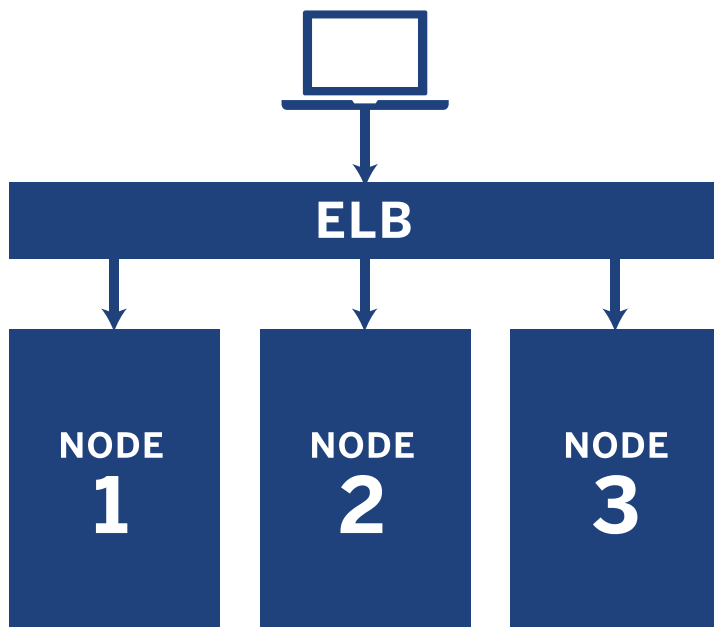


**Figure 1** The Tableau Services Manager Web UI allows you to easily add/remove nodes and processes per node in a cluster.

With Tableau Server running TSM, configuration has changed from a primary/worker server concept to all nodes as peers, i.e. all the nodes in the cluster are treated as peers. With TSM, you do not need a dedicated backup primary machines to failover to in case of failure in licensing service. See [Licensing Service](#) to understand more how we can achieve HA.

## External Load Balancer (ELB)

When deploying a highly available system we also need to consider how end users connect to the Tableau Server cluster. In most cases, users will connect via a DNS hostname that you provide to Tableau Server—this can be the hostname of any node in the cluster with a Gateway process. However, if that Gateway process goes down, then users connecting to that host will be unable to access Tableau Server. To enhance the reliability of the Tableau Server cluster, we recommend configuring multiple Gateways across the cluster and adding an external load balancer in front of your Tableau Server cluster. End users will simply connect to the DNS for the external load balancer, and the load balancer can distribute requests across the available Gateways in the cluster. If one Gateway in a cluster becomes unavailable, the load balancer can detect this failure and stop sending requests to it. See [Add a Load Balancer](#) for more details on how to set this up properly for Tableau Server.



**Figure 2** The Tableau cluster with three nodes behind an ELB.

# Understanding Tableau Server High Availability for Each Server Process

Tableau Server has several processes that ensure the entire system is accessible by end users. This section assumes familiarity with Tableau Server components and what they do. If you are not familiar with them, we suggest first reading about the [Tableau Server Processes](#).

Understanding how to make Tableau Server highly available largely includes understanding how to make each of its components highly available. In other words, high availability for every component must be ensured for the entire Tableau Server cluster to be highly available, providing redundancy for single points of failure. Let us consider each component in turn.

## General

Tableau Server has built-in automation for restarting all its server processes. This automation ensures high availability by automatically restarting server processes that fail. The hardware or virtual machine that houses Tableau Server must be healthy for this automation to occur.

To protect against failure of an entire node, it is important to configure the unique server processes so that they are redundant across different nodes in the cluster. This redundancy is possible for all processes except for the Licensing service and Administration Controller service, which can only have a single instance running across the entire cluster. We discuss how to address these two process limitations in more detail within the Failover of Initial Node section.

## TSM Services

With Tableau 10.5 Linux and Tableau 2018.2 Windows releases, we introduced a new set of services with Tableau Services Manager (TSM). The purpose of these services is to manage the Tableau Server deployment. Five TSM services are always running, even when Tableau Server is stopped. Three other TSM maintenance services are stopped except when needed. If TSM services stop, they are automatically restarted if the host computer is healthy.

### Coordination Service

The Coordination Service is built on Apache Zookeeper, an open-source project, and coordinates activities on the server, guaranteeing a quorum in the event of a failure, and serving as the source of “truth” regarding the server topology, configuration, and state. The service is installed automatically on the initial Tableau Server node, but no additional instances are installed as you add additional nodes. Because the successful functioning of Tableau Server depends on a properly functioning Coordination Service, we recommend that for server installations of three or more nodes, you add additional

instances of the Coordination Service by deploying a new Coordination Service ensemble. This provides redundancy and improved availability if one instance of the Coordination Service has problems.

