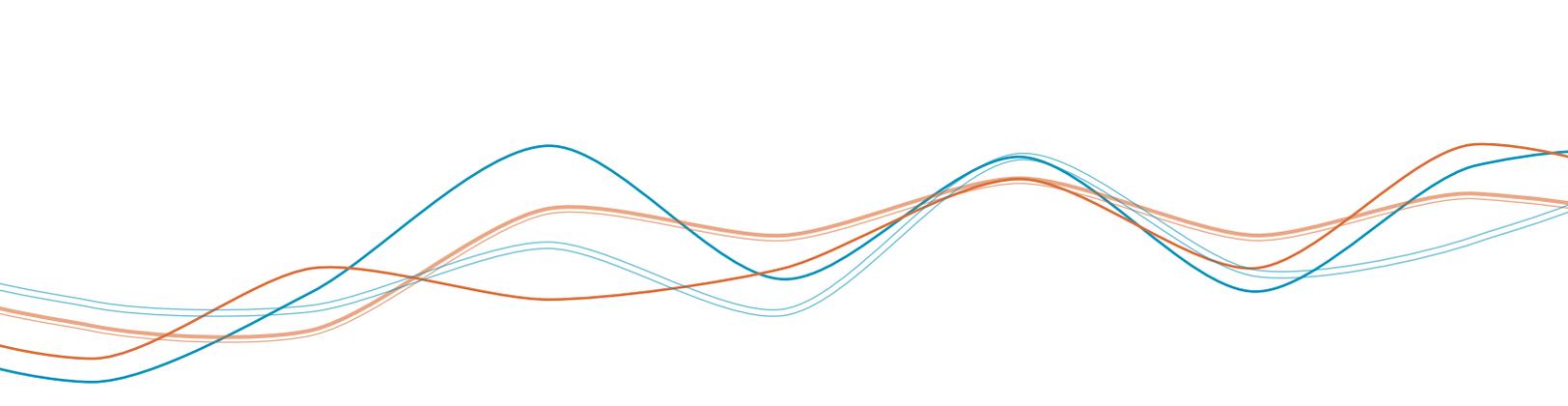


Hardware Planning and Server Configuration for Enterprise Tableau Deployments

September 2015



Introduction

Overview

About the Authors

Brad Fair
Mike Roberts
Eric Shiarla

Understanding Tableau's Product Offering

Tableau Desktop Professional
Tableau Server

Step 1: Planning a Tableau Environment

Identifying Users

Publishers - Server License + Desktop License
Report Viewers - Server License

Anticipating User Behavior & How It May Impact a Tableau Environment

Access-Related Questions
Data-Related Questions
Security-Related Questions
Performance-Related Questions

Step 2: Sizing Hardware & Configuring Tableau Server

Recommended Base Hardware

The Tableau Processes

VizQL
Background
Application Server and Data Server
Data Engine (Optional High Availability)
Repository (Optional High Availability)

Choosing an Extract, Live Connection or Hybrid Environment

Step 3: Load Testing

Why Load Testing?

Top Causes of Poor Performance

Workbook Complexity

Methodology

Load Testing Tools

HP LoadRunner
Apache JMeter
Selenium – Browser-Based Testing

Step 4: Scaling to Meet Demand

Determining When to Scale

How to Scale Up

How to Scale Out

Step 5: Maintaining & Monitoring the Production Environment

Using Tableau's Administrative Dashboards

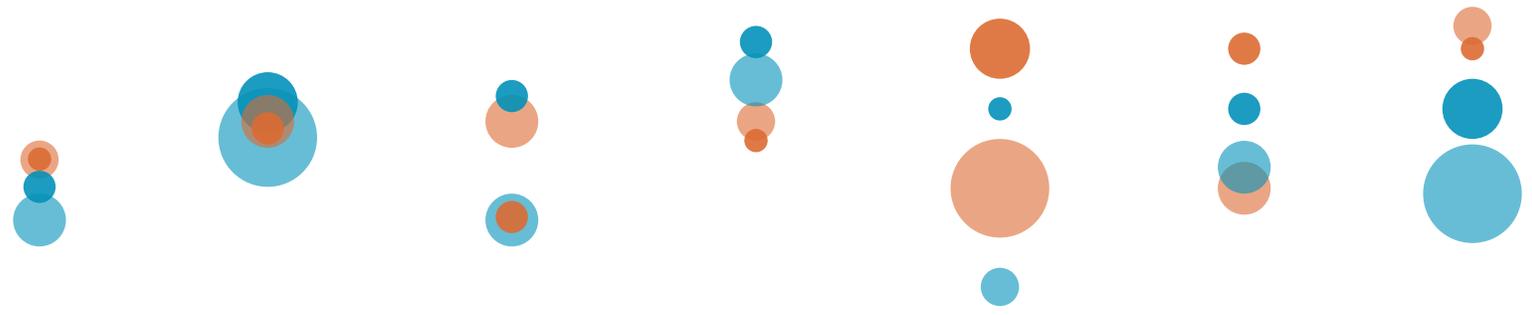
Tableau Server: Run and Maintain

Run

Maintain

Conclusion

About InterWorks



Introduction

Overview

While Tableau Server can scale to support the needs of the most mission-critical deployments, careful consideration needs to be adopted to meet scalability and performance requirements. This whitepaper will provide customers a road map that will facilitate the decision-making process and lay the groundwork for a successful deployment.

The steps essential for a well-executed deployment consist of:

- Step 1: Planning the Environment
- Step 2: Sizing Hardware & Server Configuration
- Step 3: Load Testing
- Step 4: Scaling to Meet Demand
- Step 5: Maintaining & Monitoring

While this whitepaper cannot detail all of the possible variations encountered during the planning and implementation of a Tableau deployment, additional resources and information will be highlighted after each section for more in-depth reading.

About the Authors



Brad Fair

Brad is a solution architect based out of Tulsa, Oklahoma. His background is in networking, IT services development, and virtualization/storage architecture. He has presented on numerous topics, including Tableau Server scalability and performance as well as high-performance analytical databases. Brad consults with Tableau's most performance-conscious clients to ensure their environments are running optimally.



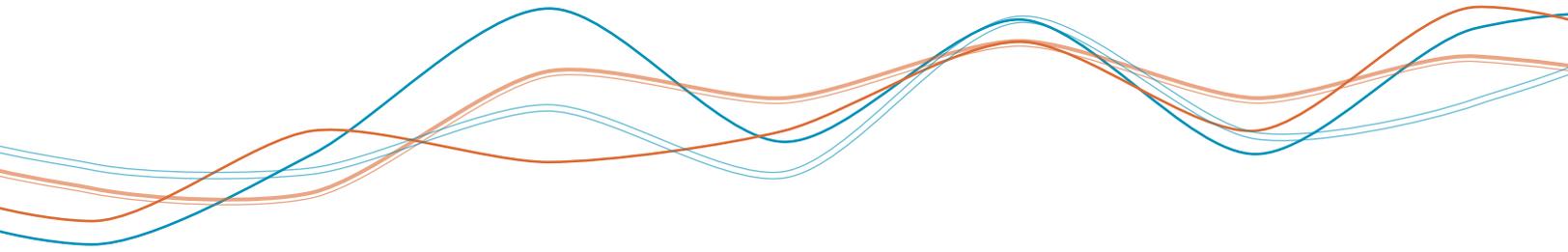
Mike Roberts

As Director of Data Analytics at Pluralsight, Mike Roberts has a wealth of experience and a varied background that encompasses databases, analytics, visualization and scripting. He has worked for Fortune 500 companies as well as small businesses, helping them understand their vast data troves. His analytical style revolves around people and context, and in an industry consumed with "defaults," he believes that data intelligence starts with giving people access to their information with collaboration in mind. Mike is a 2015 Tableau Zen Master as well as a contributing author to ["Tableau Your Data! - Fast and Easy Visual Analysis with Tableau Software"](#).



Eric Shiarla

Eric is a principal business intelligence consultant for InterWorks. Based out of Los Angeles, he manages a West Coast team of consultants that are actively involved in some of the largest Tableau deployments in the world. Eric has been a key presenter at numerous conferences and lectures on the subjects of data visualization and the impact of Tableau in the enterprise. Eric is also a contributing author to ["Tableau Your Data! - Fast and Easy Visual Analysis with Tableau Software"](#).



