Tableau for the Enterprise: An overview for IT

Table of contents

Introduction 3

Architecture 4
- Data layer 5
- Data connectors 6
- Tableau Server components 7
- Gateway/load balancer 8
- Clients: Web browsers and mobile apps 8
- Clients: Tableau Desktop 8
- Customization and extensibility 9

Data strategy 10
- Access a variety of data sources 10
- Using extracts for efficiency and offline access 11
- Data governance: The Tableau Data Server 11
- Report governance 13
- Give users more control with subscriptions 15
- Change management 15

Metadata management 16

Mobile deployment 17

Deployment models 19
- Simple configuration 19
- 3-server (24-Core) cluster 19
- 5-server (40-Core) cluster 20
- High availability cluster 20
- Virtual machine or cloud-nased deployment 20

Security 21
- Authentication – Access security 21
- SAML authentication 21
- OAuth authentication 22
- Authorization (roles and permissions) - object security 22

Data – Data security 23
- Network – Transmission security 23

Scalability 24
- Scalability performance results 25

Performance 26
- 64-Bit Tableau Server 26
- Self-service performance triage with performance recorder 27
- Optimized data extracts deliver faster response to users 27
- Reduced workload through secure shared sessions 28
- Local rendering lets users interact with data in real-time 28
- Dashboards rendering with optimized views 28

Conclusion 29

Further reading 29

About Tableau 30

A new generation of business intelligence software puts data into the hands of the people who need it. Slow, rigid systems are no longer good enough for business users or the IT teams that support them. Competitive pressures and new sources of data are creating new requirements. Users are demanding the ability to answer their questions quickly and easily. And that's a good thing.

Tableau Software was founded on the idea that data analysis and subsequent reports should not be isolated activities but should be integrated into a single visual analysis process—one that lets users quickly see patterns in their data and shift views on the fly to follow their train of thought. Tableau combines data exploration and data visualization in an easy-to-use application that anyone can learn quickly. Anyone comfortable with Excel can create rich, interactive analyses and powerful dashboards and then share them securely across the enterprise. IT teams can manage data and metadata centrally, control permissions and scale up to enterprise-wide deployments.

This flexibility helps IT organizations get out of the reporting backlogs and empower the business users to be self-reliant. But this flexibility doesn't come at the expense of IT. In fact, it's the opposite. IT can deliver this service in a scalable, secure, and easy-tomanage system which meets the Service Level Agreement of the organization. This overview is designed to answer questions common to IT managers and administrators and help them support Tableau Server deployments of any size

To create a data-driven culture for today's digital era, one that's ready to tackle today's business challenges with accuracy and speed, organizations must not only invest in new technology, but also new ways of delivering that information, including the people required to lead the charge. This parallel cultural shift is a fundamental change in the relationship between IT and the business; they are two partners working to collect and mine data, but also to refine it and deliver the right information on demand. It's when IT and the business work together that organizations can turn the pipe dream of a self-service analytics culture into a reality.

Architecture

Tableau Server has a highly scalable, n-tier client-server architecture that serves mobile clients, web clients, and desktop-installed software. There are 2 primary components of a Tableau Solution – Tableau Desktop and Tableau Server. This modern ecosystem presents three business challenges, which we'll look at now.

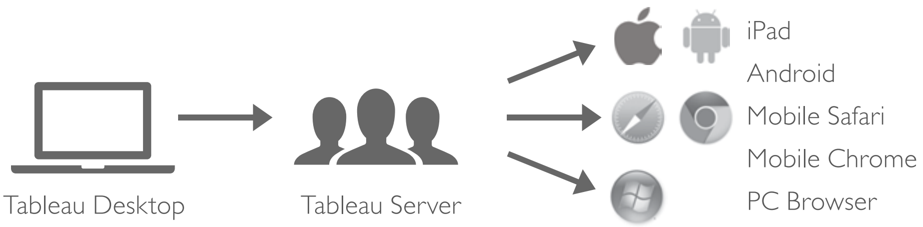


Figure 1: Tableau Server provides a scalable solution for creation and delivery of web, mobile and desktop analytics.

Tableau Server is an enterprise-class business analytics platform that can scale up to hundreds of thousands of users. It offers powerful mobile and browserbased analytics and works with a company's existing data architecture, lifecycle management, security and governance constraints.

Tableau Server meets Enterprise requirements including:

- Scalable: Tableau Server can scale-up and scale out to meet the your enterprise needs. The server can scale up by adding more CPU and RAM. Each component of Tableau Server is multi-process and can be configured based on your usage patterns. Additional scale can be achieved by adding additional nodes which can be configured to meet the demands on then organization.
- Highly Available: Provides high availability with internal cluster management, Supports external load balancers.
- Secure: SSL, Encrypts internal traffic, supports Active Directory, SAML, and OAuth integration
- Easy to manage: Straightforward administration from user management to upgrades
- Extensible: Offers robust APIs

The following diagram illustrates Tableau Server's architecture:

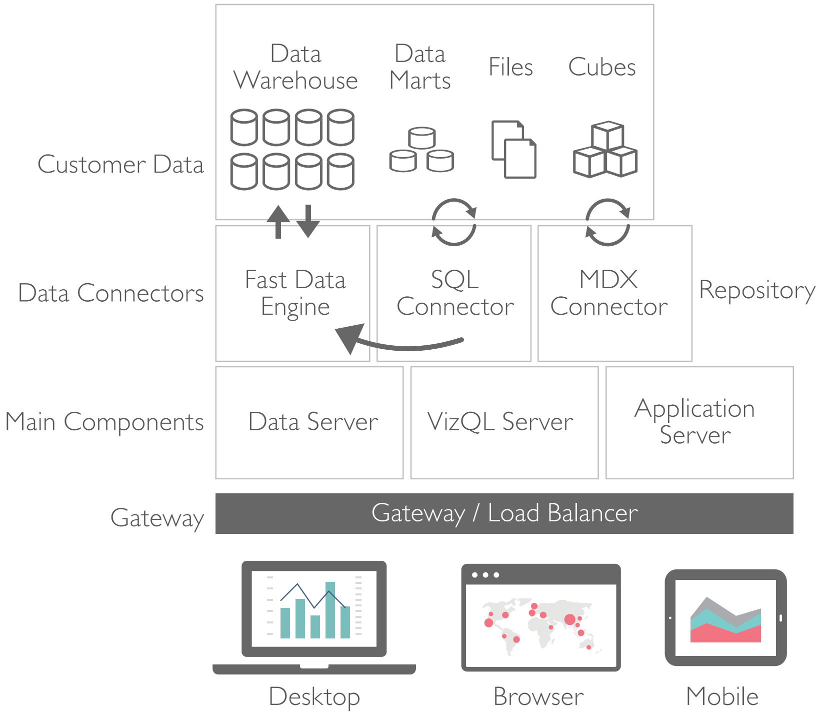


Figure 2: Tableau Server architecture supports fast and flexible deployments.

Below, we explain each of the Tableau Server layers, starting with customer data.

Data Layer

One of the fundamental characteristics of Tableau is that it supports your choice of data architecture. Tableau does not require your data to be stored in any single system, proprietary or otherwise. Most organizations have a heterogeneous data environment: data warehouses live alongside databases, whether on-premise or in the cloud. Cubes, and flat files like Excel are still very much in use. Tableau can work with all of these simultaneously. You do not need to bring all your data in-memory unless you choose to. If your existing data platforms are fast and scalable, Tableau allows you to directly leverage your investment by utilizing the power of the database to answer questions. If this is not the case, Tableau provides easy options to upgrade your data to be fast and responsive with our in-memory Data Engine.for new technologies. Flexibility is critical to creating and maintaining your distinct competitive advantage. When you are considering a future-friendly analytics architecture, avoid vendor lock-in to proprietary architecture as that can significantly hinder your ability to be agile in the future.