The number of recommended nodes to install Coordination Service on is dependent on the number of nodes in the cluster as outlined in the following table:

# of Nodes in the Cluster	# of Nodes with Coordination Service
<b>1-2</b>	<b>1</b>
<b>3-4</b>	<b>3</b>
<b>5+</b>	<b>5</b>

You may select which nodes to deploy the Coordination Service ensemble on, and you may deploy either a 1, 3, or 5-node Coordination Service ensemble.

A complete Tableau Server outage will occur if the number of running Coordination Service processes does not constitute a quorum, which is based on the total number of configured Coordination Services processes. Notably, a cluster with only two nodes cannot tolerate the loss of even a single Coordination Service process, and this is the reason why full HA, including automated failover, requires a minimum of three nodes.

What happens when a Coordination Service process fails? Nothing, as long as the number of remaining Coordination Service processes still constitutes a quorum. If the number of still functioning Coordination Service processes number less than a quorum, the entire Tableau Server cluster becomes unavailable in order to protect the referential integrity of the underlying Postgres database and service state. Failed Coordination Service processes automatically restart if the computer itself is okay.

## Agent

The Administration Agent monitors the Coordination Service for changes to configuration or topology and delivers new configurations to each service (configuration) or deploys new services and removes old ones (topology). The Agent also checks each of the services for status and reports this back to the Coordination Service. This process will be automatically configured for you for each node of the cluster during installation—no explicit configuration is required.

What happens when an Agent process fails? All other Tableau Server processes running on the same node will display as “unavailable” on the TSM status page. Tableau Server will continue to work as expected, however you will not be able to make configuration/topology changes to the cluster. Failed Agent processes automatically restart as long as the computer itself is okay. If the Agent doesn’t start up on the node, you can try to start the services manually by running the following commands:

- Windows: `sc start tabadminagent_0`
- Linux: `systemctl start tabadminagent_0`

## Service Manager

The Service Manager manages the lifecycle of the Tableau Server business processes described in the below section. Like Agent, this process will be automatically configured for you on each node of the cluster during installation, no explicit configuration is required

What happens when a Service Manager process fails? All the business services on that node will fail as well. Depending on the services running on the node Tableau Server may run in a degraded state. The Service Manager will automatically restart itself if the machine itself is healthy. If the Service Manager doesn’t start up on the node you can try to start the services manually by running the following commands:

- Windows: `sc start tabsvc_0`
- Linux: `systemctl start tabsvc_0`

## Client File Service (CFS)

The Client File Service stores and distributes files needed by TSM (i.e. SSL certs, customization files, etc.). It functions much like the File Store does for files needed by business services (i.e. extracts, thumbnails, etc.). By default, CFS is only installed on the initial node of your Tableau Server installation. In order to configure CFS to be HA, we recommend you configure an instance of CFS on each of the nodes where you deploy the Coordination Service.

What happens when a CFS process fails? Nothing, as long as there is still at least functioning CFS process in the cluster. The controller will redirect file transfer requests to the other working CFS process.

Failed Client File Service processes automatically restart as long as the computer itself is okay.



## TSM Maintenance Services

There are three TSM Maintenance Services that are installed on every node of the cluster: Database Maintenance, Backup/Restore, and Site Import/Export. These services remain stopped unless a specific maintenance task that requires them is initiated by the Server Admin. No additional HA configuration is required for these services. These services are used only for maintenance tasks such as backup and restore and should not impact the functioning of Tableau Server for the end users.

## Controller

The Administration Controller process hosts the TSM REST API for configuring and managing your Tableau Server deployment. There can only be a single instance of the Administration Controller in the entire cluster. If the Administration Controller fails, the Tableau Server cluster should continue to function; however, you will not be able to make any changes or updates to the configuration/topology until Controller is back up and running. Like other TSM services, Controller is automatically restarted if it is stopped/failed.

If this does not resolve the situation, you should move the Controller process to another node in the cluster to mitigate the situation. See more in the Recover from Initial Node Failure section.

