WHITEPAPER

Alteryx Server Technical Overview

Table of Contents

This whitepaper encompasses a technical overview of Alteryx Server and Alteryx Designer.

Alteryx Designer provides business analysts the ability to create repeatable workflow processes for accessing and blending data and performing advanced analytics (such as predictive, spatial, and statistical). Alteryx Designer includes Designer and Engine components which are described in greater detail below.

Alteryx Server is a scalable server-based analytics solution that helps you create, publish, and share analytic applications, schedule, and automate workflow jobs, create, manage, and share data connections, and control data access. Server is part of the Alteryx AI Platform for Enterprise Analytics, for integrated generative and conversational AI, data preparation, advanced analytics, and automated reporting.

alteryx

Alteryx Designer

Alteryx Designer is a Windows software application that provides an intuitive drag-and-drop user interface for users to create repeatable workflow processes. Users can drag tools from a toolbox onto a canvas, connect them together, and edit their properties to create Alteryx workflows, apps, and macros. Users can use these workflows to blend and enrich data from a range of sources, perform advanced analytics, and quickly produce results that can be easily shared with others.

The unique drag-and-drop workspace within Alteryx Designer allows users to run workflows as they are being developed in order to instantly access and process data. This allows for continuous enhancements and refinements to the workflow and data results. Additionally, users can view data anywhere within a created workflow stream as a table, map, graph, or other visual representation.

Alteryx Designer is written mostly in C# and executes the workflows through a local instance of the Alteryx Engine. Alteryx Designer users may publish their workflows, apps, and macros to the Gallery which is a component of Alteryx Server. Within the Gallery, other users can view, run, and download assets which have been shared with them. Additionally, workflows in the Gallery may be scheduled to execute at predetermined times or specific recurring intervals.

> The unique drag-and-drop workspace within Alteryx Designer allows users to run workflows as they are being developed in order to instantly access and process data.
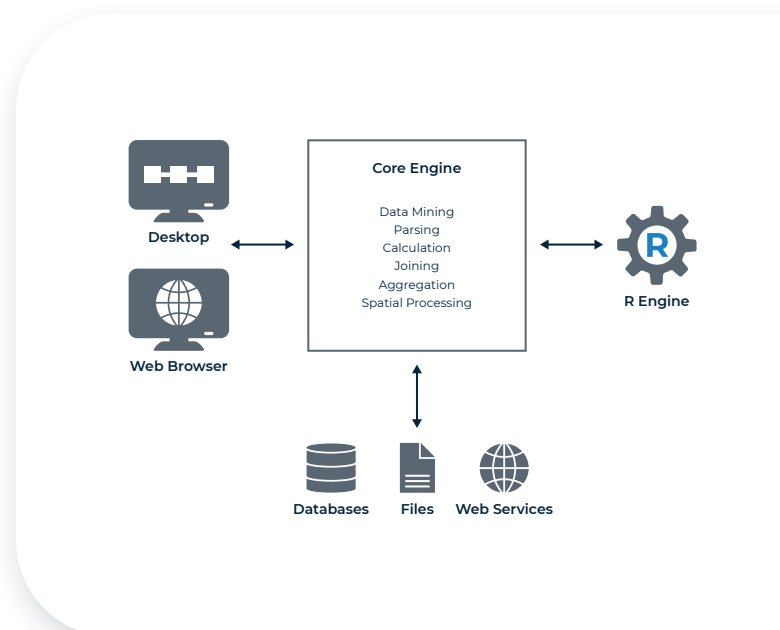
alteryx

Alteryx Engine

The Alteryx Engine, which is written in C++, is responsible for the execution of workflows that are built in Alteryx Designer and produces the output. The execution can occur across multiple threads and cores using Alteryx's Multi-threaded Processing (AMP) Engine. The Engine supports direct connections to various data sources for accessing the data and then processing it during the execution of the workflow. Processing that exceeds memory limitations is written to temp files on disk, which are deleted once the processing is complete. The Engine can be entirely self-contained in an Alteryx Designer deployment, scaled across an organization via the Alteryx Server, or deployed in the cloud.

Depending on the workflow and process, the Alteryx Engine may:

- Read or write to input/output files
- Read or write to one or more databases
- Upload or download data from the web
- Send email through SMTP
- Execute external runtime commands

Desktop

Web Browser

Core Engine
Data Mining
Parsing
Calculation
Joining
Aggregation
Spatial Processing