Data Connectors

Tableau includes over 40 optimized data connectors for data sources such as Microsoft Excel, SQL Server, Google BigQuery, Amazon Redshift, Oracle, SAP HANA, Salesforce.com, Teradata, Vertica, Cloudera and Hadoop, and new data connectors are added on a regular basis. There is also a generic ODBC connector for any systems without a native connector. Tableau provides two modes for interacting with data: live connection or in-memory. Users can switch between a live and in-memory connection as they choose.

Live connection: Tableau's data connectors leverage your existing data infrastructure by sending dynamic SQL or MDX statements directly to the source database rather than importing all the data. This means that if you've invested in a fast, analytics-optimized database like Vertica, you can gain the benefits of that investment by connecting live to your data. This leaves the detail data in the source system and sends the aggregate results of queries to Tableau. Additionally, this means that Tableau can effectively utilize unlimited amounts of data. In fact, Tableau is the front-end analytics client to many of the largest databases in the world. Tableau has optimized each connector to take advantage of the unique characteristics of each data source.

In-memory: Tableau offers a fast, 64-bit ready, columnar in-memory Data Engine that is optimized for analytics. You can connect to your data and then, with one click, extract your data to bring it in-memory to perform queries in Tableau up to 100x faster. Tableau's Data Engine fully utilizes your entire system to achieve fast query response on hundreds of millions of rows of data on commodity hardware. Because the Data Engine can access disk storage as well as RAM and cache memory, it is not limited by the amount of memory on a system. There is no requirement that an entire data set be loaded into memory to achieve these performance goals.of technologies being used to handle various characteristics, such as high volume, data location, and a variety of data source types. Every organization is truly unique, and you should take the time to prioritize which components are the most applicable to you today and in the future.

Tableau Server Components

The work of Tableau Server is handled with the following four server processes:

Application Server: Application Server processes (wgserver.exe) handle content browsing, server administration and permissions for the Tableau Server web and mobile interfaces. When a user opens a view in a client device, that user starts a session (workgroup_session_id) on Tableau Server. The default timeout for this session is easily configuration by an administrator. You can run two or more application server processes to meet your scalability and availability needs.

VizQL Server: Once the user is authenticated via the application server, the user can open a view. The client sends a request to the VizQL process (vizqlserver.exe). The VizQL process then sends queries directly to the data source, returning a result set that is rendered as images and presented to the user. In many cases, Tableau Server leverages client side rendering and caching to reduce the load on the server. Also, each VizQL Server has its own cache that can be shared across multiple users in a secure way. You can run two or more VizQL Server process to meet your scalability and availability needs.

Data Server: Unlike traditional approaches to meta data management, the Tableau Data Server is a key component that allows IT administrators to enable monitoring, meta data management and control for IT, while enabling self service analytics for business users. It lets you centrally manage and store Tableau data sources and provides end users with secure access to trusted data in a self service analytics deployment. You can centrally manage metadata, such as connections, drivers, and data source filters for data access. You can assign specific permissions to data source in away that allows IT to manage permissions to the data sources based on specific AD groups. In a managed environment, users that are closest to their data also have the flexibility to define and publish a definitions, calculations and groups. These can be shared and used by everyone in the organization or users using Tableau Desktop to create and provision their own calculations, definitions, and groups.

The published data source can be based on:

- A Tableau Data Engine extract

- A live connection (cubes are not supported as live connections)

Read more about the Data Server in the section Data Strategy below.

Backgrounder: The backgrounder refreshes scheduled extracts, delivers notifications and manages other background tasks. The backgrounder is designed to consume as much CPU as is available so as to finish the background activity as quickly as possible.

Gateway/Load Balancer

The Gateway routes requests to other components. Requests that come in from the client first hit an external load balancer, if one is configured, or the gateway and are routed to the appropriate process. In the absence of an external load balancer, if multiple processes are configured for any component, the Gateway will act as a load balancer and distribute the requests to the processes. In a single-server configuration, all processes sit on the Gateway, or primary server. When running in a distributed environment, one physical machine is designated the primary server and the others are designated as worker servers which can run any number of other processes. Tableau Server always uses only one machine as the primary server.

Clients: Web Browsers and Mobile Apps

Tableau Server provides interactive dashboards to users via zero-footprint HTML5 in a web or mobile browser, or natively via a mobile app. There is no ActiveX, Java, or Flash needed to run reports and vizzes. No plug-ins or helper applications are required. Tableau Server supports:

- Web browsers: Internet Explorer, Firefox, Chrome and Safari
- Mobile Safari: Touch-optimized views are automatically served in mobile Safari
- iPad app: Native iPad application that provides touch-optimized views, content browsing and editing
- Android browser: Touch-optimized views are automatically offered in the Android browser
- Android app: Native Android application that provides touch-optimized views, content browsing, and editing

Clients: Tableau Desktop

Tableau Desktop is the rapid-fire business analytics authoring environment used to create and publish views, reports and dashboards to Tableau Server. Using Tableau Desktop, a report author can connect to multiple data sources, explore relationships, create dashboards, modify metadata, and finally publish a completed workbook or data source to Tableau Server. Tableau Desktop can also open any workbooks published on Tableau Server or connect to any published data sources, whether published as an extract or a live connection.

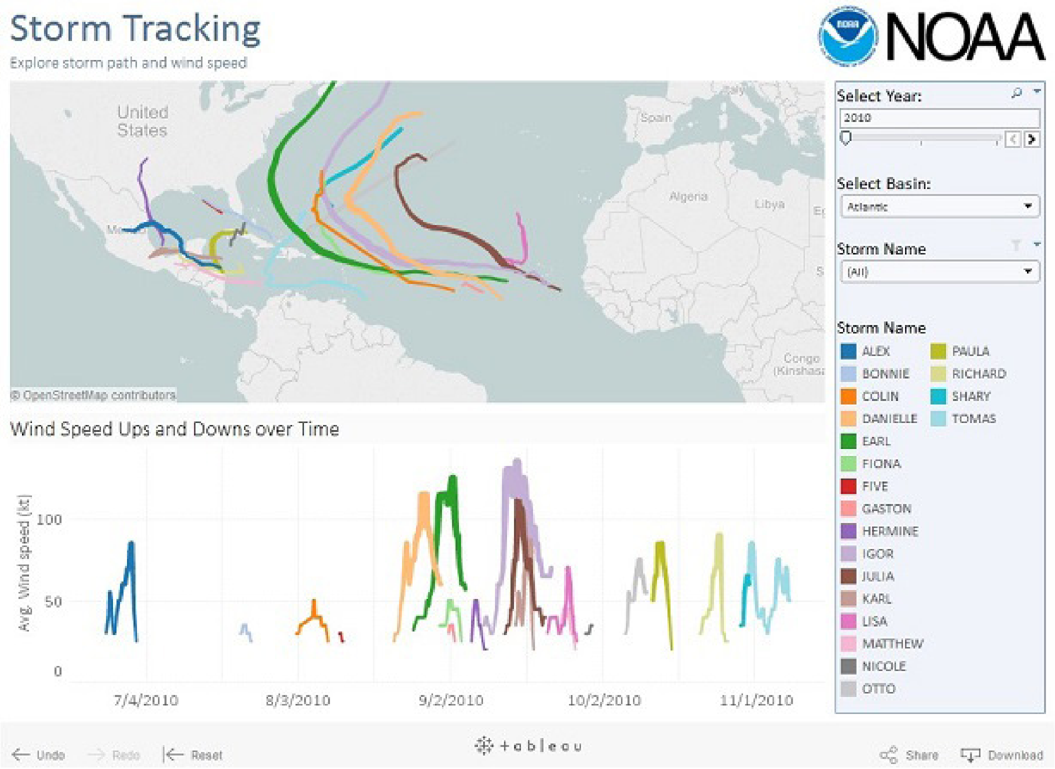
Tableau Desktop runs on both Windows and Mac desktops.