Understanding Tableau's Product Offering

Before delving into the specifics of a successful enterprise deployment, it's necessary to understand Tableau's primary product offerings: Tableau Desktop and Tableau Server. Tableau Desktop is used as the primary content creation tool while Tableau Server provides a platform to share, secure and manage your reports.

Tableau Desktop Professional

- Windows and Mac-based client
- Creates workbooks
- Connects to data, wherever it may live
- Builds rich, informative visualizations
- Combines visualizations into effective dashboards
- Saves or publishes visualizations and dashboards for users to access

Tableau Desktop Professional is licensed per desktop user.

Tableau Server

- Manage workbooks and data sources created in Tableau Desktop.
- Offers access via any browser or mobile device.
- Secures workbooks and data sources with group and user-level permissions.

Tableau Server is licensed per server user or through an enterprise license based on the number of CPU cores present in the server environment.

Step 1:

Planning a Tableau Environment

Planning a Tableau deployment starts with defining specific business requirements. There are a lot of considerations that can make this process seem very challenging, but it all starts with two essential questions:

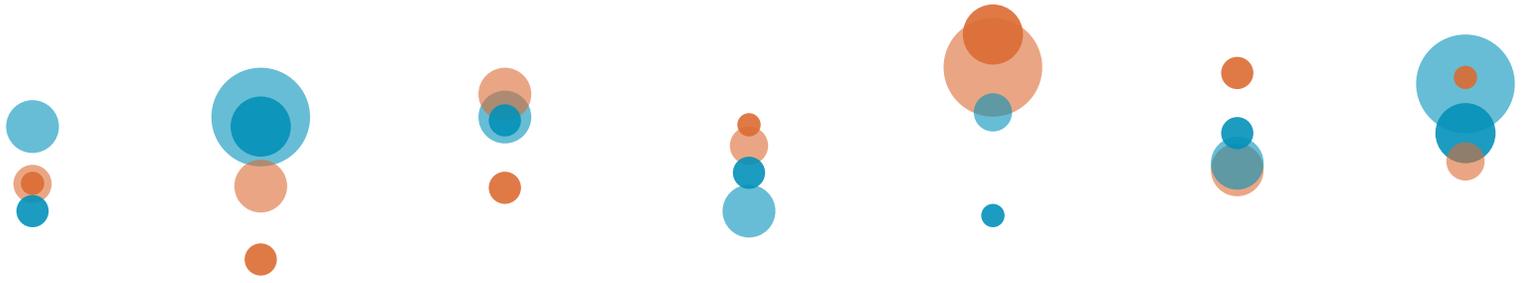
“How many concurrent users will the environment support?”

“What actions are these users likely to perform?”

Identifying Users

It is essential to identify approximately how many users require access, what type of user they are, where they come from (internal, external, etc.) and how they plan on using Tableau.

At the highest level, users can be classified under one of two groups: publishers or report viewers.



Publishers - Desktop License

Publishers are users that create content or maintain existing content published to Tableau Server. Publishers require a license for Tableau Desktop and a license for Tableau Server.

Report Viewers - Server License

Report viewers are the primary consumers of the reports that live on Tableau Server. Since report viewers do not create or publish reports, they only require a licensed account on Tableau Server. When sizing and scaling Tableau Server, the number of concurrent report viewers is the immediate factor considered.



Anticipating User Behavior & How It May Impact a Tableau Environment

Once the user scope has been determined, the next consideration is to fully catalog their expectations of the reporting environment and how they plan on using Tableau.

Below are user and data source questions that should be answered and how those answers might impact the Tableau environment:

Access-Related Questions

Does Tableau Server need to be publically accessible or only accessible from within the corporate network?

Exposing reports to customers external to the primary organization, such as for shareholders or media, has the potential of growing the user base exponentially. Guest Access can facilitate this requirement without purchasing additional individual licenses, but it is only available through an enterprise license.

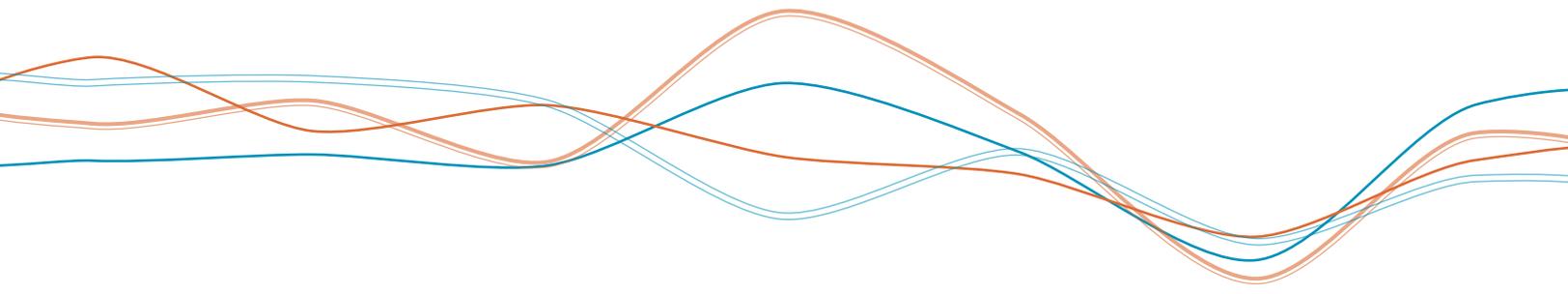
Does the organization use Active Directory?

Tableau offers two methods of adding users to Tableau Server: local accounts and Active Directory accounts. If the organization does not implement Active Directory, it is necessary to manage accounts locally on Tableau Server. This process may be done manually or through a scripted process using [tabcmd](#) - Tableau's command line utility.

Does the organization need to implement Single Sign-On to Tableau Server?

A few options currently exist for Single Sign-On: SAML, Trusted Ticket Authentication and Kerberos. If [SAML](#) for Single Sign-On is already utilized in other applications, enabling it for Tableau can often be done through a configuration of Tableau Server.

[Trusted Ticket Authentication](#) requires a custom development effort but can be leveraged in a multitude of ways.



Data-Related Questions

Where does the data currently live?

Tableau allows for two types of data connections: live (to a database, flat file or service) or extract (imported into Tableau's proprietary Data Engine). Each approach impacts how the Tableau environment is designed and scaled.

While actual Tableau workbooks are quite small, the data they connect to can be quite large. If that data resides in an external system, there is less need for large storage within Tableau. However, data that is extracted from an external source will reside within the Tableau Server environment. This requires more storage and faster disks to process requests.

How large is the data? How wide?

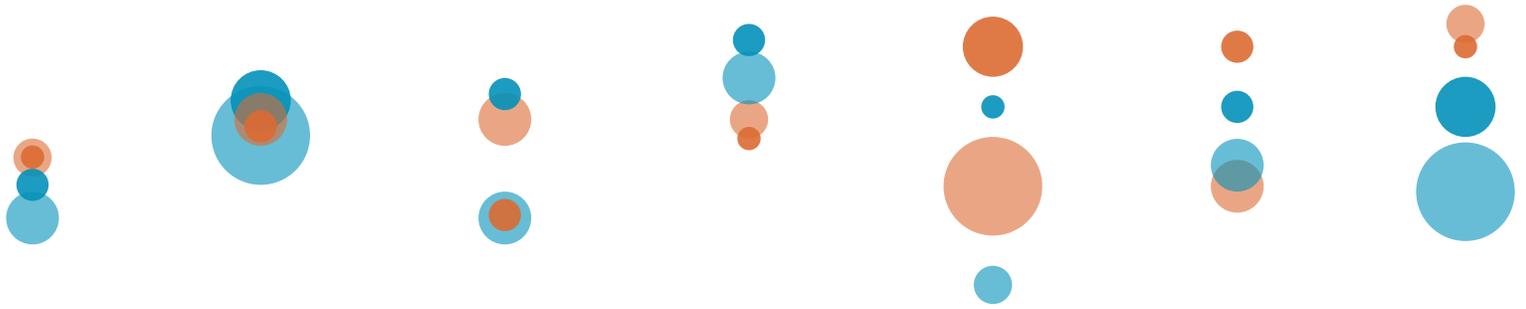
The total number of data rows determines how large the data is while the number of fields describes the width of the data. While a data set of 30 fields may not be very wide, it may still provide a challenge if it's quite large at hundreds of millions of rows. Conversely, a data set of a few million rows may not be that large but can quickly become complex if it contains hundreds of columns of large data types (e.g. descriptive string fields).