R Engine

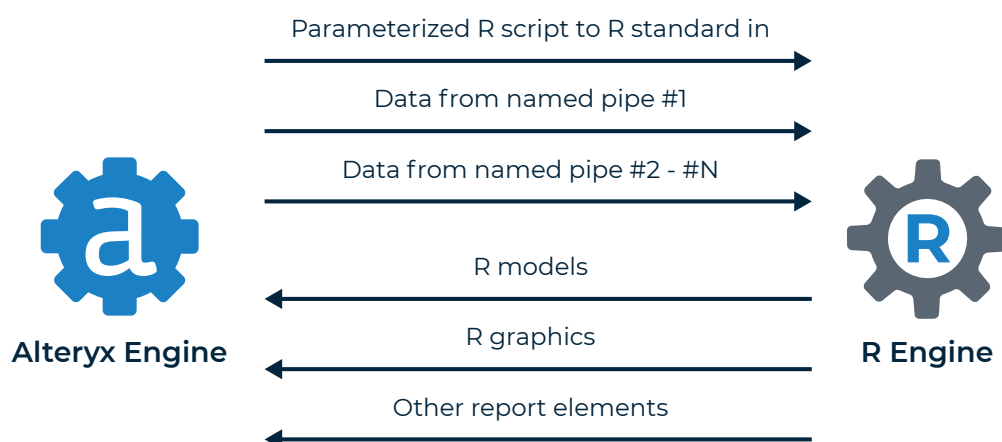Databases   Files   Web Services

alteryx

The Alteryx Engine seamlessly integrates with R. When users install the Predictive Tools package which includes a suite of R-based tools and macros used for predictive analysis, Alteryx also installs the R program and the additional packages that provide connectivity between Alteryx and R and are required for the tools and macros to work. Users can use R scripts or the R-based macros to process data directly within a workflow. When the workflow is run, the Alteryx Engine communicates with the R engine via a command line executable to process and transfer data between the Alteryx Engine and R engine.

Developers can write applications that call directly into the Alteryx Engine using the Application Programming Interfaces (APIs).

The Alteryx Software Development Kit (SDK) gives developers an easy way to build their own tool and add them to the Alteryx toolbox. Both the Alteryx API and SDK are available in an Alteryx Server deployment. For more information, refer to the Appendix.

Whether the Alteryx Engine is running on the desktop, through the API, or via the Alteryx Service, workflows may leverage tools that communicate with databases, input/output files, web services, email servers, R, Python, or run commands.

Alteryx Engine

Parameterized R script to R standard in

Data from named pipe #1

Data from named pipe #2 - #N

R models

R graphics

Other report elements

R Engine

alteryx

Alteryx Service

The Alteryx Service allows the Alteryx Engine to be deployed across multiple servers, providing a highly scalable architecture for the scheduling, management, and execution of Analytic Process Automation workflows.

The Alteryx Service, which is written in C++ with some C# wrappers, can be deployed across multiple servers using a Controller-Worker architecture. This means one server acts as the Controller and manages the job queue, and the others act as the Workers and perform the work. The Service relies upon the Service Persistence layer to store information critical to the functioning of the Service, such as Alteryx application files, the job queue, and result data. The Service also serves content and information to the Gallery when it requests it.

Controller

The Controller acts as an orchestrator of workflow executions and is responsible for the management of the service settings and the delegation of work to the Workers. The Controller retrieves jobs from the persistence layer where it maintains all of the queued jobs and delegate jobs to the Workers. If Alteryx is deployed across multiple servers, only one machine may be actively enabled as a Controller.

The Scheduler is a feature of the Controller that provides users the ability to schedule the execution of workflows, either on a recurring basis or at a specific date, time, and frequency in the future. Users can schedule analytic workflows and manage their scheduled jobs in the Gallery.

The Alteryx Service allows the Alteryx Engine to be deployed across multiple servers, providing a highly scalable architecture for the scheduling, management, and execution of Analytic Process Automation workflows.

alteryx

The Scheduler interface is accessed via the Gallery in an Alteryx Server deployment. Users can only schedule workflows that they have access to in the Gallery. Schedules can be created to require user credentials in order to run. Only administrators (with the Curator role) can view and manage all scheduled jobs from the Gallery Admin interface. The Gallery interface also includes a Schedule Forecast view which offers the ability to see all schedules in a forward-looking calendar view and identify potential upcoming bottlenecks of scheduled workflows.

Worker

The Worker manages one or more instances of the Engine and is responsible for the execution of Analytic Process Automation workflows. Once the Controller delegates a job to the Worker, the Worker executes the job using the Alteryx Engine and produces the output. There must be at least one machine enabled as a Worker in order to execute applications through the Service. You may configure the same machine to be both the Controller and a Worker, or they can be configured on separate machines.

The number of Workers needed will depend upon the capacity requirement of the Alteryx Server environment. If the Worker is not enabled on the same machine as the Controller, then the connection information for the Controller, which includes the Controller's name or IP address and a unique Controller token for that Controller, must be specified for the Worker and Controller to communicate with each other.

JOB TAGS

Job Tag settings can be used to direct workflows to execute on specific Workers. Each Worker can be configured with customized Job Tags. These tags can be specified by users when submitting a job or creating a schedule. The Alteryx Service will route jobs that include a job tag only to Workers with a matching Job Tag. If no Worker with a matching job tag exists, the job will remain in the job queue indefinitely.

QUALITY OF SERVICE (QOS)

The Quality of Service (QoS) setting is used to manage resource allocation in an environment where multiple Workers are deployed by restricting which jobs will be run by each Worker. For normal operation with one machine configured as a Worker, the QoS value should be set to 0. The QoS value for a Worker can be set higher for specific types of jobs to reserve the Worker for higher priority requests.

For example, the default QoS settings in the Alteryx Gallery are:

QoS Setting	Description	Allowable Jobs by QoS Value
0	All jobs	0 - 6
1	Medium priority and higher jobs only	1 - 6
2	High priority and higher jobs only	2 - 6
3	Critical priority and higher jobs only	3 - 6
4	Chained Analytics Applications and higher jobs only (all apps in the chain aside from the last)	4 - 6
6	Workflow validation requests only	6 only

alteryx

Persistence Layer

The Alteryx Service includes a persistence layer that it uses to store information critical to the functioning of the service, such as Alteryx application files, the job queue, and workflow execution results. The Service supports two different mechanisms for persistence layer: SQLite and MongoDB. For lightweight and local deployments which don't require the Gallery, SQLite is adequate for most scheduling needs. For heavier usage, or if the Alteryx Gallery is deployed, MongoDB must be used.