Customization and Extensibility

Tableau supports a robust extensibility framework for deep and complex enterprise integrations. The extensibility span rich tableau visualization integration to enterprise portal applications, bringing any data from any source into a tableau supported format and delivering server automation with a growing set of standards based RESTful APIs.

JavaScript API

With Tableau's JavaScript API you can not just embed, but fully integrate Tableau visualizations into your own web applications. The API uses an event based architecture that provides you with flexibility for round trip control of the users actions and tableau visualizations. It lets you tightly control your users' interactions and combine functionality that otherwise couldn't be combined. For example, your enterprise may have a web portal that bridges several lines of business applications as well as dashboards and reporting. To make it easier for users, you may prefer to have a consistent UI across all applications. With the JavaScript API, you could create buttons and other controls in your preferred style that control elements of a Tableau dashboard.



Storm Tracking

Explore storm path and wind speed

Select Year: 2010

Select Basin: Atlantic

Storm Name: (All)

Figure 3: Example usage of JavaScript API to integrate a Tableau dashboard within your own web application.

Data Extract API

Tableau provides direct support and connection to a large number of data sources. However, there are times when you may want to pre-process or access and assemble data from other applications before working with it in Tableau. With Tableau's Data Extract API, developers can write their own programs to access and process those data sources into a Tableau Data Extract (TDE). The TDE file can be used natively in Tableau Desktop or published to Tableau Server using the same API. Once the TDE has been published to Tableau Server, it is available for an individual to use with the web authoring capability or in Tableau Desktop. The API works with C/C++, Java and Python, in both 32-bit and 64-bit. The Data Extract API is available for developers on Windows and Linux platforms.

REST API

With the Tableau Server REST API you can create, read, update, delete and manage Tableau Server entities programmatically, via HTTP. The API gives you simple access to the functionality behind the data sources, projects, workbooks, site users, and sites on a Tableau server. You can use this access to create your own custom applications or to script interactions with Tableau Server resources.

Data Strategy

Every organization has different requirements and solutions for its data infrastructure. Tableau respects an organization's choice and fits existing data strategy in two key ways: first, Tableau can connect directly to data stores or work in-memory; and second, Tableau works with an ever-increasing number of different data sources.

Access a Variety of Data Sources

At its very simplest, Tableau connects to a single data source with a single view whether the data is in large data warehouses, data marts, or flat files. The view can be a join of multiple tables within that data source, which could be a:

- Relational database– Multiple tables within a single schema can be joined together in relational databases such as SQL Server, Oracle, Teradata, DB2, and Vertica

- Cloud-based business applications—Google Analytics and Salesforce

- Cloud data warehouse – Google BigQuery and Amazon RedShift. Multiple tables can be joined together

- Multi-dimensional (OLAP or cube) database – Technologies such as SQL Server Analysis Services and Essbase.

- Access MDB file – Multiple tables within the Access database can be joined together.

- Excel spreadsheet – Each tab in the spreadsheet is treated as a single table and different tabs can be joined in the same way relational database tables can.

- Flat files – Files using the same delimiter (comma, tab, pipe, etc.) residing in the same Windows folder can be treated as individual tables within a database.

Users have the ability to define joins between tables as long as those joins are supported by the database. If all of the data needed is in a single database management system (DBMS), such as Oracle, SQL Server, or Teradata, the database administrator (DBA) can create a database view pulling data together from various schemas, or the user can create a logical view of the data with custom SQL. Data can be stored in any structure including transactional (3rd, 4th, or 5th normal form), denormalized "flat" forms, and star and snowflake schemas. The Tableau Server and Desktop view performance is directly related to the speed of the underlying structure of the database. While multi-dimensional databases generally perform best, a relational database with a clean star schema or an analyticsoptimized database will perform higher than most other highly-normalized transaction-oriented databases.

Using Extracts for Efficiency and Offline Access

Tableau can connect directly to data or bring data in-memory. If you have invested in fast, analytics-optimized databases, Tableau will connect directly with an optimized connector to let you leverage the value of that investment. If you have a data architecture built on transactional databases or want to keep analytical workloads off your core data infrastructure, Tableau's Data Engine provides an analytics-optimized in-memory data store. Switching between the two is simple.

By default, Tableau provides a "real-time" experience by issuing a new query to the database every time the user changes their analysis. While this can have its own benefits, it can also be an issue if datasets are large or data sources are underperforming or even offline. When data is not constantly changing, real-time queries create an unnecessary workload.

In these cases, Tableau offers an extract capability that brings back data from the initial query and stores it on the user's local machine. The extract is stored in Tableau's columnar database that is highly compressed and structured for rapid retrieval. Extracts can be created from all database types except multidimensional databases (cubes).

Using Tableau Data Extracts can greatly improve the user experience by reducing the time it takes to requery the database. In turn, extracts free up the database server from redundant query traffic. Extracts are a great solution for highly-active transactional systems that cannot afford the resources for date-time queries. The extract can be refreshed nightly and available offline to users during the day. The ability to access data offline can be helpful for users who are traveling or otherwise off of the network.

Extracts can also be subsets of data based on a fixed number of records, a percentage of total records, or a filtering of the data. The Data Engine can even do incremental extracts that update existing extracts with new data. Extract subsets can speed up development time. Developers can use a small subset of data to build a visual application and they won't have to wait for a delayed query response every time they make a change. Extracts are necessary to share packaged workbooks. Tableau's packaged mworkbooks (.twbx file type) contain all the data that was used making it both portable and shareable with other Tableau users. Packaged workbooks can malso be shared using Tableau Reader, which gives users an interactive mexperience but with static data, and without the security measures of Tableau Server.

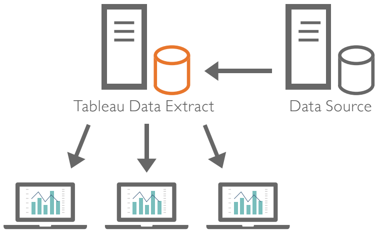
If a user publishes a workbook using an extract, that extract is also published. Future interaction with the workbook will use the extract instead of requesting live data. mIf enabled, the workbook can be set to request an automatic refresh of the extract. Lastly, keep in mind that the amount of temporary disk space used to build an extract can be significant. An example of this would be a star schema with a long fact table mand many dimensions, each with many long descriptive fields.

Data Governance: The Tableau Data Server

Ensuring that the right data is available to the right people in the organization, at the time they need it is important to IT organizations. Often times in spite of stringent IT governance policies, users often go the route of saving critical analysis documents on their desktop for quick analysis or resort to going to the cloud. In a self-service environment, the role of Data Governance is to ensure security is enforced while letting users get the answers they need.

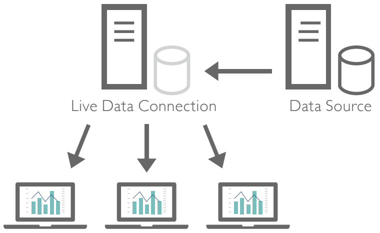
Centralized Extracts

The Tableau Data Server allows for managing Data Extracts, including both data and metadata.



Shared Proxy Connections

The Tableau Data Server can also support live proxy connections.



Figures 4 & 5: Tableau supports centralized management of both live data connections and extracts.