## Licensing Service

The Licensing Service is responsible for ensuring that the overall server is properly licensed. Similar to the Controller, there can only be one instance of Licensing Service running in the entire cluster. Licensing service needs to be co-located with controller. If the Licensing Service fails, Tableau Server will continue to run for up to a maximum of 72 hours until the next scheduled licensing check or until a licensed service is restarted. In case of Licensing Service failure, you should move Licensing Service to another node and follow steps to reactivate your license key to avoid going into an unlicensed state. See more in the Recover from Initial Node Failure section.

## Business Services

### Gateway

The Gateway process can be run on any and all nodes of the Tableau Server. The Gateway is a web server that handles all requests to Tableau Server from browsers, Tableau Desktop, and other clients. Each node can only have a single Gateway process, so we recommend configuring an instance of the Gateway on multiple nodes in the cluster for redundancy in the system. In fact, we recommend that you configure a Gateway process on each node. This mitigates the risk of the Gateway process being a single point of failure leading to service unavailability. As long as there is at least one Gateway process that is running and taking traffic, the cluster can still continue to service user requests.

The Gateway also serves an important function in providing HA for Tableau Server as it is the component that is responsible for load balancing or redirecting traffic to specific business services with Tableau Server. The Gateway load balances traffic across the set of active services of the same type. It will detect when certain processes are unavailable and stop sending user traffic to those unavailable processes for some time. The Gateway will periodically check for the status of those processes and add them back in rotation once they have recovered. In this way, traffic is automatically load balanced and routed only to healthy services in the cluster without any manual intervention.

So, what happens when a Gateway process fails? As mentioned previously, if no Gateway processes are running the entire Tableau Server cluster will be unavailable. If other Gateway processes remain running, requests made to those working Gateways will be processed normally. However, any requests received by the failed Gateway will not be redirected and will continue to fail, despite the presence of other functioning Gateways. Failed Gateway processes automatically restart; so as long as the computer itself is functioning, the failed Gateway process will relaunch and resume serving requests.

To make your installation even more robust to Gateway failures, we recommend, placing your Tableau Server cluster behind an external load balancer to ensure that requests only get routed to functioning Gateway processes. The article [How to add a Load Balancer to Tableau Server](#) provides some additional guidance on setting this up.

## Repository & Cluster Controller

The Repository is a PostgreSQL database that is the central repository of metadata for Tableau Server that includes user information, permissions, workbooks, data sources, schedules, etc. If there is no fully-functioning Repository, the entire Tableau Server cluster is unavailable. At any point in time, there should be one “active” Repository, which should be fully-functioning. All operations involving the Repository use this active Repository.

To improve availability, you can configure Tableau Server with an additional, “passive” Repository on a different node of the cluster. The contents of the active Repository constantly stream to the passive Repository. In the event of a failure of the active Repository, a cluster configured for high availability will automatically promote the passive Repository to active status for continued availability of the server. Customers who want high availability will certainly want to configure a passive Repository. In an entire cluster, there can be a maximum of only two Repositories—with one active and one passive—they cannot be on the same node and must therefore be on separate nodes.

Cluster Controller is a required server component for any node in the cluster. This process will be automatically configured for you for each node of the cluster during installation; no explicit configuration is required. The primary function of Cluster Controller is to monitor and manage Repository startup, shutdown, and any failover from active to passive. If a Cluster Controller process fails, any Repository process running on that node will also be unavailable. Like other server processes, a failed Cluster Controller process will automatically restart as long as the computer itself is okay. When the Cluster Controller restarts, it also restarts any Repository process configured on that node. This is why a failure in the Cluster Controller process can cause a problem with the Repository.

What happens when a Repository process fails? That depends.

The following list summarizes the various cases:

- If the passive Repository fails, then users should experience no impact. Everything will continue to work since the active Repository is still functioning. In the background, the passive Repository will be restarted, and data replication will resume, though there may be some delay before the passive Repository is again fully synchronized with the active Repository.
- If the active Repository fails, and there is a fully synchronized passive Repository available, and the cluster is configured for HA, then a failover to the passive Repository will be automatically triggered after five minutes. After the failover, the previously passive Repository will be the new active Repository. The system will restart the failed previously active Repository as the new passive Repository and begin synchronization. It will also restart other relevant processes automatically, so they are aware of the newly-promoted active Repository and can reconnect. During this short window of restarts, users will experience a service interruption. However, administrators do not have to intervene to ensure continued availability as this is an automated sequence. Should you wish to manually promote the passive Repository to active, you can do so using the following command:

```
tsm topology failover-repository
```

- If the active Repository fails, and if there is no fully synchronized passive Repository, then Tableau Server is unavailable until the active Repository can be restarted. The system will attempt to do this automatically, but depending on the reason for the failure, this may not be possible. This means if only the active Repository is currently usable, then Tableau Server cannot be considered highly available. The active Repository is a single point of failure for the entire system if there is no synchronized passive Repository.