If MongoDB is used, you can elect to either use the embedded MongoDB native to the Alteryx Service or connect the Alteryx Service to your own User-Managed MongoDB instance. Embedded MongoDB support is limited to one server and does not support replica sets. It is highly recommended that you provide an automated backup system for whatever persistence mechanism you choose.

Alteryx Gallery

The Alteryx Gallery is a cloud or self-hosted application for publishing, sharing, and executing workflows. Alteryx offers the Alteryx Analytics Gallery (https://gallery.alteryx.com/) where users can sign up and share workflows, apps, and macros publicly or with selected users. Additionally, an Alteryx Server deployment provides companies their own private Gallery to support their internal users hosted on their own server infrastructure or in a cloud environment.

alteryx

A private Gallery allows organizations to deploy the same architecture used in the Alteryx Analytics Gallery to their internal users within their own IT server infrastructure, whether on premises or in a cloud environment. This provides a greater level of control over the system as well as allowing internal data to remain inside an organization's firewall. By default, the Gallery will be configured to run on a single server, but it can be configured to work across multiple servers behind a load balancer. Each layer of the Gallery architecture is independently scalable, allowing a Gallery deployment to be customized to each individual organization's needs. In a Gallery, administrators can configure and enable or disable features such as workflow scheduling and credentials.

The Gallery user interface, which is written in JavaScript, HTML5, and CSS3, communicates with the Gallery server component via HTTP requests using a REST API. The Gallery web server is written in C# and WCF (Windows Communication Foundation) and handles all the backend logic for what is displayed within the Gallery. WCF receives the HTTP requests and sends them to the server to handle the logic. The server communicates directly with MongoDB for handling its persistence, including information such as the users who have registered with the system, the collections that exist, and the workflows that are in those collections.

The Gallery also communicates with the Alteryx Service for the management and execution of workflows. When a user requests a workflow to be executed, the Gallery communicates with the service layer to execute that workflow and then retrieve the output.

Alteryx supports a REST API for the Gallery which developers can use to interact with the Gallery when creating custom interfaces for running apps. For more information refer to the Appendix.

alteryx

Security

An Alteryx Server deployment that includes a Gallery uses three main communication protocols:

- Communication between a web browser and the Gallery uses standard HTTP (via port 80). The Gallery supports TLS/SSL encryption and can be configured to use HTTPS (via port 443).

- Internal communication between Worker, Gallery, and Controller nodes occurs over HTTP and is encrypted using AES-256 encryption. Each request is encrypted using the Controller token as a shared secret, as well as including additional salt and time-based information, to ensure a secure, time-sensitive payload.

- Communication between the Service Layer components and the Embedded MongoDB persistence layer occurs over TCP/IP (via port 27018). For embedded MongoDB, the host, username, and password are automatically generated, and the password is encrypted as well. For User-Managed MongoDB deployments, communications occur over TCP/IP (via port 27017) and SSL communication can be configured between the Service Layer and MongoDB.
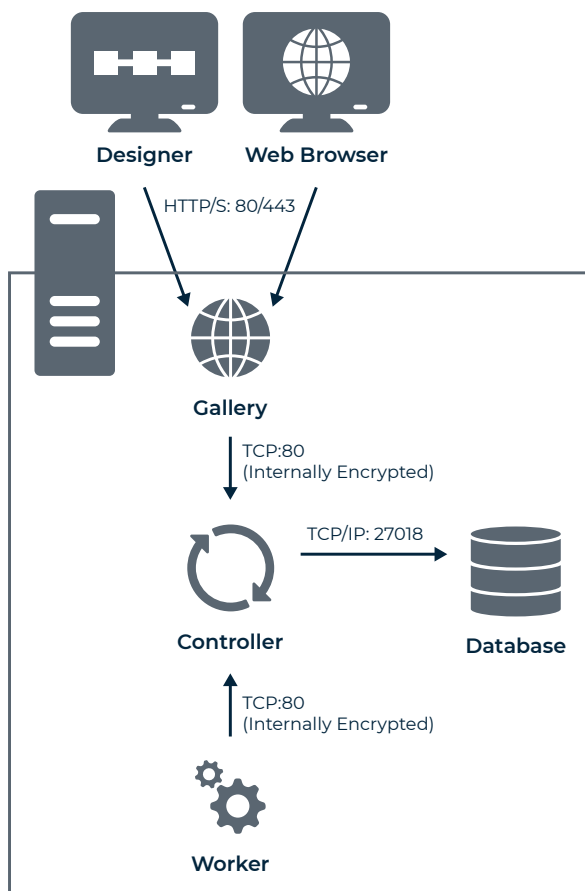
alteryx

Alteryx Server supports three types of authentication to establish a user's identity and prevent unauthorized access to the Gallery: Built-in Authentication, Windows Authentication, and SAML Authentication. If you are accessing the Gallery via the API, you can authenticate with OAuth 1.0A.

- Built-in Authentication requires an email address and password for each user to sign in. User passwords are never sent or stored in clear text. Passwords are hashed, a salt is added, and hashed again. The HMAC-SHA256 hashing function is provided by BCrypt.

- Integrated Windows Authentication allows users to access the Gallery via their network or Windows user credentials. Instead of storing the password and network credentials of users, Alteryx refers to them via an SID within the Active Directory. Integrated Windows Authentication can be configured to utilize NTLM or Kerberos authentication protocols.