The Tableau Data Server is the component by which Tableau Server provides sharing and centralized management of Tableau Data Extracts and shared proxy database connections. This allows IT to make governed, measured and managed data sources available to all users of Tableau Server without duplicating extracts or even data connections across workbooks. This means the organization has a way to centrally manage:

- Data connections and joins
- Calculated fields (for example, a common definition of "profit").
- Field definitions
- Sets and groups
- User filters

At the same time, to enable self-service and flexibility, users can extend the data model by blending in new data or creating new definitions and allow the newly defined data model to be delivered to production in an agile manner. The centrally managed data will not change, but users retain flexibility.

Published data sources can by of two types:

1. Tableau Data Extracts: Users can connect directly to a published data extract. An organization may choose this approach to provide users with fast, self-service analytics while taking load off critical systems. Centralized Data Extracts also prevent a proliferation of data silos around an organization. Data refreshes need only be scheduled once per published extract and users across the organization will stay up to date with the same shared data and definitions.

2. Shared Proxy Connections: Users can connect directly to live data with a proxy database connection. This means that each user does not have to set up a separate connection, making it easier to get started working with data. The user also does not need to install any database drivers, reducing the load on IT to distribute drivers and keep them up to date.

Report Governance

As data and information continues to grow, information governance becomes critical. It's important that users only have access to information they are authorized to see.

There are two ways to manage report governance in Tableau: Through sites and through projects.

Tableau Server has an out-of-the-box deployment method that provides data isolation and multi-tenancy. A server can have one or more sites, a site can have one or more projects and a project can have one or more workbooks. These projects, and workbooks can be managed and monitored for usage with Tableau. A site is the tenant and Tableau Server ensures data is solated between any two sites.

That is you cannot run queries across sites so there is a strong data isolation boundary between sites. This process creates what is sometimes referred to as a “Chinese Wall” between views.

If you want full data isolation, the best way to accomplish this is to create a site, create a project in the site and manage permissions on the project and workbooks to govern access.

Alternately, if you use only a single site, you can create multiple projects. Projects isolate views and limit individual users to seeing only the views for which they have permission.

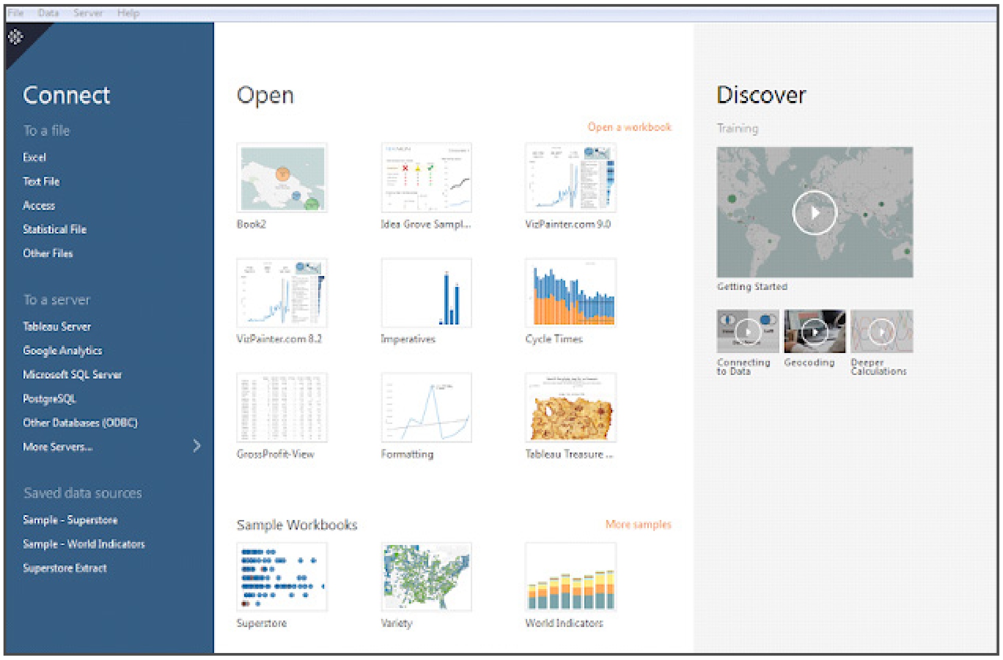
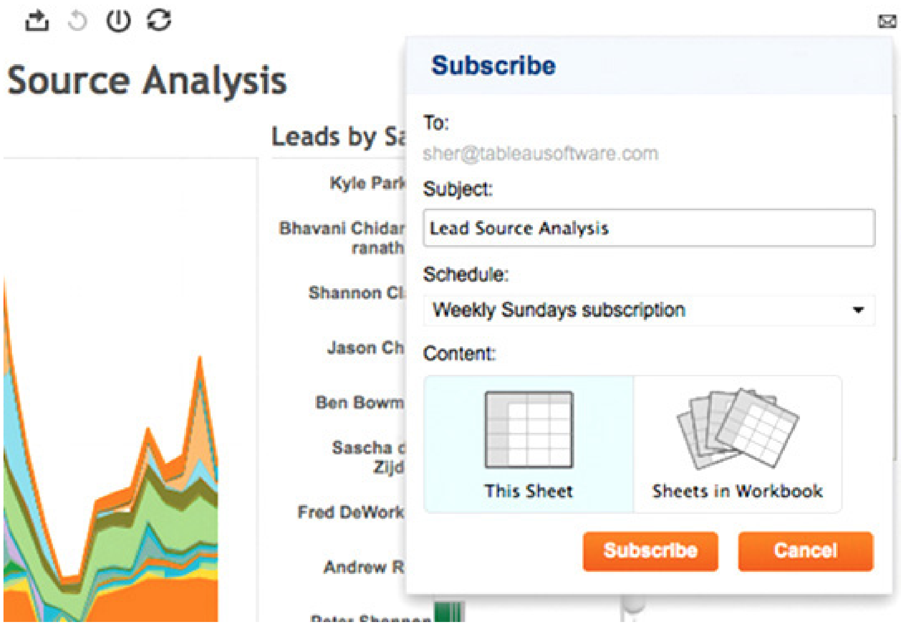


Figure 6: The Tableau content management interface allows easy report governance.

Many organizations choose to create a project for each business unit, like Sales, or for each logical business function, like Finance. Once a project is created, users or user groups can be associated with the project. Users not associated with a project do not see any of its views.

Give Users More Control with Subscriptions

Subscriptions let users subscribe to content they are most interested in and have it automatically delivered to their email inbox on a schedule. Users can subscribe and manage their subscriptions to a worksheet or workbook with a single click.



To:
sher@tableausoftware.com
Subject:
Lead Source Analysis
Schedule:
Weekly Sundays subscription
Content:

Figure 7: Subscription user interface.

Change Management

Tableau offers several options when it comes to performing change management with workbooks. Organizations that have existing change management tools should use them to track changes in Tableau workbooks. Those without these types of tools can set up a manual change management process by either creating user folders on the network or using a backup server and performing a nightly backup that is saved. These can then be restored as needed.

As with any development process, moving work from development to production should follow stringent guidelines that include testing and approvals. One of the best and easiest ways to support the promotion to production is to set up a staging area project that parallels each production project. Staging projects should be set up directly on the production server. Users publish new work to the appropriate staging area and send a follow-up request to the team responsible for validation and promoting to production. When the data source behind the Tableau view used in development is different than production, the connection information will need to be changed.

Metadata Management

Most Business Intelligence platforms say they offer metadata functionality but require modeling the entire enterprise as a first step, or they don't provide metadata capabilities at all. Tableau has taken a hybrid approach so that IT can add value by providing a rich metadata layer – yet business people are empowered to modify and extend it. This means the metadata layer does not require extensive modeling exercises before getting started. Tableau has been so successful in making metadata seamless, approachable and transparent, that customers often don't realize Tableau has a metadata layer.