Data extracts can have a significant impact on the necessary storage and memory available in the server environment. Also, larger data extracts require more CPU intensive processes, such as extract refreshes.

How often does the data refresh? How often must the data in published reports refresh?

This will likely vary across different groups of users. Do some data sets refresh weekly? Daily? Are some data sets transactional (always up to date)? If the data must always be current, connecting live to the data source may be the best choice. If the data can be refreshed on a scheduled basis, then extracts may offer better performance.

If data is extracted into Tableau's proprietary Data Engine and those extracts must be refreshed during peak usage hours, it may be necessary to isolate certain Data Engine and Backgrounder components on separate servers through a [distributed environment](#). This will ensure data refreshes do not affect performance and end-user experience by creating hardware contention on a single server.



Security-Related Questions

How sensitive is the data? What level of security is required? Is that security already available in the current data environment?

Tableau offers a [robust security framework](#), which allows publishers to secure every report or data source that is published to Tableau by setting group or user-level permissions. However, if data security has already been implemented within the external data environment, it may be more beneficial to leverage that security by implementing live connections and restricting the use of extracts into Tableau's Data Engine. Doing so minimizes overhead by managing data security within a single environment.

If data-level security is implemented through the Tableau Server environment, regular and thorough security auditing reports are essential.

Do the users require row or column-level security?

[Row-level security](#) can be implemented for individual workbooks or across multiple workbooks at the data source level for Tableau Server. Workbook authors can also implement techniques to achieve specific column-level security via custom calculations, but they may face degraded performance the more numerous and complex these calculations become. [Tableau also released support for Kerberos](#), which provides Single Sign-On Top Down to the data layer for certain databases.

It's important to note that, in general, the more that data security is enforced, the less likely Tableau will be able to improve performance by leveraging cached results.

Performance-Related Questions

Do the users require 24/7 uptime?

Tableau Server offers [High Availability configurations](#). Implementing them will require additional hardware and a distributed environment configuration.

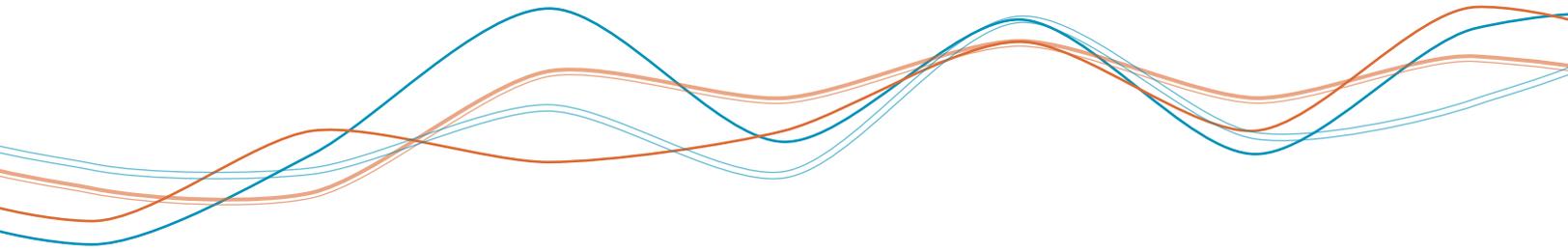
Do the users have an expectation regarding report performance?

Report performance is dependent on many factors, but none are more important than the performance of the connecting data sources. Reports will never be faster than the speed at which the data sources can retrieve and return results to Tableau. Other contributing factors may also include the [design of the report and the configuration of the server environment](#).

Once the expectations of the reporting environment have been defined, it is time to begin sizing the hardware to accommodate these requirements by designing the base configuration and scaling it to match the amount of users and the level of activity anticipated.

Additional Resources

- + [tabcmd](#)
- + [SAML](#)
- + [Trusted Ticket Authentication](#)
- + [Distributed Environment](#)
- + [Robust Security Framework](#)
- + [Row-Level Security](#)
- + [High Availability Configurations](#)
- + [Basic Performance Tips: Report Design & Server Configuration](#)



Step 2:

Sizing Hardware & Configuring Tableau Server

Sizing hardware and configuring Tableau Server is an iterative process heavily dependent on the expected use case. Tableau [provides a set of recommendations](#) as a starting point for base hardware and for base configuration. Once the base configuration is selected, it can be optimized and expanded to the expected use cases and user demands.

Recommended Base Hardware

Deployment Type	CPU	RAM	Free Disk Space
Evaluation/proof-of-concept (32-bit Tableau Server)	2-core	4 GB	15 GB
Evaluation/proof-of-concept (64-bit Tableau Server)	4-core	8 GB	15 GB
Production (single computer)	8-core	32 GB	50 GB
Enterprise	16-core	32 GB or more	50 GB or more

The recommended configurations in this whitepaper will assume machines of the enterprise deployment type above, each running a 64-bit installation of Tableau Server unless stated otherwise.

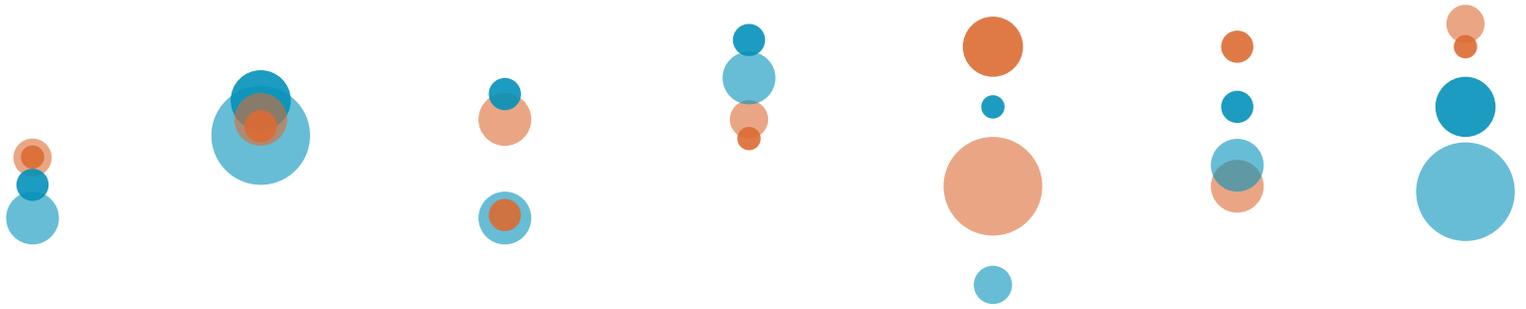
The Tableau Processes

Tableau has several configurable processes. Below is a quick summary of each along with considerations that should be made when configuring them. See [The Tableau Server Processes](#) page in the administrative guide for more information on each process.

VizQL

The VizQL process is the primary work horse in Tableau Server. It is tasked with querying data wherever it may live and ultimately rendering that data as visualizations.

- In a 64-bit installation, Tableau recommends configuring two VizQL processes - one primary and one additional backup process, should the primary become inaccessible. This is recommended for enterprise deployments.
- In a 32-bit installation, it may be necessary to spin up additional processes as the number of concurrent users grow in the environment. This need is driven by the memory limit a 32-bit process may address.



Backgrounder

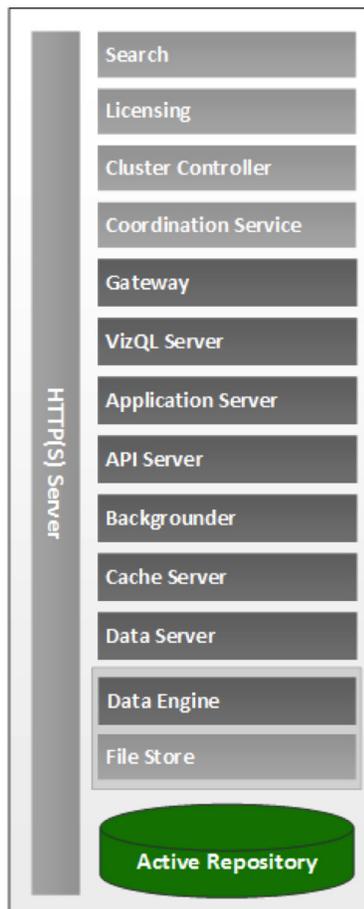
The Backgrounder process is tasked with many behind the scenes operations, including extract refreshes. Configurations vary depending on the level of extract usage and their complexity. A single Backgrounder process can use up to 100% of a CPU core, but multiple processes can be configured to manage parallel processing of tasks.