- SAML Authentication allows users to access the Gallery via Single Sign On using a third-party Identity Provider (IDP).

- Alteryx also supports a Gallery REST API which uses OAuth 1.0A via a key and secret for authentication. The key and secret are used only for accessing the API and not for logging in to the Gallery. API keys and secrets can be encrypted in transit with TLS/SSL (HTTPS).

- With Alteryx Server, authorization for what a user can access and do once authenticated is managed by user roles and permissions.

- Users can be assigned roles that govern what they can and cannot do within the Gallery, such as publish, run, and share workflows, or administer the site itself.

- In addition to User Roles, fine-grained permissions are available to control users' ability to perform specific actions such as scheduling jobs, creating collections, or API access.

- Access to data assets is governed by the database or network permissions that are set up by a company's IT department.

- For information on how Alteryx integrates with other systems via Email and Run Command tools and events see the Appendix.

alteryx

Deployment

The Alteryx APA Platform is based on two products — Alteryx Designer and Alteryx Server. Alteryx Designer is a desktop client application which provides business analysts the ability to create repeatable workflow processes for accessing and blending data and performing advanced analytics (such as predictive, spatial, and statistical). Alteryx Designer includes the Designer and Engine components.

Alteryx Server enhances Designer by providing a comprehensive and scalable server-based analytics solution that provides end users and business decision makers the ability to share and run analytic applications in a web-based environment. It also gives users the ability to schedule their workflows to run at specific times.

Deploying Alteryx Designer

A deployment of Alteryx Designer is typically done on a desktop and all data is processed in real time on that machine.

Alteryx recommends a minimum of 500 GB of space when installing Designer on a single desktop to account for the operating system, application, and data installation. The Engine is designed to make use of the resources available on the machine on which it is deployed, and its performance is dependent upon the hardware specifications. The Engine consumes more memory, processor, and disk space when workflows and apps are executed.

The Engine may run workflows that use tools that communicate with databases, input/output files, web services, email servers, or run commands. The Engine performance is dependent on the ability to access data and bring it into the workflow, as well as the amount of temp space available for storing files.

alteryx

Alteryx provides a wide assortment of datasets for analysts who integrate data into their analysis. Depending on an analyst's needs, an install of the data may use up to 200 GB of space on a machine. For information about available datasets refer to the Alteryx website.

Deploying Alteryx Server

An Alteryx Server deployment provides for the ad-hoc execution, scheduling and sharing of workflow processes that have been built via Designer. Once Alteryx Server is installed it is scalable to accommodate for higher levels of usage by enabling certain components on additional machines.

With the Alteryx Service (which includes the Controller, Worker, and Persistence), jobs can be scheduled to be executed at predetermined times, or at specific recurring frequencies, through the Gallery interface. The Service manages the job queue, how the job processes, and the results and logs associated with the job in a persistence layer that can be either SQLite or MongoDB. Both SQLite and MongoDB are included in the Server installation, or an existing User-Managed MongoDB instance may be used.

The Alteryx Gallery component is a web-based application that allows users to publish, share, and execute workflows and apps in a browser, as well as providing an interface for administrators to manage the environment. The Gallery web server communicates with the Service for the management and execution of the workflows and a MongoDB persistence layer for maintaining information about the users who have registered with the system, the collections that have been created, and the workflows and apps that are in those collections. When deploying Alteryx Server, companies can take advantage of all the functionality provided in the Alteryx Analytics Gallery while offering a private Gallery to internal users hosted on their own server infrastructure.

When deploying Alteryx Server, companies can take advantage of all the functionality provided in the Alteryx Analytics Gallery while offering a private Gallery to internal users hosted on their own server infrastructure.

alteryx

The Alteryx Service is composed of several different processes that may be running on the machine to handle various aspects of the Alteryx Server functionality such as the Gallery web interface, execution and scheduling of the workflows, and the database persistence layer. To ensure system stability, the Service continually monitors these processes and if a process is not running, the Alteryx Server will try to start it back up. If it is unsuccessful after multiple attempts, the Alteryx Server will stop all of the other processes, shut down, and log the problem.

Scaling Out Alteryx

The complete Alteryx Server can be enabled on a single machine, selected components can be enabled on a machine, or certain components can be enabled on multiple machines to accommodate for higher levels of usage.