*Please Note: Synchronizing between two Repositories is done and controlled by PostgreSQL.  
Tableau Server has no control over it.*

## File Store

The File Store process manages storage and replication of extracts and other files between nodes. To make the File Store highly available, simply configure File Store process on multiple nodes of the cluster.

## How File Store Works

An extract file is created in the system when a user first publishes it to Tableau Server or an extract refresh of it occurs. Immediately after one of these events, the extract exists on a single File Store. The specific extract file cannot be said to be highly available yet as it does not have redundancy and is therefore a single point of failure. The File Store processes communicate with each other to quickly replicate local extracts to all other File Store nodes in the cluster. The File Store process is designed to copy the files as quickly as the cluster network resources allow, but can take a variable amount of time depending on the size of the extract and other resource demands on the system. Once a copy is available on multiple nodes within the clusters, the extract file is replicated and has redundancy.

What happens if a File Store process fails? There are two consequences:

- Copying of extract files to and from the affected node stops.
- Removing extract files that are no longer needed on the affected node is suspended. (This removal process is usually referred to as “extract reaping.”)

The suspension of extract reaping has no immediate effect. It simply means that a buildup of unwanted extract files will begin to consume disk space on that node. Eventually this can cause a problem, but correctly sized nodes should have a comfortable buffer of disk space.

The lack of file replication means that new extract files added to the working File Store node will not be available on failed File Store nodes in the cluster. After the File Store process restarts, the system corrects itself by ensuring synchronization between all File Stores on all nodes.

Failed File Store processes automatically restart as long as the computer itself is okay. The File Store process should quickly resume its function, including synchronization of all files, whether they were added during the period of failure or after.

## Application Server

The Application Server (VizPortal) handles the web application, REST API calls, and supports browsing and searching. Achieving high availability with Application Server is easy. You just need to configure Application Server instances on each node in the Tableau Server cluster.

What happens when an Application Server process fails? Requests being handled by that instance will fail, but subsequent requests will be routed to other running Application Server processes. Assuming the node containing the failed Application Server is still running, the failed process should automatically restart within seconds.

## SAML Service

For Tableau Server installations that have [Site-Specific SAML](#) enabled, there will also be a SAML Service instance running on each node that is configured with Application Server. This will be automatically configured when site-specific SAML has been enabled on the server. SAML Service on Tableau Server will show as stopped unless site SAML is enabled. If this process goes into a failed state, then users will not be able to login to Tableau Server if the user request is routed to an Application Server on this node. Like other processes, when a node containing the SAML Service fails, the failed process should automatically restart within seconds.

## Backgrounder

The Backgrounder runs server tasks, including extract refreshes, subscriptions, 'Run Now' tasks, and tasks initiated from tabcmd. To make the Backgrounder service highly available, you should configure one or more instances to run on multiple nodes in the cluster. When deciding where and how many Backgrounders to run, consider how other server processes will affect each machine's available capacity .

What happens if a Backgrounder process goes down? Refresh and subscription jobs on the failed Backgrounder process are retried once the Backgrounder process recovers from failure. Most background jobs are scheduled to run periodically, and the same background task will be picked up and performed normally at the next scheduled time by a functioning Backgrounder process.

Failed Backgrounder processes automatically restart as long as the computer itself is okay, and the failed jobs will be retried.

## Data Server

The Data Server manages connections to Tableau Server data sources. To make Data Server highly available, configure one or more Data Server processes to run on multiple nodes of the cluster.

What happens if a Data Server process fails? Queries running through the Data Server process will fail, resulting in a failed view rendering, extract refresh, or alert. Subsequent requests, including a retry of the failed operation, should succeed as long as a working Data Server is available to accept rerouted requests.

Tableau Server is not dependent on Data Server to function; however, without a running Data Server, workbooks on the server lose the ability to connect to or query published data sources. Any view that does not use Data Server for any of its data sources should still function correctly.

## VizQL Server

The VizQL Server loads and renders views, and computes and executes queries. To achieve high availability for the VizQL Server process, simply configure one or more instances to run on multiple nodes.