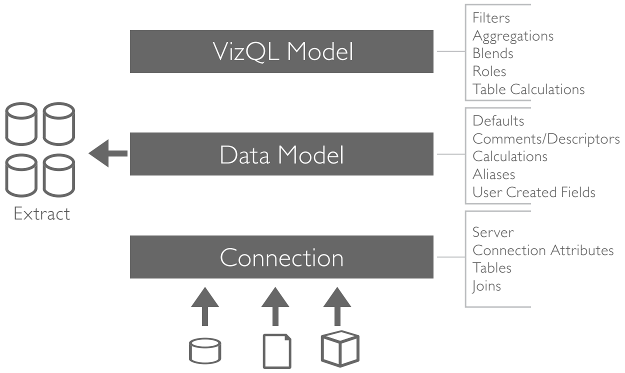


Figure 8: Tableau's flexible metadata management system allows for rich yet flexible metadata.

Tableau's metadata system is a 3-tier system with 2 layers of abstraction and a run time model (VizQL Model). The first layer of abstraction is the Connection which stores information about how to access the data and what data to make available to Tableau. It includes attributes for the database, tables, views, columns, joins, or custom SQL used to access the data.

The second layer is the Data Model which automatically characterizes fields as dimensions and measures. When connecting to cubes, the fields are read directly from the metadata in the cube. In relational data, Tableau uses intelligent heuristics to determine whether a field is a dimension or a measure. The Data Model also keeps track of user-generated fields, such as data sets and calculations. Known together as a Data Source in Tableau, the Data Model is independent and insensitive to changes in the Connection.

The third layer is the VizQL Model, which is unique to Tableau. The VizQL Model lets the user adjust the role and aggregation of the fields at run time. For example, a user can change a measure to a dimension, using employee age in one scenario as a measure to calculate average employee age, and in another scenario as a dimension to view how the workforce is distributed across employee ages. Many calculations and comparisons that are difficult to define in a typical data model are easy to define in the VizQL Model.

Tableau provides additional metadata flexibility. Users can combine data from different Data Sources into one hybrid model without any changes to the Connections or Data Models. Data Models can have dependency on other Data Models. Additionally, a Connection can be used in multiple Data Models, a Data Model can be used in multiple views, and multiple views can be used in a dashboard.

The real value in metadata is the ability to share and reuse components. Data Sources (Connections and Data Models) can be published to Tableau Server creating relationships to workbooks. This means changes in a master data source is automatically propagated to the workbooks that use that data source. Also, other users can use a Data Source as a starting point for their analysis. And, Data Sources can also be exported and shared as files.

Unlike connections, explicit changes to the Data Model must be made in Tableau Desktop due to the broader scope of changes, such as redefining a calculation. When a change is made, Tableau Desktop automatically manages that change across all sheets in a workbook. And, although the VizQL Model is insensitive to changes in the database, such as renaming a member of a column, Tableau Server is sensitive to renaming or removing columns used in a view. If an expected column is missing, the VizQL Model will temporarily remove it from the view.

Mobile Deployment

Enterprise adoption of mobile devices is soaring. Bringing business intelligence into the places where decisions are made and discussions are happening is the real promise of business intelligence. Users have come to expect the same experience on their mobile devices that they have had on their laptops and computers— business intelligence capabilities included.

Tableau delivers mobile business intelligence with the same power and simplicity as the rest of its solution. Tableau's author-once approach to mobile business intelligence means that a dashboard on Tableau Server works automatically on both mobile devices and in a computer's web browser. There is no need for custom dashboard development, as Tableau automatically detects the device and optimizes the visual output and capabilities accordingly. Users will see touch-optimized views on a native iPad app, mobile Safari, mobile Chrome, and Android app.

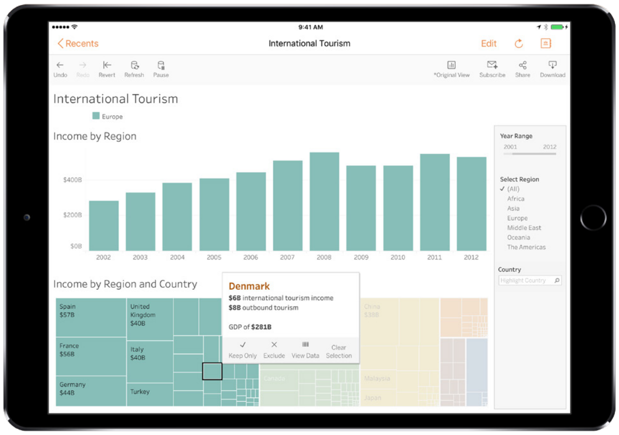


Figure 9: Edit and publish capability from iPad and Android devices.

Tableau mobile apps provide users the ability to view, edit and create new views without using Tableau Desktop. Users can add fields and filters, change view type and do other ad-hoc analysis right in the view. When a user is finished editing, their changes can be saved to the original Tableau workbook, if their permissions allow.

Tableau's mobile apps connect to any Tableau Server and provide native touch controls such as finger scrolling and pinch and zoom. Views are touch-enabled and are optimized for full touch support allowing finger control of filters, parameters, pages, highlighting, and drag a& drop authoring. Users of the Tableau iPad app and Android app also get touch-optimized content browsing, with the ability to search for workbooks, save favorites and see recently used content.

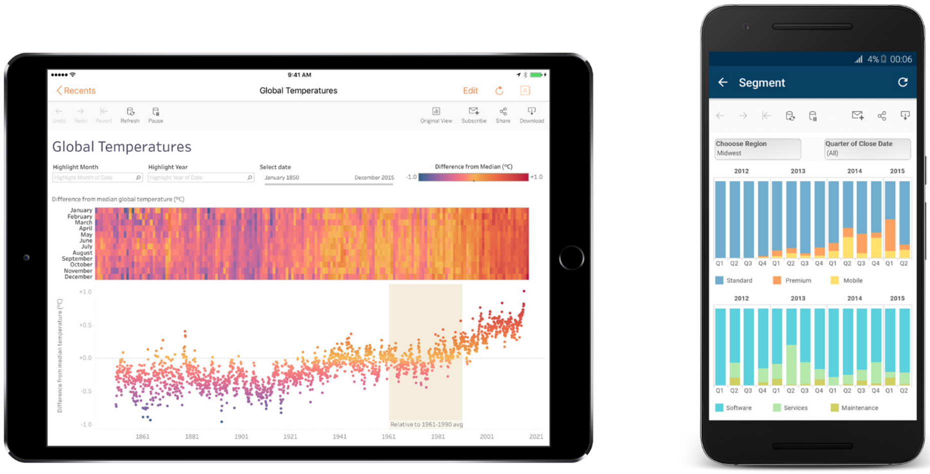


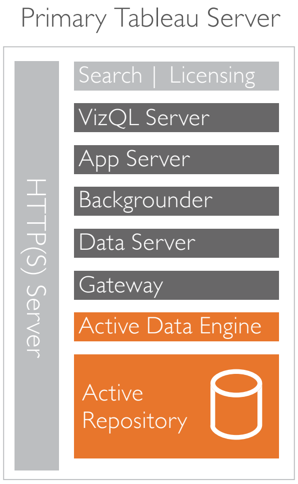
Figure 10: Tableau dashboards are automatically touch-enabled for mobile devices.

Mobile device security is a critical concern for many organizations. Tableau Server enforces the same security for all views, including data-level and user-level security, whether they are served on a desktop or a mobile device. And since no data is stored on the device, there is little risk that lost or stolen devices will compromise data security.

Deployment Models

Tableau can be configured in a variety of ways depending on your data infrastructure, user load and usage profile, device strategy, and goals. Tableau Server can be clustered with any number of machines. Below are examples of common configurations.