Application Server, Data Server and API Server

The Application Server handles all the web application tasks, such as authentication and browsing. The Data Server manages connections to Tableau Server-hosted data sources. User load for both can be typically handled by a single process each, although two can be used for further redundancy. The API Server handles REST requests. It is not critical to normal operations of Tableau Server.

Data Engine and File Store

The Data Engine's primary responsibility is to load extract data into memory and fulfill query requests against the data. The amount of memory used depends heavily on the size and number of extracts. The Data Engine can only have a single active process, but like the Repository process (see below), the Data Engine can be configured for High Availability by configuring a standby process on a separate server within a distributed environment. The File Store is responsible for replicating the extract files across Data Engine nodes in the Tableau Server cluster.



Repository

The Repository is Tableau Server's database that stores workbook and user information. It can remain on the primary server unless the deployment requires High Availability. Only a single active process can be configured, but like the Data Engine, a standby process can be configured on a separate server within a distributed environment.

Cache Server

The Cache Server provides a shared, in-memory query cache that speeds up user experience in many instances. The Cache Server is single-threaded, so if contention arises, you may need to increase the number of Cache Server processes.

Cluster Controller and Coordination Service

These processes are installed by default on every node. The Cluster Controller is responsible for monitoring the various components, detecting issues and performing failover as required. The Coordination Service is responsible for ensuring a quorum exists for making decisions during failover.

Search & Browse

The Search & Browse process is responsible for fast searching, filtering and displaying content metadata. The process is primarily memory bound and secondarily I/O bound. Memory requirements scale with the amount of content the server has.

Recommended Base Configurations

In each of the configurations below, N refers to the number of CPU cores.

Single Machine

A single machine may be sufficient for smaller deployments. As user load increases, certain processes may become CPU, I/O or network bottlenecks. In such cases, a distributed environment consisting of multiple machines can be deployed to scale capacity.

An added benefit of a distributed environment is it allows for High Availability of the critical Data Engine and Repository processes.

Highly Available Distributed Environment

Distributed environments accomplish two objectives:

- Distribute the load across multiple machines
- Add standby processes that will automatically become active should one machine go down

For most environments, there is little or no concern about processes like the Data Engine or Backgrounder causing resource contention on the machines.

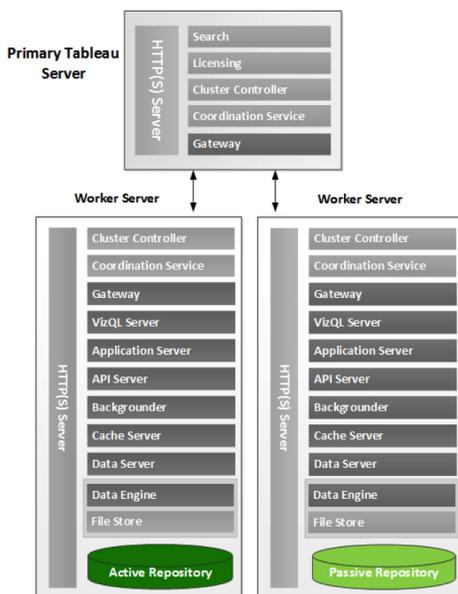
Each worker is configured similarly; however, the active Data Engine and Repository processes are split between the machines for further redundancy. Since the VizQL process handles much of the load for standard user activity, additional workers that contain the VizQL or other processes face contention.

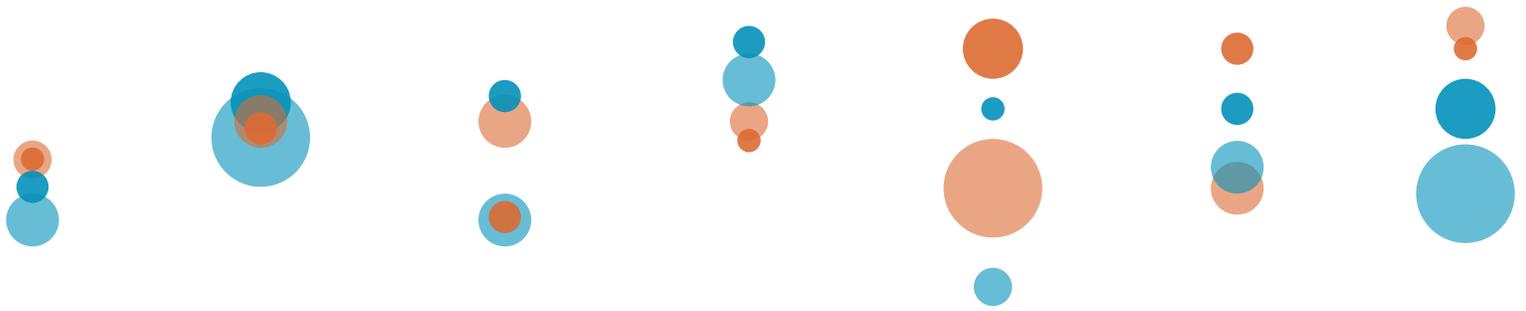
Configuring a distributed environment is a relatively straightforward process. Tableau provides two installers: one for the primary server, which handles search and licensing features, and a second worker installer for all additional machines.

See [Tableau's guide to distributed environments](#) for more information.

More Information

To learn more about server configurations and their respective impacts on performance, see Tableau's [Improve Server Performance document](#).





Choosing an Extract, Live Connection or Hybrid Environment

One extremely important component of sizing and optimizing performance of Tableau Server is the visualizations' data sources. Ensuring that Tableau's Data Engine is as effective and efficient as possible requires that a few additional details are considered.

Data Extracts

Since extracts are loaded to and run in memory, servers running the Data Engine process need to have sufficient RAM to store as much of the currently-used extracts as possible. These servers will also need to have sufficient disk speed to quickly read an extract into memory when required. It's for these reasons that the Tableau Data Engine should exist on multiple machines in the Tableau cluster for redundancy.

If heavy extracts are frequently used, one Backgrounder process per concurrently-refreshing extract is recommended. Backgrounders also consume heavy CPU and memory resources, which should be separated from other processes if possible.

Live Connections

If the Tableau environment uses live connections, there are a few things worth noting. First, live connections may be queried from one of two processes within Tableau Server.

- If the workbook is connecting while using a published [Tableau Data Server](#) data source, the query will be executed from the Data Server process.
- If the workbook is connecting directly to the data source, the query will be executed from the VizQL process.

Hybrid Connections

Most implementations use both Live and Extract data sources to varying degrees. All of the above performance characteristics and considerations apply to hybrid environments. Any of the recommended configurations can be used for a hybrid environment. The expected usage of extracts should drive the final decision.

Tableau Server provides the best user experience when queries return fast. For this reason, Tableau Data Extracts are recommended when not connecting to fast analytical databases such as Google BigQuery, Teradata, Vertica, Netezza, etc. When using extracts is not permissible, the database should be [tuned for best performance](#).

Tableau will only perform as fast as its data source.

Additional Resources

- + [Hardware & Configuration Recommendations](#)
- + [The Tableau Server Processes](#)
- + [Distributed Environment](#)
- + [Improve Server Performance Document](#)
- + [Database Query Performance](#)



Step 3: Load Testing

Once a base configuration has been deployed, establishing a performance baseline is the next priority. This is accomplished by load testing, which determines how many users the configuration can reasonably support without a significant degradation of performance. Load testing will reveal whether the current environment adequately covers the expected user base and their level of activity. If not, the environment should be scaled up or out as necessary and tested again.

Why Load Testing?

Every organization is unique in their exact needs from Tableau Server. Some organizations might make use of heavy extracts with very complex visualizations. Other organizations might make use of live connections and simple visualizations.

The best way to understand what performance looks like for specific scenarios is to use load testing software to simulate the production load. Load testing also offers the ability to run the exact same situation multiple times for comparing results of any configuration changes.

Load testing simulates a variety of performance scenarios that will determine if the Tableau environment can still run efficiently within your required usage parameters or if it is necessary to scale.