What happens if a VizQL Server process fails? If there is only one VizQL Server process and it fails, then Tableau Server will no longer be able to render any views. High availability requires configuring redundant VizQL processes. A fairly typical configuration consists of two to four VizQL Server processes on each node. This simultaneously serves the need for high availability and scalability. If multiple VizQL Server processes are running, then the failure of a single process will result in the failure of any requests and the loss of session data at the time of its failure. Any future requests will be routed to the other working VizQL Server processes on the Tableau Server cluster.

## Data Engine

The Data Engine processes loads and queries data extracts when using in-memory analytics. A Data Engine process will be automatically configured on any node that runs one or more File Store, VizQL Server, Backgrounder, Data Server, or Application Server processes.

What happens when a Data Engine process goes down? Ongoing extract refreshes will fail. New refreshes of .tde extracts will not be impacted; however, .hyper extract refreshes on that node will continue to fail until Data Engine is recovered. If viz rendering is running any query on the failed node, rendering will fail. Running the same operation again will cause it to automatically be reassigned to a different Data Engine that is functioning. If the viz includes a shadow extract and the request is routed to a node with a failed Data Engine, then the view rendering will continue to fail until the Data Engine on that node is recovered.

Failed Data Engine processes are automatically restarted; as long as the computer itself is okay, the Data Engine process will relaunch.

## Cache Server

The Cache Server provides a shared external query cache. It's a cache of key/value pairs which hold information from previous queries to speed up future requests. To make Cache Server highly available, configure one or more Cache Server processes on multiple nodes of the cluster.

What happens when Cache Server process goes down? The consequences are relatively mild. Tableau Server will still work but actions may take longer as they do not have pre-cached results available. As the queries rerun, the restarted Cache Server gets repopulated, eventually speeding things up for end users. In effect, Cache Server does not have an availability impact; however, it does have an impact on various end user performance scenarios. To reduce impact on user performance, run multiple processes of this type across the cluster

A failed Cache Server process is automatically restarted; as long as the computer itself is okay, the Cache Server Process will relaunch.

## Search & Browse

The Search Service handles fast search, filter, retrieval, and display of content metadata on the server. Achieving high availability for the Search & Browse process is easy. The system should be configured so that the Search & Browse process is running on multiple nodes.

What happens if a Search & Browse process fails? Tableau Server is largely usable, and though users can still log into the system, workbook content will appear to be missing. The content is not actually missing, it is simply not being returned in the search results; it will be displayed again after the Search & Browse process restarts. If more than one Search & Browse process is configured and running on multiple nodes when the failure occurs, requests made to a failed Search & Browse process will also fail, but subsequent requests will be routed to working Search & Browse processes. Each Search & Browse process indexes across all nodes in the cluster, therefore if all but one Search & Browse process fails, results will still be returned across all nodes.

## Monitoring Cluster State

So far, we have seen how each server process behaves on failure and how you can mitigate risks of failure to ensure that the Tableau Server cluster as a whole is highly available. In addition to planning for each failure scenario, you should also proactively monitor the cluster for failures that have occurred in the past.



System administrators can monitor the state of Tableau Server on the TSM status page. This provides administrators with an understanding of the health of the overall cluster by displaying the server processes across all nodes.

Process	node1	node2	node3
Gateway	✓	✓	✓
Application Server	✓	✓	✓
VizQL Server	✓ ✓ ✓ ✓	✓ ✓ ✓ ✓	✓ ✓ ✓ ✓ ✓
Cache Server	✓ ✓	✓ ✓	✓ ✓
Cluster Controller	✓	✓	✓
Search & Browse	✓	✓	✓
Backgrounder	✓ ✓	✓ ✓	✓ ✓
Data Server	✓ ✓	✓ ✓	✓ ✓
Data Engine	✓	✓	✓
File Store	✓	✓	✓
Repository	✓		✓
TSM Controller	✓		
License Server	✓		

Refresh Status      ✓ Active    ⌂ Busy    ✖ Error    ✕ Stopped

Figure 3 The TSM Status page shows the status of each process on each node.

Tableau Server can be configured to send email alerts to server administrators when there is a system failure. This can also be used to provide early warning of disk space capacity issues.

**Notifications**  
Configure Tableau Server to send email notifications about critical events, processes and server health. Email notifications must be sent through an email (SMTP) server. [Learn more](#)

**Email Server**      **Events**

**Events**  
You can specify which server events will trigger an email notification. We recommend enabling all notifications. [Learn more](#).