A typical deployment of Alteryx Server includes configuring the Controller, Worker, Persistence Layer, and Gallery components, on a single machine. There can only be one active instance of the Controller, but the Worker, Gallery, and Persistence Layer components can be enabled on individual or multiple machines as the need to handle different workloads increases. Decisions around which Alteryx components to scale out are based on the type of work that is needed and who is doing it.
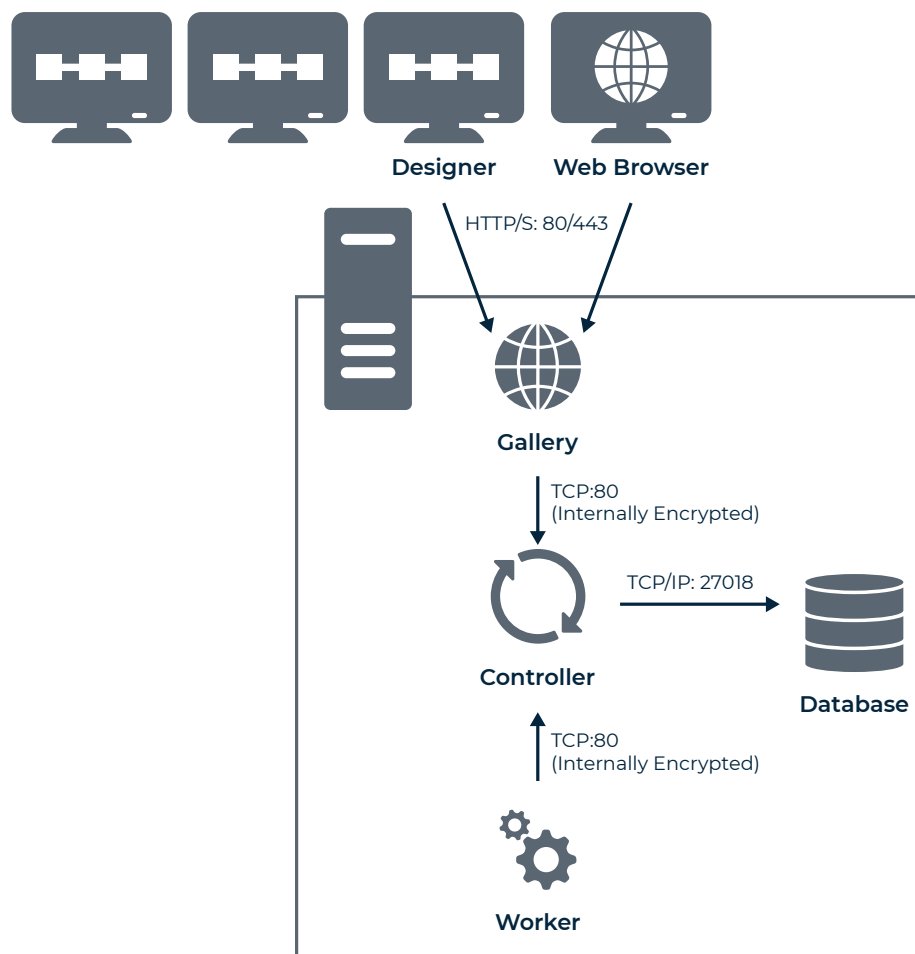
alteryx

Handling Increases in Workflow Execution Requests

If more work needs to be processed and there is not enough capacity to handle it, the Engine can be scaled by configuring multiple machines to act as Workers. Each additional Worker machine must be configured with the unique Controller token of the Controller to be able to communicate with it. Then, as workflow execution requests begin to queue up, each Worker machine will communicate with the Controller and the Controller will delegate jobs for it to process.
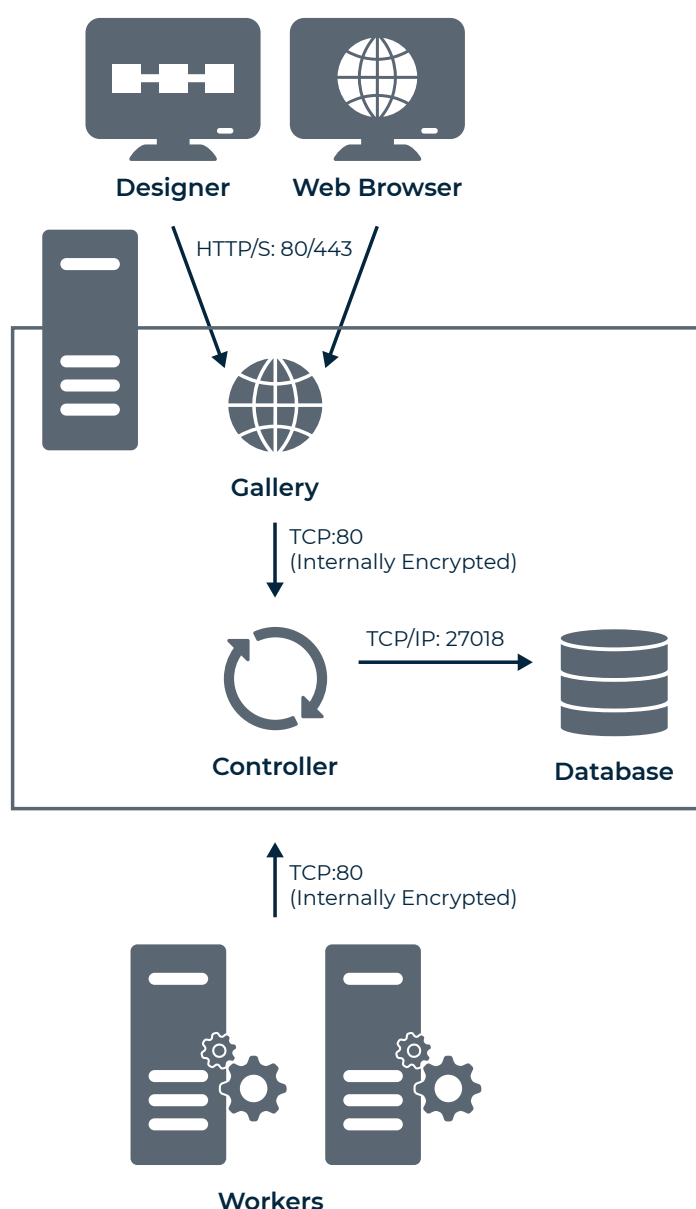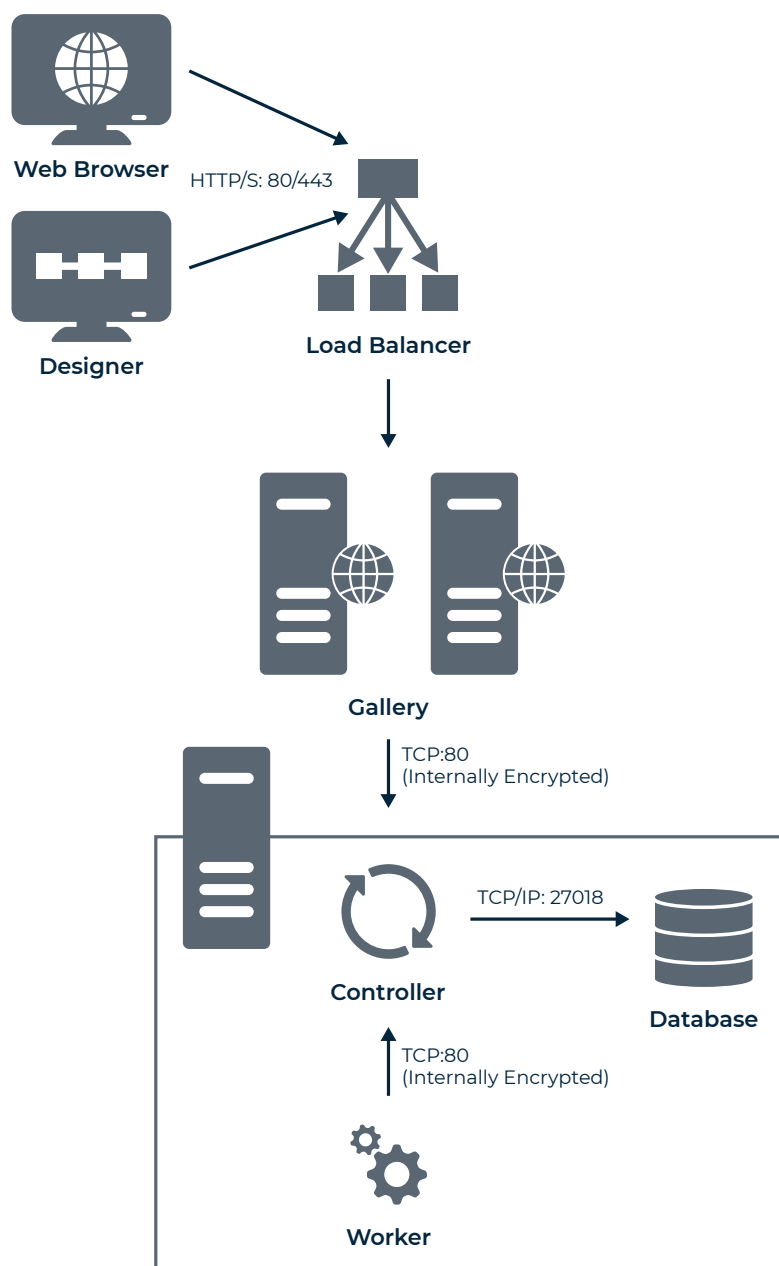
alteryx