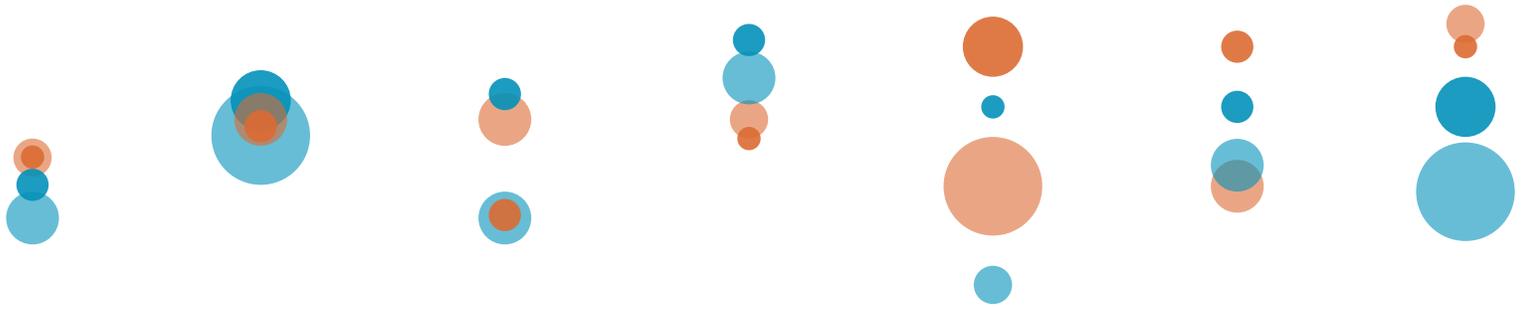
Top Causes of Poor Performance

Before discussing the load testing process, it's important to first highlight some of the common sources of poor performance, regardless of the number of active users or design of the server environment.

Slow Data Sources

Tableau visualizations will never perform faster than the time it takes to retrieve the data used to build them. This simple fact often goes unaccounted for and at great cost. As discussed previously, one option is to leverage Tableau's Data Engine by taking extracts of the data so that it can be loaded and retrieved by Tableau Server locally.

Other options include fast, columnar analytical databases such as Tableau partners HP Vertica, Teradata, Netezza or others. Cloud solutions are also available, including Google BigQuery or Amazon Redshift.



*THERE MAY BE A
SMALL NUMBER
OF USERS THAT
INTERACT VERY
HEAVILY WITH THE
DASHBOARDS.*



Workbook Complexity

Workbook complexity plays a very important part in the performance equation. If a dashboard contains a large number of sheets, marks, headers, quick filters or actions, there's more work required to query the data and render the view. Tableau Server takes this into consideration when determining whether to render the view client-side or server-side.

Because of these inherently time-consuming components, [authoring for performance](#) is highly recommended from the start of authoring views. Authoring for performance will assist in building the workbook in such a way that guides users towards the data most important to their needs.

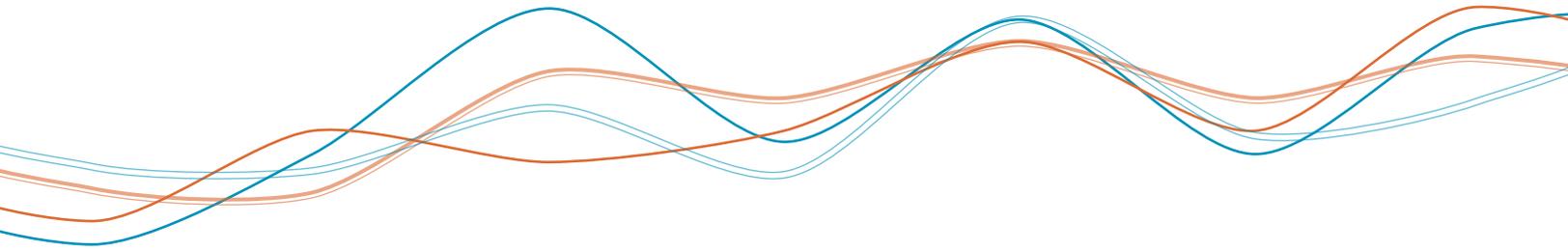
How complex are your workbooks?

- Number of sheets, marks or headers in dashboard
- Number of quick filters
- Number and complexity of calculations
- Number of reference lines or annotations
- Levels of interactivity

Types of User Activity

From a Tableau Server standpoint, most typical use cases involve a mix of a few different types of workloads. There may be a small number of users who interact very heavily with the dashboards. More users may interact moderately with your dashboards. Finally, a subset of users might only load the dashboard and not interact with it.

Be aware that each of these use cases impact Tableau Server differently. Complex/heavy interactions will require more resources whereas view-only interactions will scale well with fewer resources (see Step 4).



Additional Resources

- + [Authoring for Performance](#)
- + [Explaining Tableau Server Scalability](#)

Methodology

The principle of load testing is to simulate the users' interactions with Tableau Server. This can be achieved any number of ways but is generally accomplished by walking through a typical user interaction, such as browsing a site and interacting with a workbook. The session is recorded, either manually or via a piece of software, and put into a load testing tool.

After the session is defined within the load testing tool, this process is repeated until all reasonable user interactions are recorded. A test is then built by playing through those sessions a certain number of times in parallel, where each time is effectively a simulated user.

When performing a load test, simulating a "time between clicks" is essential to allow for the time users will be considering the data on the screen. It's also recommended that an accurate workload mix is represented by splitting simulations across view-only scenarios and scenarios with heavy interactions.

It may also be relevant to ramp up a test from a smaller number of users to a larger number of users to represent the notion that traffic generally starts small and increases as the business day progresses.

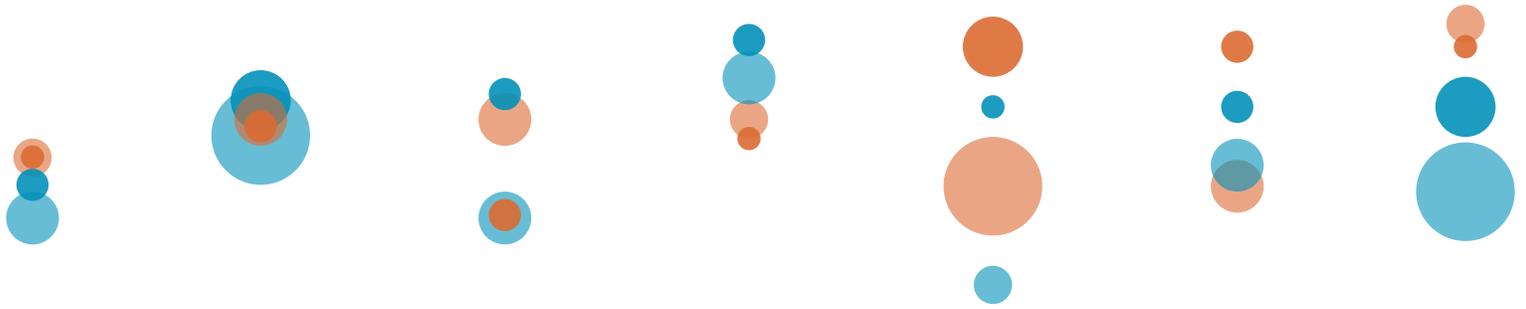
For more information surrounding the methodology of load testing Tableau, see [Tableau Server Scalability Explained](#).

Load Testing Tools

Load testing can vary from simple to complex. There are many tools that facilitate effective testing. Each tool has their own pros and cons, as presented below:

HP LoadRunner HP LoadRunner is one of the most common load testing tools. It provides a very powerful feature set that includes the ability to record and modify tests, configure and use variables, and distribute the testing workload. HP LoadRunner can test Tableau Server by making HTTP requests to the server in much the same way a user's web browser would. The upside of this methodology is things like tooltips, grabbing images and dynamic AJAX interactions can be included in the test. HP LoadRunner does not support executing client-side JavaScript, however. As a significant portion of Tableau Server relies on JavaScript and rendering HTML5 canvas objects, HP LoadRunner will not demonstrate the exact load times users would experience, rather close approximations.

Apache JMeter Apache JMeter is an open-source solution similar to HP LoadRunner in methodology. JMeter also provides tools for recording and modifying tests, variables and other components. It also provides an easy-to-use distributed testing interface. It shares the same HTTP-request methodology that underpins LoadRunner and has similar drawbacks.