**Content updates**

- Send email when flow runs, encryption jobs, or scheduled refreshes fail
- Allow users to receive email for views that they have subscribed to

**Server health monitoring**

- Send emails for Tableau Server process events (up, down, and fallover)
- Send emails for Tableau Server license reporting

**Drive space**

- Send emails when unused drive space drops below thresholds
- Warning threshold:  %
- Critical threshold:  %
- Send threshold alert every:  minutes
- Record disk space usage information and threshold violations for use in custom administrative views

Cancel    Save Pending Changes

Figure 4 The TSM Web UI provides the ability to set alerts on system failures.

## Integrating with Third-party Monitoring Tools

In addition to monitoring the system's health using Tableau Server's built-in mechanisms, you can also remotely receive a machine-readable JSON formatted version of each process's status. You will need to access the TSM REST API for each status. Here's an example:

```
GET /status
Response status code
200 OK
Response body example
{
  "clusterStatus": {
    "nodes": [
      {
        "services": [
          {
            "serviceName": "filestore",
            "instances": [
              {
                "code": "ACTIVE",
                "processStatus": "Active",
                "instanceId": "0",
                "timestampUtc": 1497060680268,
                "currentDeploymentState": "Enabled",
                "binaryVersion": "<build>"
              }
            ],
            "rollupStatus": "Running",
            "rollupRequestedDeploymentState": "Enabled"
          }, {
            "..."
          }
        ],
        "nodeId": "node1",
        "rollupStatus": "Running",
        "rollupRequestedDeploymentState": "Enabled"
      }
    ],
    "href": "/api/0.5/status",
    "rollupStatus": "Running",
    "rollupRequestedDeploymentState": "Enabled"
  }
}
```

## Recover from Initial Node Failure

When you install Tableau Server, the License Service and Controller processes will be installed by default on the initial node in the cluster. Since you can only have one instance of each of these two processes across the entire cluster, they will not be available on any other nodes in the cluster and thus become single points of failure in the system. When these processes fail, the server could become unlicensed and disabled. In addition, you will be unable to administer and make changes to the server cluster until you recover. Therefore, it is critical to use the monitoring techniques above to detect and react to such failures.

When you detect a failure on the initial node (or a node running Licensing and Controller), Tableau Server will continue to function until next license check or until licensed services restart. You should try to identify what caused the issue and fix any node or service with issues. If you are unable to bring the node or services to healthy state, then we recommend that you follow the steps documented in the [Recover from Initial Node Failure](#) topic to move these two services to one of the other nodes in your cluster. In addition, these steps can be automated and scripted for easy of repeatability and to avoid manual errors during the execution of the steps.

## Architectural Considerations

For a successful implementation of high availability, you should understand the uptime objectives and expected service levels of your organization, then plan for redundancy accordingly. Determining your optimal cluster configuration will depend on both the needs of your business and the resources available in your organization. Although there are several possible successful configurations for high availability, it's important to understand the tradeoffs between clusters of various sizes so you can make the best choice for your environment.

Additionally, you may choose between symmetrical (each node has all of the same components) or asymmetrical computer configurations (each node has a varying number and type of components). Often times, having a symmetrical configuration for worker nodes simplifies cloning and appending additional nodes to the cluster. However, keep in mind the constraints of a single active and single passive Repository as you plan your configuration.

## Basic 3-Node HA Deployment

As we discussed earlier, you must have at least three nodes in the cluster to be running fully in HA mode. A 3-node cluster is also a great starting point for larger deployments. Should one node fail, you will still have a quorum with the remaining active two nodes. Figure 5 below shows a basic 3-node HA deployment. As discussed, the initial node (Node 1) hosts the administrative and licensing functions for the entire cluster, but other than that there is nothing special about the initial versus that the other nodes in the cluster. In this configuration, if Node 1 with the active Repository fails, Node 3 with the passive Repository automatically becomes active. All three nodes in the cluster are configured to power the data and visualization functionality of Tableau Server with redundancy both across the cluster and within a single node. Coordination service is deployed on all three nodes. If any one nodes fails, there would still be quorum (two out of three nodes still working). The exact mix of these components may vary depending on your scalability needs in addition to your HA needs. For more on Tableau Server scalability, please also see our [whitepaper](#).