Allowing More Users to Create Workflows

If there are multiple end users creating workflows in Alteryx Designer, each Designer instance can be configured to communicate with the same Gallery allowing workflows to be published and executed in a more robust Server environment. The Worker will execute the job and provide the output which users can retrieve from the Gallery interface.

Designer   **Web Browser**

HTTP/S: 80/443

Gallery

TCP:80
(Internally Encrypted)

TCP/IP: 27018

Controller

Database

TCP:80
(Internally Encrypted)

Worker

alteryx

Handling Increased Load on the Gallery

If web requests increase because there are many users who need to access or share the workflows and apps published to the Gallery, the Gallery web server can be scaled out by configuring additional machines to act as web server nodes. When setting up multiple web server nodes these nodes would generally be behind a load balancer to help distribute the amount of web requests across all the web servers. Each individual web server communicates with the Controller when there is work to be processed.

alteryx

Data Backup and Reliability

If an Alteryx Server deployment does not require the Gallery for visual interaction, SQLite can be used as the persistence layer. However, use of the Gallery is highly encouraged and doing so requires a MongoDB persistence layer. Alteryx includes an embedded version of MongoDB for ease of setup and use but the embedded MongoDB is limited to one server and does not support replica sets. If you deploy and manage your own instance of MongoDB (User-Managed), you can follow the MongoDB recommendations for scalability and redundancy.