Simple Configuration



For many customers, a single server with the recommended minimum hardware configuration—8 CPU cores and 32GB of main memory—will provide good performance. This type of configuration is useful for a proof of concept for a larger deployment or for a departmental server. Tableau recommends running two instances of each major process: Data Server, Application Server, VizQL Server and Backgrounder on a single server 8-core deployment of Tableau Server to allow for better availability.

3-Server (24-Core) Cluster

Environments with heavier user load will require clustering additional servers. In a 3-Server configuration, the primary server will host the Gateway, Licensing and administrative services such as search. The other two worker nodes will have VizQLServer, Application Server, Backgrounder, Repository and Extract Host, Data Server. An administrator can configure the number and type of processing running in the system to support heavy or light extract usage and other characteristics.

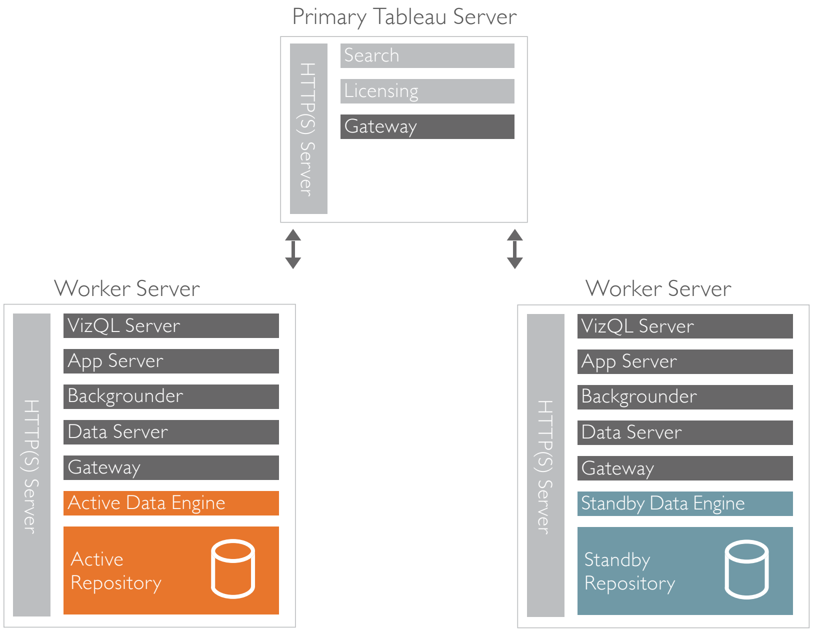


Figure 11: Configuring of a simple 3-server cluster.

5-Server (40-Core) Cluster

More worker machines can be added to a cluster to support heavier data usage or higher user load. In a larger cluster using data extracts, you might choose to isolate the Repository and Extract Host on one machine, the Backgrounders on another, and let the VIzQL and Application Servers reside on the other worker machines. Different configurations are available to support different workload profiles.

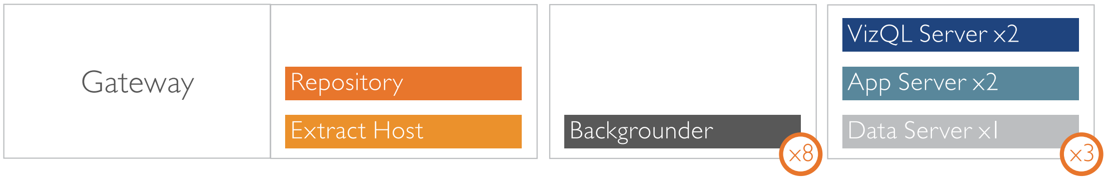


Figure 12: Configuration of a 5-server cluster optimized for many data extracts.

High Availability Cluster

Tableau's High Availability feature helps IT organizations meet SLAs and minimize downtime. Tableau's High Availability solution provides automatic failover capabilities for the Repository and Data Engine components. A primary node serves as the Gateway, search, licensing, and load balancer. The 2 additional nodes host the active processes. You can increase reliability of the system by adding a fourth computer to serve as a backup primary. Gateway failover is automatic when you have an external load balancer configured. This provides a seamless failover capability and increases the workload capacity of the Data Engine, providing higher scalability.

See our High Availability whitepaper for more information.

Virtual Machine or Cloud-Based Deployment

There are no special considerations for scalability or performance when running Tableau Server on virtual machines or in a cloud deployment. However, you should note that each virtualization platform provides sufficient administrative and management infrastructure to deploy the VMs in a scalable, performant topology. You should follow the best practices from your virtual infrastructure provider to ensure Tableau Server is has access to the appropriate compute, memory and data resources. Virtual machines can be used to virtually limit the number of cores available to Tableau Server or to provide disaster recovery via the virtual machine itself. When running Tableau in the cloud, bear in mind that Tableau Server requires static IP addresses.

Security

As organizations make more data accessible to more people, information security becomes a critical concern. Tableau Server provides comprehensive security solutions that balance the variety of sophisticated requirements with easy implementation and use.

Tableau's enterprise-level security features manage authentication, authorization, data security, and network security. Together, these capabilities provide a complete security solution that serves the needs of a broad and diverse user base, whether internal to the organization or external on the Internet. Tableau Server has passed the stringent security requirements of customers in the financial services, government, and healthcare sectors.

Authentication – Access Security

The first level of security is to establish the user's identity. This is done to prevent unauthorized access and to personalize each user's experience. This process is typically referred to as 'authentication'. It should not be confused with 'authorization' which is covered in the next section titled 'Authorization – Object Security'.

Tableau Server supports several types of authentication:

- Microsoft Active Directory (SSPI/NTLM and Kerberos)
- SAML, which uses external identity provider (IdP) to authenticates Tableau Server users.
- Trusted Authentication — a trusted relationship between Tableau Server and one or more web servers
- OAuth for some cloud-based service providers
- Native Authentication managed by Tableau Server

Tableau provides automatic login timeouts that can be configured by administrators.

SAML Authentication

SAML is an standards based authentication protocol mechanism that allows Tableau Server to leverage existing IT investments in Identity Management suites. With SAML, users are authenticated directly with the existing Identity providers, so if the user is already logged in to another enterprise application using the same IdP, they don't have to go through another login process with Tableau Server.

OAuth Authentication

As more and more enterprises are starting to use and leverage cloud based solutions, providing seamless and secure access to cloud resources becomes critical. For some cloud-based data sources, an alternative to storing sensitive database credentials in Tableau Server is to establish a limited-purpose trust relationship between Tableau and the data-source provider. Through this relationship, you can access your data while keeping your credentials secure with the data-source provider.

Tableau works with these protected connections as specified in the OAuth 2.0 openauthorization standard. Using OAuth connections provides the following benefits:

1. Security: Your sensitive database credentials are never known to or stored in Tableau Server, and the access token can be used only by Tableau.

2. Convenience: Instead of having to embed your data source ID and password in multiple places, you can use the token provided for a particular data provider for all published workbooks and data sources that access that data provider. Tableau Server's OAuth authentication is available for Google BigQuery, Google Analytics, and Salesforce.com data sources.

Authorization (roles and permissions) - Object Security

Authorization is what a user can access and do once the user is authenticated.

Authorization is handled by the following:

1. Roles and permissions

2. Licensing and user rights

In Tableau, a role is a set of permissions that is applied to content to manage how users and groups can interact with published content and projects. Published content such as data sources, workbooks, and views can be managed with the typical permission actions of view, create, modify, and delete. Administrators can create groups such as "Finance Users" to make permission management easier.

Projects control the default permissions for all workbooks and views published to the project. The use of projects can be on a single server where support for multiple external parties (multi-tenants) is needed.