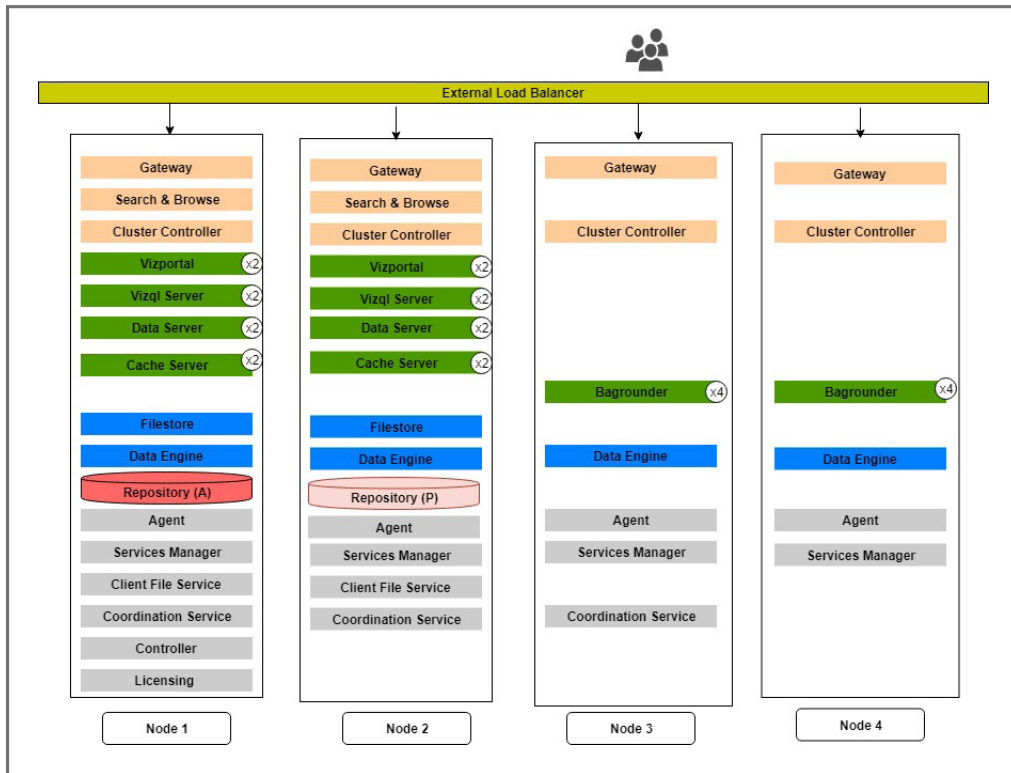


**Figure 5** Example of a basic 3-node HA deployment.

In addition, you should deploy the cluster behind the external load balancer for improved end-user availability. If you are just getting started with high availability, this is a good configuration to start with.

## Deployments with More than Three Nodes

When deploying a cluster with an even number for nodes (e.g. four or six nodes), the quorum ability of this cluster will be the same as a cluster with one less node. Coordination services can only be deployed on odd numbers of nodes. Let's consider an example of a 4-node cluster.



**Figure 6** Example of a basic 4-node HA deployment.

The initial node (Node 1) hosts the administrative and licensing functions for the entire cluster. In this configuration, if Node 1 with the active Repository fails, Node 3 with the passive Repository automatically becomes active. Node 1 and Node 2 in the cluster are configured to power the data visualization functionality of Tableau Server with redundancy both across the cluster, and both nodes have multiple instances for VizPortal, VizQL Server, Data Server, and Cache Server. Node 3 and Node 4 are configured as dedicated Backgrounder node. This will ensure that the extract refresh workload doesn't interfere with data visualization. Coordination Service is deployed on all the three nodes. If any one of the nodes with Coordination Service fails there would still be quorum (two out of three nodes still working). If Node 4 and one other node from the cluster fails, cluster will continue to work since Coordination Service will have quorum. If two nodes from Node 1, Node 2, or Node 3 fail, the cluster will no longer have quorum since two of the three Coordination Services have failed. In such a situation you need a plan for recovering your system from this disaster.

Setting aside other considerations and just focusing on the assumption that you want the least downtime, you should consider a deployment architecture that is five or more nodes. While we recommend the 3-node architecture in this paper as a great starting point, large enterprise-wide deployments with mission-critical high availability requirements should consider using five or more nodes as a baseline.

For organizations that rely heavily on extracts and extract refreshes, a commonly used deployment architecture includes deploying Backgrounder processes onto dedicated worker nodes. Extract refresh workloads can sometimes interfere with the visualization workloads supported by VizQL Server processes, so by partitioning Backgrounder processes to dedicated worker machines, you can avoid resource contention between these two different workloads. To ensure that you have process redundancy across multiple nodes in the cluster, Backgrounder-only nodes are often deployed in pairs.