Selenium – Browser Based Testing

Selenium is a tool that is built to automate web browser interactions. As a browser automation tool, it is capable of simulating the exact user experience, including tool-tips, JavaScript execution and HTML5 canvas render times. Unlike LoadRunner and JMeter, Selenium has a much higher memory and CPU footprint, therefore requiring more resources to accomplish the same level of testing. Selenium is also more of a browser interaction framework, which means that tests are not generally created by recording interactions but by code.

Step 4: Scaling to Meet Demand

As the user base grows... the deployment will eventually need to scale. Tableau Server has the ability to scale up and out as necessary. Scaling up Tableau Server might include adding more physical resources to the same server or increasing the number of particular processes. Scaling out Tableau Server involves adding more servers into the environment and load balancing or distributing the workload.



Determining When to Scale

Here are a few indicators that it may be time to scale:

- Out of Memory errors
- Reports running longer than usual
- Resource contention – Disk, Memory, Processor or Network

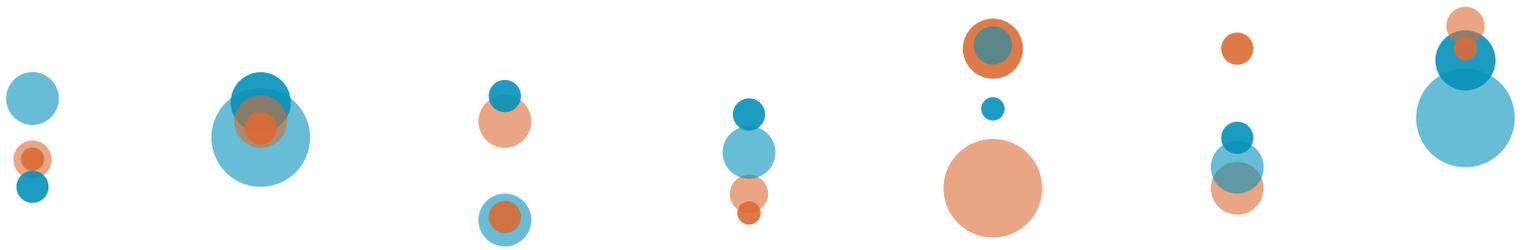
Each is presented in more detail below:

Out of Memory Errors

Out of Memory errors are more common in 32-bit versions of Tableau Server, since those versions are only capable of addressing up to 3 GB of memory. 64-bit versions can address significantly more memory, meaning that an Out of Memory error indicates your server's memory capacity has been used in its entirety. In some cases, memory saturation in 64-bit versions can lead to a lot of swapping instead of out of memory errors, leading to poor performance. As a result, it is also worth monitoring swap usage in addition to overall memory usage.

Long Report Runtimes

Long running reports are the most common indicator of contention somewhere within the system. As the user base grows, reports that load slower than they have historically can indicate that it is necessary to scale up or out.



Additional Resources

+ [Guide to Distributed Environments](#)

Resource Contention

Lastly, another symptom is resource contention on the servers themselves. Any of the following are indicative:

- Processor utilization that is at or near 100% during heavy utilization periods
- Disk transfer speeds are unacceptably slow
- Network transfer speeds are unacceptably slow
- Memory is at or near 100% utilization

For further information on how to track these metrics, see [Step 5: Maintaining & Monitoring the Production Environment](#).

How to Scale Up

Many implementations of Tableau Server run within a virtual server. Scaling up these types of implementations often can be as simple as powering down the server and increasing the number of virtual CPUs or the amount of memory dedicated to Tableau Server.

If physical servers are being used, additional machines must be purchased and physically installed.

Once installed, Tableau Server will have immediate use of these resources. In certain situations, configuration changes may also be required.

How to Scale Out

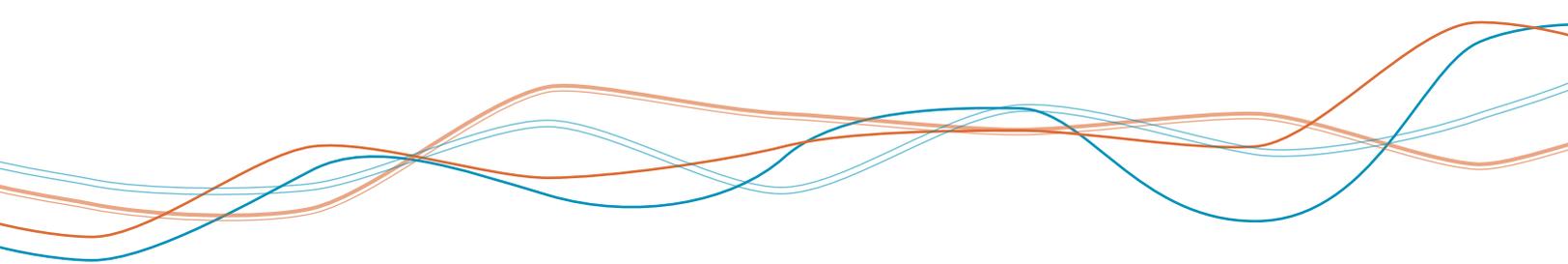
Scaling out is relatively simple for Tableau Server. In each of the recommended distributed environments, an organization can scale out by adding workers that contain the processes you need, using Tableau's [guide to distributed environments](#).

Step 5:

Maintaining & Monitoring the Production Environment

At this stage of the deployment, Tableau Server is installed, configured and has an active user base. The final step is maintaining the environment and ensuring all operations continue to perform well and run smoothly. This can be accomplished through a series of activities:

- Using Tableau's administrative dashboards
- Creating a proper run and maintain plan by:
 - (RUN) Closely monitoring hardware and processes across the distributed environment
 - (RUN) Automating common administrative tasks such as backups and the adding/removing of Active Directory groups/users
 - (MAINTAIN) Auditing assigned permissions, published content and logs
 - (MAINTAIN) Automating archival and cleanup of unused content
- Scaling when needed



Using Tableau's Administrative Dashboards

While the sections below describe ways to augment system monitoring, Tableau Server does ship with [standard administrative views](#) that analyze:

- Traffic to Views
- Traffic to Data Sources
- Actions by All Users
- Actions by Specific User
- Actions by Recent Users
- Background Tasks for Extracts
- Background Tasks for Non Extracts
- Stats for Load Times
- Stats for Space Usage

Analysis	
Dashboards that monitor Tableau Server activity.	
Views	Analysis
Traffic to Views	View count, viewers, and viewer behavior for published views.
Traffic to Data Sources	Data source usage, users, and user behavior for published data sources.
Actions by All Users	Actions for all users.
Actions by Specific User	Actions for a specific user, including items used.
Actions by Recent Users	Recent actions by users, including last action time and idle time.
Background Tasks for Extracts	Completed and pending extract task details.
Background Tasks for Non Extracts	Completed and pending background task details (non-extract).
Stats for Load Times	View load times and performance history.
Stats for Space Usage	Space used by published workbooks and data sources, including extracts and live connections.

Accessing the Administrative Dashboards

Most if not all of these views require a user to have administrator privileges on Tableau Server. Optionally, these views can be republished to a public site so all users can access. The standard views, however, do not keep the history that is often necessary for administrators to gauge either workbook performance or Tableau adoption over time.

These views can be improved by adding more history and drilling deeper into what each user is doing on each view and/or each Tableau dashboard. [Create Custom Administrative Views](#) has more information on how to enable access to Tableau Server's database.