Roles provide a default permission structure to differentiate users. For example, a user may be assigned the role of Interactor for a particular view, but not for all content. A user with a Viewer role can see a particular view but does not have the ability to change the view. There are over 20 parameterized customizations available to help manage object security. These role-based permissions do not control what data will appear inside of a view.

Data – Data Security

Data security is becoming increasingly important, especially for organizations needing to meet regulatory requirements or who deliver content externally. Tableau offers flexibility in helping organizations meet their data security needs in three different ways: implement security solely in the database, implement security solely in Tableau, or create a hybrid method where user information in Tableau Server has corresponding data elements in the database.

When a user logs into the Tableau Server, they are not logging into the database.

This means that if you implement security in the database, Tableau Server users will also need to have credentials to log in to the database in order to see views.

These login credentials can be passed using Windows Integrated Security (NT Authentication), embedding the credentials into the view when published, or prompting for specific user credentials.

Tableau also provides a User Filter capability that enables row-level data security with the username, group, or other attributes of the current user.

The filter appends all queries with a 'where' clause to restrict the data and can be used with all data sources.

Network – Transmission Security

For many internal deployments, network security is provided by preventing access to the network as a whole. However, even in these cases it is important to securely transmit credentials across the network. For external deployments, transmission security is critical to protect sensitive data and credentials and to prevent malicious use of Tableau Server. There are three main network interfaces to the Tableau Server, though Tableau pays special attention to the storage and transmissions of passwords at all layers and interfaces.

- The client-to-Tableau Server interface defaults to standard HTTP requests and responses but can be configures for HTTPS (SSL) with customer supplied security certificates.
- Tableau Server-to-database uses native drivers whenever possible and uses generic ODBC adapters when native drivers are not available.
- Secure communication between Tableau Server components is only applicable in distributed deployments and is done using a stringent trust model to ensure each server receives valid requests from other servers in the cluster.

In addition to securing network transmission, all user passwords and credentials are encrypted at rest and in transmission and passwords are not stored in clear text. Each customer can create their own pass phrase to customize their private keys to allow for a secure way to encrypt their user data at rest.

In addition to these network interface security capabilities, Tableau Server provides additional safeguards. There are a variety of encryption techniques to ensure security from browser to server tier to repository and back, even when SSL is not enabled. Tableau Server also has many built-in security mechanisms to help prevent spoofing, hi-jacking, and SQL injection attacks, and actively tests and responds to new threats.

Scalability

Tableau Server is highly scalable, serving the largest enterprises and up to tens of thousands of users. General Motors, Wells Fargo, Bank of America, eBay, Facebook and Cisco are some organizations that have deployed Tableau broadly. Ray White, a large real estate company, uses Tableau to serve reports to 10,000 real estate agents.

Since 2009, Tableau Server has been running at a high scale in Tableau's own data centers to power Tableau Public, a free service for online visualization of public data. Tableau Public has served over 200 million

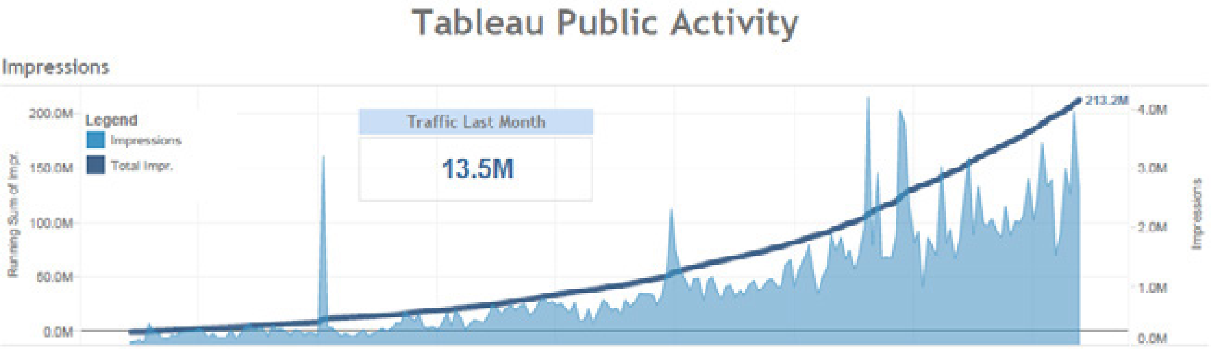


Figure 14: Tableau Server has scaled up to extremely high loads as the infrastructure for Tableau Public.

Every environment is unique and there are many variables that impact performance. Factors that affect the scalability of a Tableau deployment include:

- Hardware considerations: Server type, disk speed, amount of memory, processor speed, and number of processors.
- Architecture: Number of servers, architecture design, network speed/traffic, data source type, and location.
- Usage: user concurrency, interactive, and cache settings.
- Workbook design: Numbers and complexity of views, use of blending and calculations.
- Software configuration: Configuration settings of Tableau Server.
- Data: Data Structures, volumes, aggregation, materialization, and database performance.

Scalability Performance Results

Based on our lifecycle performance methodology and internal benchmark testing, we were able to demonstrate that for a complex workload, Tableau Server scales nearly linearly. As a rule of thumb, sensibly designed reports will allow Tableau server to support 100 concurrent users per 8 cores.

Based on our testing and customer usage estimates that the number of concurrent users on the system is 10%, we demonstrated that Tableau Server scales from 1900 total users on a single node cluster of 16 cores to 5540 total users on a 4-node cluster across 64 cores. This is for a typical workload mix where we see 40% of users interacting.

Cluster Nodes	Concurrent Users	Total Users
1 Node	190	1900
2 Nodes	270	2700
3 Nodes	436	4360
4 Nodes	554	5540

For a more active workload, where 100% of users are interacting with the report, assuming 10% concurrency, Tableau Server can support from 1190 total users with a single primary with 16 cores up to a total of 3470 total users on cluster of primary plus 3-node worker cluster with 64 cores.

Cluster Nodes	Concurrent Users	Total Users
1 Node	119	1190
2 Nodes	206	2060
3 Nodes	269	2690
4 Nodes	347	3470

We periodically perform scalability tests on Tableau Server. Ask your Tableau Account Manager for the most recent scalability test results.

See Further Reading section at the end of this document for links to additional Tableau Server Online Help and Tutorials regarding server and process configurations.

Performance

Performance of a specific report depends on many factors. In a traditional business intelligence report development process, a development team typically spends several weeks to structure the report, optimize the layout, queries and views to deliver a specific performance threshold.

Unlike traditional approaches, where queries are pre-defined and optimized, Tableau's patented VizQL technology allows an exploratory, agile, iterative approach to report generation. Tableau will regenerate optimized queries as needed or re-issue queries on behalf of the user based on what question the user is trying to answer.