### External Coordination Service Ensemble

The Coordination Service can generate a large amount of I/O as it communicates with other components of the server, so if you are running Tableau Server on computers that meet or just exceed the minimum hardware requirements or you just want to have a more robust Tableau Server installation, you may want to install Tableau Server in a configuration that uses Coordination Service-only nodes. This means installing Coordination Service on nodes that run no other server processes and removing Coordination Service from any nodes that are running other server processes. See [Configure Coordination Service-Only Nodes](#) for more information on setting up the Coordination Service Ensemble on dedicated hardware. The figure below shows a cluster setup with external Coordination Service Ensemble.

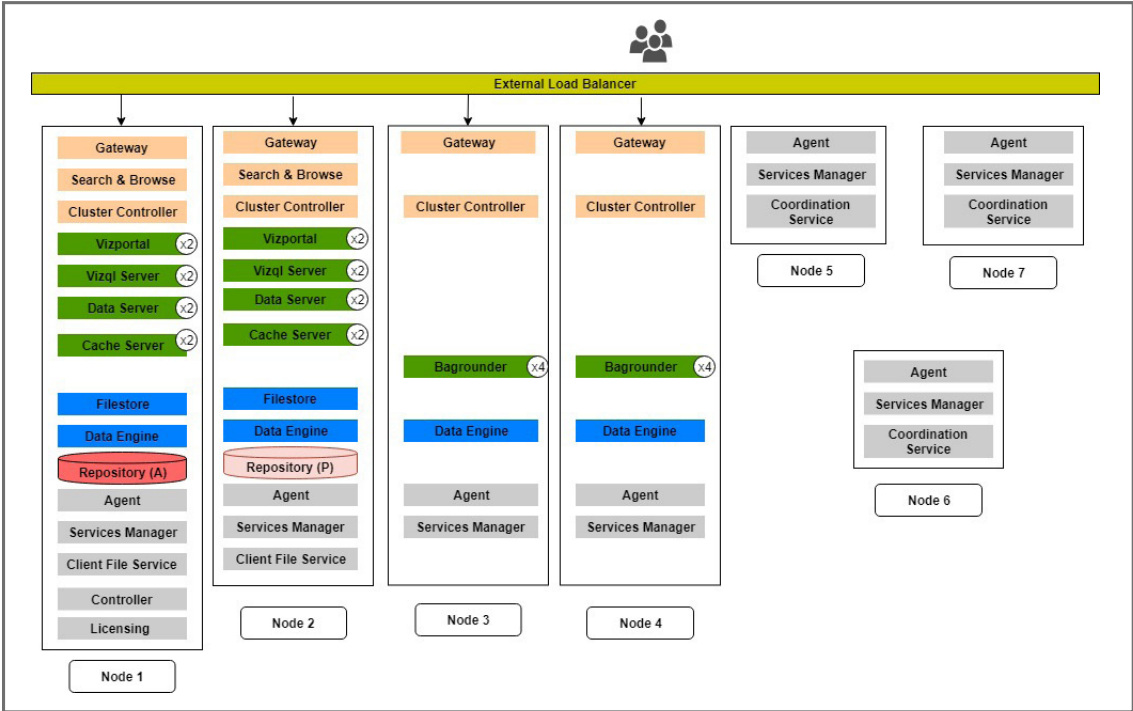


Figure 7 Example of an external coordination service HA deployment.

## Beyond High Availability

Achieving high availability requires more than just the features and functionality built into Tableau Server. Failures can occur outside the scope of the Tableau Server software, such as hardware or network failures. You can achieve improved high availability by deploying Tableau Server on infrastructure that is more resilient to these types of failures such as virtualized hardware in public or private clouds.

True high availability comes from understanding what users need and following best practices and processes to meet those needs. Although Tableau makes high availability easy to achieve with its robust features, these should not replace best practices such as regular maintenance and scheduled backups. Tableau realizes the importance of business intelligence applications to help your business make decisions. Tableau Server will be ready for your organization's analytical questions.

## About Tableau

Tableau is a complete, easy-to-use, enterprise-ready visual business intelligence platform that helps people see and understand data through rapid-fire, self-service analytics at scale. Whether on-premises or in the cloud, on Windows or Linux, Tableau leverages your existing technology investments and scales with you as your data environment shifts and grows. Unleash the power of your most valuable assets: your data and your people.

## Additional Resources

[Tableau Services Manager Overview](#)

[Distributed and High Availability Tableau Server Installations](#)

[High Availability and Disaster Recovery in Tableau Server](#)

[Tableau Server Scalability](#)