alteryx
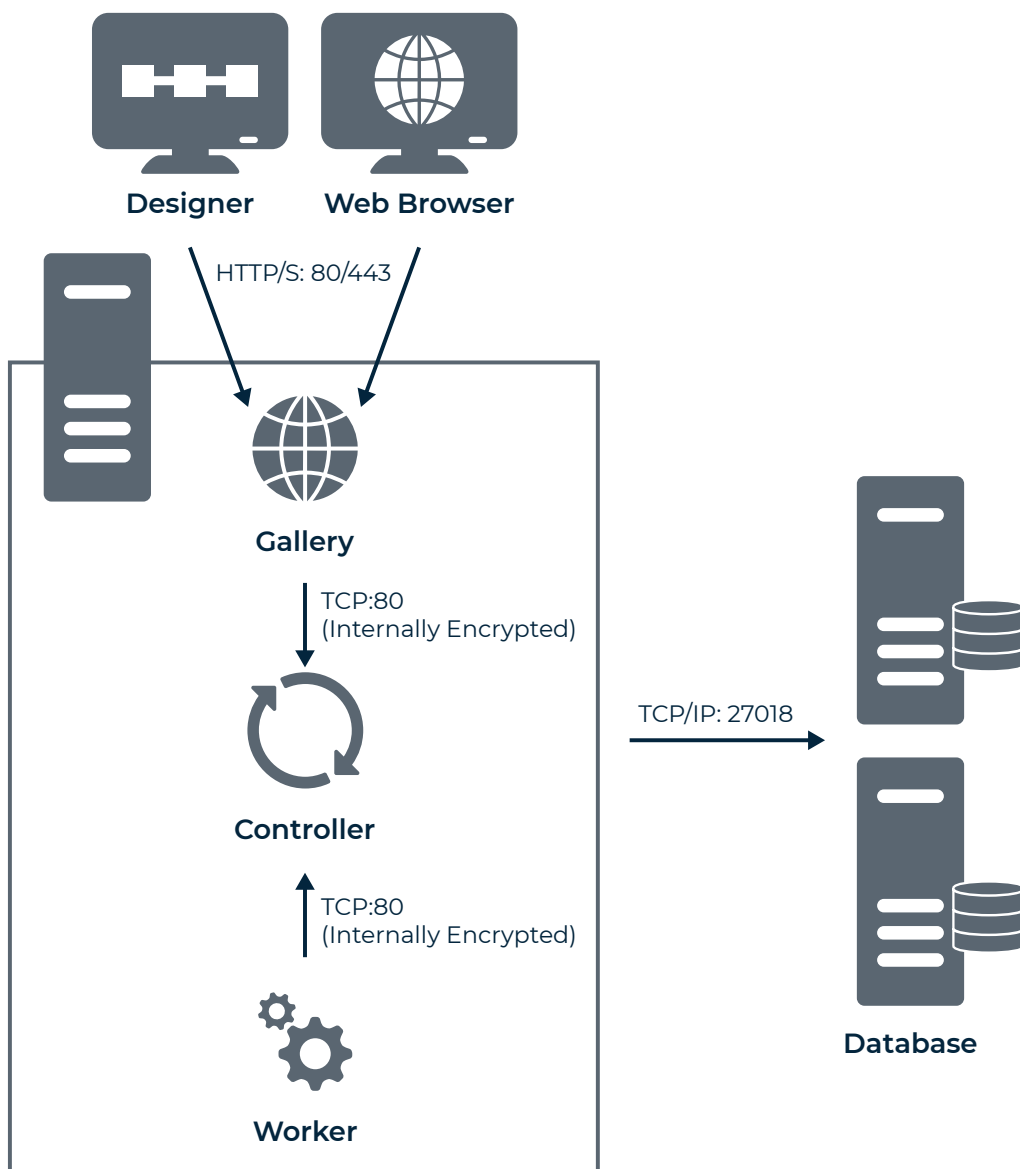
Performance and Scaling Considerations

Multiple factors should be considered when making decisions for a Server deployment. Machine specifications, number of users and their needs, types of workflows being built, automation required for jobs, and volume of data to process are all things that will impact the type of hardware and the configuration needed for optimal performance. In general, optimizing hardware improves processing speed and adding more hardware increases processing capabilities for multiple concurrent users.

A typical deployment of Alteryx Server includes configuring the Controller, Worker, and Galley web server component on a single machine. There can only be one instance of the Controller, but the Worker, Persistence Layer, and Gallery components can be enabled on individual and multiple machines as the need to handle different workloads increases.

The following table provides some guidelines for determining when more hardware for Alteryx Server components may need to be added. A load balancer may be needed to increase capacity and distribute web requests when additional Gallery nodes are added.

	Worker	Gallery	Database
Redundancy needed	✓	✓	✓
High availability needed	✓	✓	✓
Workflow execution time increases	✓		
Number of simultaneous users increases	✓	✓	
Number of backend jobs increases	✓		
Workflows sit in queue for long periods of time	✓		
Memory or CPU consumption is high on web nodes		✓	

Optimizing hardware improves processing speed and adding more hardware increases processing capabilities for multiple concurrent users.

alteryx

An Alteryx Server deployment often involves configuring multiple machines to act as Workers. The following image provides guidelines for determining the recommended minimum number of Workers that may be needed, based on the total number of simultaneous users and the average amount of time it takes for a workflow to execute. These numbers are general guidelines and may not be a fit for every scenario. Work with your Alteryx representative to conduct a proper sizing to ensure a good user experience.

Average Workflow Execution Time

Number of Simultaneous Users	5 secs	30 secs	1 min	2 mins
1-20	1	1	2	3
20-40	1	2	3	4
40-100	2	3	4	5
100+	3	4	5	6

Number of Workers

In some cases, performance and processing speed can be improved by optimizing the hardware.

In some cases, performance and processing speed can be improved by optimizing the hardware.

- Using a Solid State Drive over a Hard Disk Drive could improve processing speeds for workflows involving CASS, Calgary, and geocoding functions.

- Increasing the CPU speed may result in faster processing and lower latency.

- Adding more memory, based on the amount of data to be processed, may improve performance.

- Increasing the number of cores could improve processing speeds for workflows involving drive time and CASS.

alteryx

Data Connectivity and Sources

Alteryx provides data connectivity options to enable users to access and integrate data from many different sources. The various types of data sources Alteryx supports include relational databases, document stores, flat files, and web services.