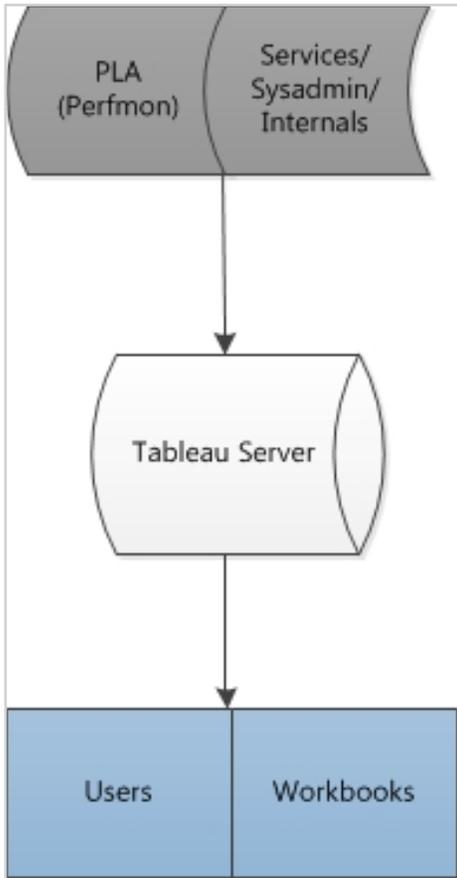
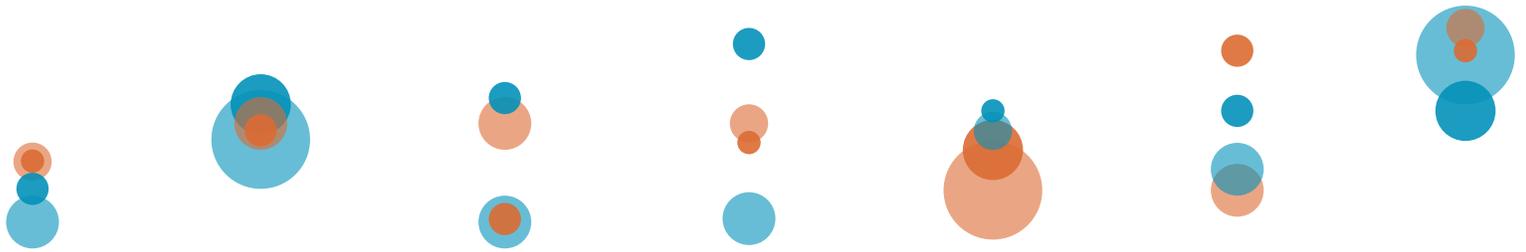


Figure 1: Top portion is the “run” section; bottom section is the “maintain” section.

Note

The term “Perfmon” describes the technology embedded in Windows that allows a user/administrator to examine how programs affect a computer’s performance both in real time and by collecting data for later analysis.

Tableau Server: Run and Maintain

When requirements change or when needs expand, a proper “run and maintain” plan saves time and increases the functionality of Tableau Server. From initial cluster configuration to automated scripts, the run and maintain plan is perfect for a single node installation all the way up to a multiple machine cluster.

Run

In any Tableau Server setup, scripts and procedures should be scalable. While one ad hoc script may work in a particular situation (and there are many practical uses for them), a fully robust script is one that increases productivity and functionality for both users and machines.

Perfmon

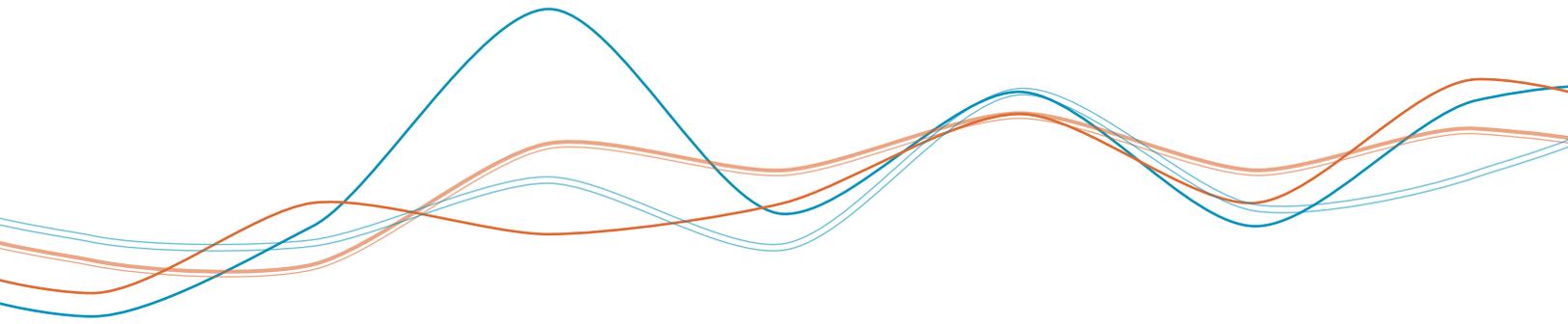
Performance Monitor (also known as Perfmon) is an excellent way to gauge system performance. It is highly recommended, as Perfmon has deep integration into the system and its many components.

For most testing and monitoring, the following counters are used:

- \System\Processor Queue Length
- \Processor(*)\% Processor Time
- \PhysicalDisk(*)\Disk Reads/sec
- \PhysicalDisk(*)\Disk Writes/sec
- \PhysicalDisk(*)\Current Disk Queue Length
- \PhysicalDisk(*)\Avg. Disk Read Queue Length
- \PhysicalDisk(*)\Avg. Disk Write Queue Length
- \PhysicalDisk(*)\Avg. Disk Queue Length
- \PhysicalDisk(*)\Disk Transfers/sec
- \PhysicalDisk(*)\Avg. Disk sec/Transfer
- \Network Interface(*)\Bytes Total/sec
- \Network Interface(*)\Packets Outbound Errors
- \Network Interface(*)\Packets Outbound Discarded
- \Network Interface(*)\Output Queue Length
- \Network Interface(*)\Packets Received Errors
- \Memory\Available MBytes
- \Memory\Committed Bytes

There are a significant amount of other counters available. To find the full list for the Processor set, simply open an instance of PowerShell and run the following command:

Get-Counter -ListSet Processor | Select-Object -Expand Paths



A typical script used in combination with Perfmon would help extract information from the generated logs by doing the following:

- First, it takes the path of the csv, tsv or blg file from the Perfmon log and parses for the specified counters.
- Second, it averages the metrics over the timeframe specified by the user.
- Finally, after it's done parsing the file and averaging the metrics, it exports to csv so that the data can be consumed by Tableau for trending, analysis and the like.

Run Plan Checklist

A general run plan should also include, but is not limited to, the following scripts/procedures:

Proper setup of the “Run As User” account.

This includes granting the necessary folders, both Program Data and Program Files, the correct permissions. With a distributed configuration or a single machine install, this uses Windows remoting to configure each machine with the desired AD account via [Icacls](#).

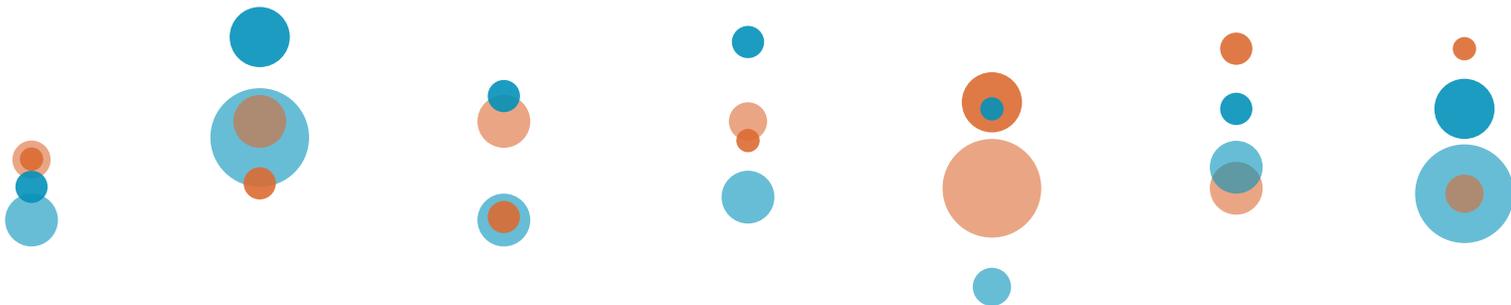
Confirm Tableau Services are running on each machine (Secondary Login, FlexNet licensing, etc.).

Name	StartName	StartMode	Service Running?
FlexNet Licensing Service	LocalSystem	Manual	✘
FlexNet Licensing Service 64	LocalSystem	Manual	●
seclogon	LocalSystem	Auto	●
tablicsvr	LocalSystem	Auto	●
tabsvc	[REDACTED]	Auto	●
FlexNet Licensing Service 64	LocalSystem	Manual	●
seclogon	LocalSystem	Auto	●
tablicsvr	LocalSystem	Auto	●
tabsvc	[REDACTED]	Auto	●

It then queries WMI (Windows Management Instrumentation) for the necessary properties for all Tableau’s services and creates a Tableau view.