Every server environment is unique and there are many variables that impact performance. These variables include:

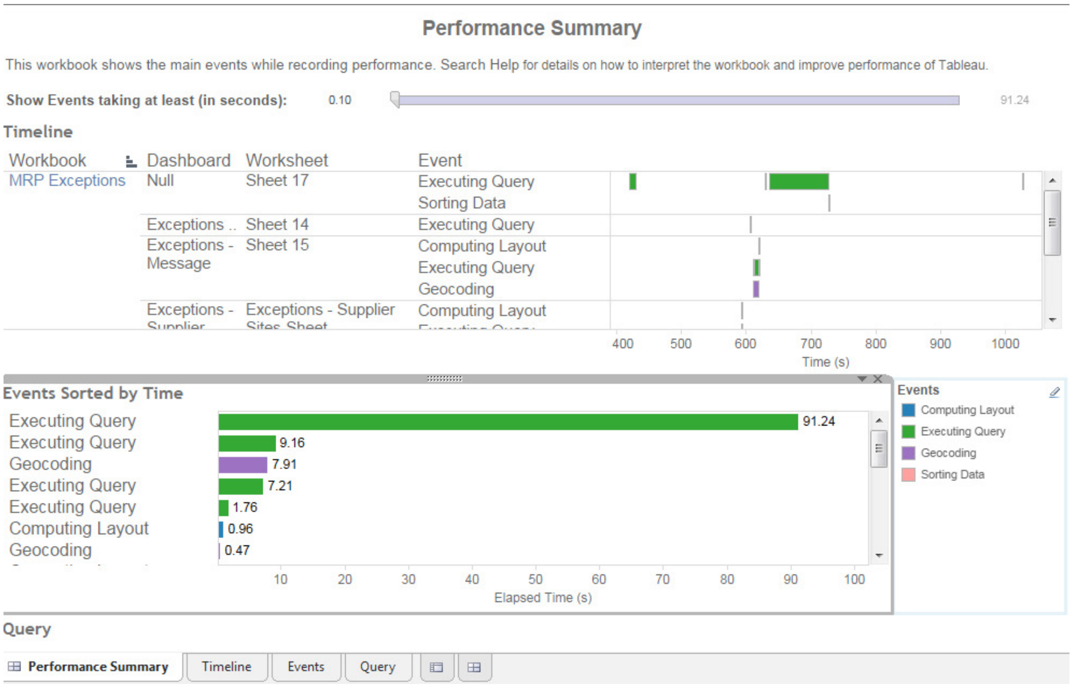
- Hardware details such as disk speed, memory, and cores; the number of servers in your deployment.
- Network traffic
- Usage factors such as workbook complexity, concurrent user activity, mand data caching
- Tableau Server configuration settings, such as how many of each server process you're running Data considerations—such as data volume, database type, and database configuration

64-Bit Tableau Server

For best performance in your large production deployments, deploy a 64-bit operating system and install the 64-bit version of Tableau Server. With 64 bit, Tableau can use a lot more memory than the 32 bit architecture allowed for. While the minimum recommended hardware for an 8 core machine is to use 32 GB or RAM, in practice, if you are able to invest in RAM you can use a general rule of thumb of allocating 8GB per core on a machine to ensure you get good performance.

Self- Service Performance Triage with Performance Recorder

Given the variability in performance, one of the many Tableau Server's administration tools that helps administrators and users is the Tableau Performance Recorder. Diagnosing server or workbook performance issues can be a challenge. Many times slow performing workbooks result from the way a user created joins, the fields they dragged on workbook, data binding in their workbook etc. Tableau's Performance Recorder can profile the performance of a workbook by collecting performance metrics, whether by administrators from the server, or by users on the desktop, web browser, or mobile device.



Performance Summary

This workbook shows the main events while recording performance. Search Help for details on how to interpret the workbook and improve performance of Tableau.

Show Events taking at least (in seconds): 0.10

Timeline

Workbook	Dashboard	Worksheet	Event
MRP Exceptions	Null	Sheet 17	Executing Query
			Sorting Data
	Exceptions ..	Sheet 14	Executing Query
	Exceptions - Message	Sheet 15	Computing Layout
			Executing Query
			Geocoding
	Exceptions - Supplier	Exceptions - Supplier Sites Sheet	Computing Layout

Figure 15: Tableau Performance Recorder Summary.

The Performance Recorder can help you quickly isolate performance problems in mworkbooks, data connections, or queries. For example, the recorder will log the time mit takes to connect to the data source, run a query, or create the visualization. This provides administrators and end users tools that are built into the system to help them isolate issues in workbook design.

Optimized Data Extracts Deliver Faster Response to Users

Tableau's data extracts produce a data file that is small in terms of file size footprint compared to the original data file's size. Tableau utilizes compression techniques mthat optimize file size reduction, especially when data includes text values. This provides an experience of extremely high performance with queries, calculations, filters, and ultimately rendering.

In addition, you can optimize an extract by hiding unused fields in a report which optimizes the extract for faster performance.

Reduced Workload through Secure Shared Sessions

Whenever a report is mainly only read by the users, Tableau securely shares that data across users after ensuring appropriate permissions are met. This provides the users with fast response times to read-only data, minimizes the workload on the server and improves user scaling. This dynamic process utilization offers better on-demand session scaling while still maintaining a secure environment.

Local Rendering Lets Users Interact with Data in Real-Time

Web and mobile authoring leverages the power of HTML5 to render data on the user's own machine or device. There is no ActiveX, Java, or Flash needed to run reports and vizzes. Since the browser now renders the view, there are fewer trips to the server. This reduces the load on the server, making the server more scalable, and ultimately delivering sub-second rendering to users to derive insights.

Dashboards Rendering with Optimized Views

When the views in a dashboard use independent data sources, Tableau loads these views as each query finishes, so the dashboard loads with minimal time. Tableau optimizes the layout and other calculations on the data after the query is returned to allow visualizations to render instantly. Of course, other factors such as query complexity or overall resource contention on the machine may still impact performance.

System Administration

The process of system governance and the role of the Tableau Server administrator are much like that of any other application. However, in Tableau Server, administrators can be assigned to either System or Content administrator roles. System Administrators have complete access to all software and functions within Tableau Server. They can then assign select users to the role of Content Administrator who will manage users, projects, workbooks, and data connections within the group they are assigned to. This allows each group to better manage their own needs.

Key areas the Administrators will be responsible for are:

- Software installation
- Software upgrades
- Monitoring performance, server utilization, and system tuning
- Processes that support security, backup and restore, and change management
- Managing users, groups, projects, workbooks, and data connections

Although Tableau is extremely flexible and can handle tens of thousands of users and more, its server administration tasks are very much part time. In fact, after the initial setup, most organizations find they spend very little time on Tableau Server administration. The time required will likely depend on the number of users, the frequency of user changes, and whether the administrator provides any level of user support.

Conclusion

Tableau Server provides a robust infrastructure that meets the security, scalability, extensibility and architectural requirements of IT managers and administrators. It provides flexible deployment options that scale up to the largest businesses. It supports your data architecture decisions by allowing live connection to a variety of databases or in-memory analytics. And most of all, it lets IT managers get back to strategic IT by getting them out of the dashboard-creation and update cycle. Tableau gives organizations what today's business demands: a self-service, rapid and agile business analytics solution that is truly enterprise-ready.

Further Reading

For more depth on some of the subjects in this paper, please refer to:

Online Help: Tableau Server Administrators' Guide

Online Help: Tableau Server Machine and Process Configuration Guide

Knowledge Base: Monitoring Tableau Server Performance

Knowledge Base: Optimizing Tableau Server Performance

Whitepaper: Rapid-Fire Business Intelligence

Whitepaper: Tableau Server Security, Version 8

Whitepaper: In-Memory or Live Data: Which is Better?

Whitepaper: Tableau Metadata Model

Whitepaper: Tableau Server Scalability Explained

Whitepaper: Tableau Online Security in the Cloud

Gartner positions Tableau as a Leader in 2014 Magic Quadrant

About Tableau

Tableau helps people see and understand their data, no matter how big it is, what channel it's coming from, or what database it's stored in. Quickly connect, blend, and visualize your data with a seamless experience from the PC to the iPad. Create and publish marketing dashboards with automatic data updates, and share real-time insights with colleagues, teams, executive leaders, partners or customers—no programming skills required. Try it for free today!

Additional resources

Download Free Trial

Related Whitepapers

Why Business Analytics in the Cloud?

5 Best Practices for Creating Effective Campaign Dashboards

See All Whitepapers

Explore Other Resources

Product Demo

Training & Tutorials

Community & Support

Customer Stories

Solutions