Confirm Tableau process (VizQL, Data Engine, etc.) are running and email user(s) if any processes go down.

Using the “[systeminfo](#)” section of Tableau Server’s admin views, this will enable the administrator to see what processes are down. It is similar to running “tabadmin status -v” in the console.



**A PROPERLY
INSTALLED “RUN”
PLAN WILL ALLOW
FOR THE QUICK
ADDITION OF
MACHINES TO THE
SERVER CLUSTER
AND MAKE IT A
STRUCTURALLY
SOUND
ENVIRONMENT.**



PLA (Performance Logs and Alerts) technology to monitor and parse Perfmon log files (to name a few) for hardware and/or network issues.

Using performance counters via PLA, this script parses and pulls out the necessary counters (Processor, Hardware, Network, Memory, etc.) and averages them over time. This is also used to create a Tableau workbook so that the administrator can monitor trends over time or look at a live feed. See Permon section above for more details.

Scheduling [Tableau Server backup and maintenance](#).

Automating Active Directory (AD) synchronization with Tableau Server.

Tableau provides a way to [sync Active Directory groups and users](#), but it is a manual process which requires an admin to initiate the process. Automating this process through a script using tabcmd gives greater control to the administrator to actively and automatically sync groups and/or individuals from Active Directory to Tableau Server. Additionally, one can set the licensing level and publishing rights.

Verify Data Engine and Repository folders for cluster synchronization (in highly available distributed environments).

This process can be managed through a script that checks to ensure the necessary folders are synced before taking further steps in the HA/Failover process. Not confirming the folder sizes could result in loss of data.

A properly installed “run” plan will allow for the quick addition of machines to the server cluster and make it a structurally sound environment.

Maintain

Now that Tableau Server is setup and stable, maintaining what is on it becomes an ongoing priority. As the organization matures, there are going to be many workbooks (with or without extracts) that take up valuable space and, without oversight, could remain there indefinitely. Since publishers have ultimate authority in assigning permissions to content that is published, it’s also necessary to audit those permissions to ensure they are in compliance.

Below are common questions that an administrator must ask after the server environment has been operational for a while:

- How many distinct users are there?
- How many distinct users for the past 30 days?
- What does user growth look like for the past six months?
- What workbooks have extracts greater than 30 MB? Data sources?
- How has my dashboard been trending for the last quarter? Two quarters?
- Can we identify stale content and notify its owner that it will be archived?



Scripts and Procedures

The general maintain plan should include, but is not limited to, the following scripts/procedures:

TDE triggers

Listens to the PostgreSQL workgroup database for changes and triggers a Tableau Extract to run on Tableau Server. It's also possible to trigger Tableau Extracts to run after certain events occur outside of Tableau Server (such as an ETL process).

Email owners when an extract fails

Email (or text message) is immediately sent to user if his/her extract fails. Similar to the extract triggers in that it listens to the Postgres database for updates/changes.

Email owners when their workbook has not been viewed in 30/60/90 days with intent to archive and remove from Tableau Server

This is probably the biggest way to ensure the environment is clean and controlled. This script queries the PostgreSQL workgroup database for workbook metadata. Once it meets the desired parameters (30/60/90 days), tabcmd takes over and pulls down the appropriate workbooks then emails the users.

Email administrators when a rolling 30 concurrent user level reaches a certain level

Understanding user growth is a great way to expand and scale the infrastructure. This script queries PostgreSQL workgroup database and takes a daily snapshot of the users. It is then used for a Tableau workbook to see trends over time.

Parse log files for common fields used in Tableau Server data sources

An excellent way to analyze fields Tableau viewers/interactors are actually using is to check the logs for the queries issued by the Tableau Server data source. This script would parse those queries and displays a list of common fields (a.k.a. dimensions/measures).

Additional Resources

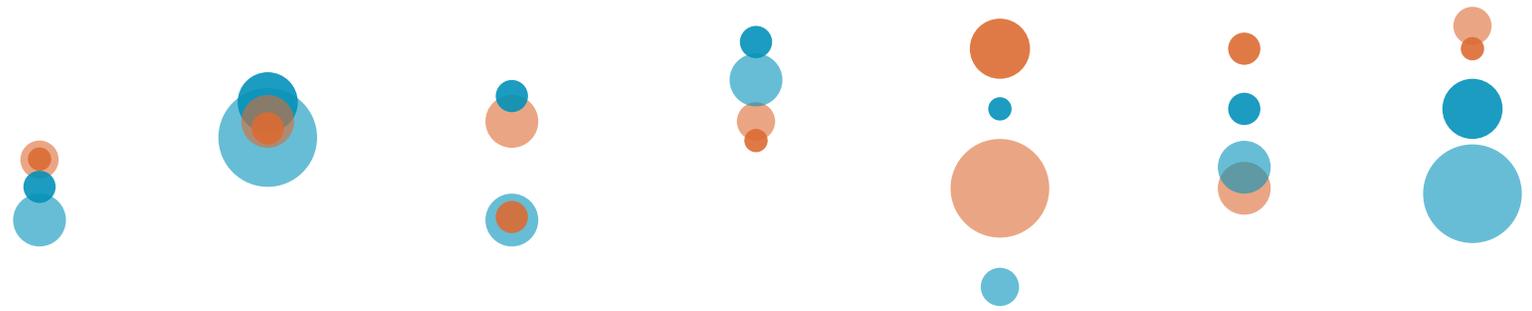
- + [Standard Administrative Views](#)
- + [Create Custom Administrative Views](#)
- + [Icacs](#)
- + [systeminfo](#)
- + [Tableau Server Backup and Maintenance](#)
- + [sync Active Directory groups and users](#)
- + [Script using tabcmd](#)
- + [Upgrading Tableau Server](#)
- + [Database Restore](#)

Upgrading Tableau Server

The process of upgrading Tableau Server is designed to be as simple and straightforward as possible. In all cases, one should always take a backup before doing any upgrades and installs. Tableau provides a guide that details the [upgrade process and provides a pre-upgrade checklist as well](#). The general process of a Tableau Server upgrade includes:

- Backing up Tableau Server
- Uninstalling the previous version of Tableau Server
- Installing the new version of Tableau Server

All published content, users and settings from the previous Tableau Server installation should be automatically migrated to the new install. In the case of failure, the [backup can be restored](#).



Conclusion

While every Tableau implementation varies, this white paper has highlighted the steps critical to success in any enterprise environment:

Step 1: Determining the expected use cases of the environment

Step 2: Size the hardware and server configuration to cover the expected use cases

Step 3: Load test the environment to determine the baseline performance of the environment

Step 4: Scale your environment up (better hardware) or out (more hardware) as needed

Step 5: Maintain and monitor the environment to keep things running smoothly

By using these steps as a guide and planning tool, the administrator will avoid most of the pitfalls and obstacles in an enterprise Tableau deployment. Within each section, there are additional resources related to each step to further learning and understanding.

Recognition

- + Tableau Software's First Gold Partner (2012-present)
- + Tableau VAR Partner of the Year (2013, 2014)
- + Tableau Partner Excellence Award (2013)
- + Tableau Technology Partner (Power Tools for Tableau)
- + Alteryx Premier Partner (2013-present)
- + HP Vertica Partner (2012-present)
- + Dell Partner and Dell Partner of the Year (2010)
- + Inc. 5000 List of America's Fastest Growing Companies (2008-2015)
- + CRN Fast Growth 150 Award (2010-2015)

About InterWorks

Founded in 1996, InterWorks was formed to address the growing IT infrastructure and support market. InterWorks integrated business intelligence and Tableau support, services and growth has been exponential since.

In recent years, InterWorks has expanded both nationally and globally- opening offices from coast to coast as well as entering the European market. As a result, InterWorks continues to recruit uniquely talented employees that welcome challenges, possess positive fortitude and add value to the entirety of the organization.

As Tableau's inaugural Gold Partner, InterWorks provides industry-leading Tableau services and data solutions with an unrivaled team of consultants. InterWorks has proven to be the leader in global Tableau enterprise implementations with one of the most extensive client bases in the world.

The ultimate mission of InterWorks is to provide unprecedented and unmatched customer service. We love what we do. Most of all, we love helping others through what we do. We know that the success of our clientele is directly linked to our own. This mentality is evident in everything we do at InterWorks and has led to our continual growth. 🚀