Through partnerships with leading data providers, Alteryx also provides a wide assortment of datasets for analysts who require spatial or syndicated data to be integrated into their analysis. This data can be optionally installed to enrich data analysis.Data is bundled into packages in a way that allows users to easily install all data at once or to install specific datasets within a package individually. The complete data package may be up to 200 GB in size when installed.

For a current list of data connectivity options and information about available datasets refer to the Alteryx website.

Through partnerships with leading data providers, Alteryx provides a wide assortment of datasets for analysts who require spatial or syndicated data to be integrated into their analysis.

alteryx

Appendix

Alteryx SDKs

The Alteryx SDKs allows developers to create their own custom tools for use within Alteryx workflows. Custom tools allow for distribution within your organization and environment while still leveraging the speed and flexibility of Alteryx Designer and Gallery. A custom tool consists of two components:

1. The GUI, shown in Designer's configuration window. The GUI component must be written in HTML5 using the HTML GUI SDK.

2. The Engine, which processes the records passed to and from the Alteryx Engine. The Engine component can be created using the C++ SDK or the Python SDK.

C++ SDK

The C++ SDK allows you to create custom tools using C++. Most of the tools in the Desktop Designer were built with the Alteryx C++ SDK.

Alteryx workflows consist of a series of tools connected together. Each of these tools falls into one of three categories:

- Input tools, which generate data either spontaneously (such as a random number generator) or by reading information from a data source (such as a file or database)

- Output tools, which consume data either displaying it (such as a table or map) or storing it to a data source

- Computation tools, which consume data, process it, and then generate new or additional data

All tools, regardless of their type, are implemented in a nearly identical fashion. Each tool has a user interface implementation, which handles the visual rendering of the tool object as well as its configuration properties dialog. Each tool also has an engine implementation, which performs the actual processing work. These two pieces together form the single tool.

When needed, Alteryx calls into a tool's user interface implementation to provide its Properties view. When the workflow runs in the Alteryx Engine, an instance of each tool's engine implementation is created, and the outputs of upstream tools are connected to the inputs of later tools. Alteryx then invokes each input tool in the workflow. During the run, tools push their output directly to one another, minimizing execution overhead.

User interface implementations can be written in any .NET language. Alteryx uses C# for this purpose, but others have been successful at using VB.NET. Engine implementations are written in any language which can produce DLLs with an unadorned C-style binding. Alteryx uses Visual C++ for its engine implementations, with a few exported C-style functions to provide the binding layer. For more information on the Alteryx SDKs, see the SDK documentation.

alteryx

Alteryx .NET API

Use the .NET API to integrate Alteryx functionality into your own applications and back-end systems. The APIs consist of native C++ DLLs with.NET wrappers that allow you to access various portions of the Alteryx functionality.

Included with the API are two high-level .NET wrappers for running Alteryx workflows.

- AlteryxAPI.Net.dll: This assembly includes classes for running workflows, accessing properties and settings for Alteryx, as well as high-level classes for accessing Allocate, Solocast, and Calgary functionality.

- AlteryxDocument.dll: This assembly provides access to Alteryx workflow properties and settings, including runtime settings for apps and macros.

In order to work with the API, simply add the appropriate assemblies to your .NET project. The Alteryx installation includes documentation with its APIs in order to help get you started. There are also direct APIs for specific functionality (Allocate, Solocast, Calgary, etc.). The Alteryx Engine API Overview has more information.

Alteryx Engine API

The Alteryx Engine API allows programmatic calls to execute using the Alteryx Engine. This provides the ability to build applications that can programmatically execute Alteryx Designer workflows.

In order to work with the API, link your application with the Alteryx.lib library, and then call into the Engine using the methods defined in the AlteryxEngineAPI.h header. The engine checks that Alteryx is licensed whenever it receives a request to execute a workflow.

alteryx

Alteryx Gallery API

The Gallery APIs allow developers to interact with the Gallery when creating custom interfaces for running apps.

The Gallery API gives developers the ability to:

- Retrieve Workflows in a Subscription
- Retrieve questions for a given Analytic App
- Execute a Workflow or Analytic App
- Retrieve jobs associated with a given Workflow
- Retrieve job status and results for a Workflow execution
- Download app package
- Retrieve credentials IDs shared with a user
- Publish input files for use when executing an Analytic App

The Gallery API gives administrators the ability to:

- Retrieve Audit information
- Find Collections and Subscriptions
- Find Workflows and Data Connections
- View job and Schedule information
- Analyze future job runs to forecast periods of congestion

Each resource in the API is a named URI, and are accessed via HTTP, or HTTPS if enabled. Data transport to and from the API is via JSON. Authentication is done via OAuth 1.0a using an API Key and Secret which can be obtained from a user's profile. User API Keys and Secrets can be encrypted in transit with TLS/SSL (HTTPS).

Interactive API documentation to help developers learn elements of the API, along with example code in various programming languages, is available at Alteryx Engine API Overview for your private Gallery and at https://gallery.alteryx.com/api-docs/ for the Alteryx Analytics Gallery. The Gallery API Overview has information for getting started.

The Gallery APIs allow developers to interact with the Gallery when creating custom interfaces for running apps.

alteryx

Integration with Other Systems

Alteryx can be expanded further and integrated into other systems by way of Developer tools.

EMAIL

Using the Email building block or event, Alteryx can prepare outgoing emails and submit the emails through the SMTP protocol to any SMTP-enabled email server. The user can:

- Set the To, From, CC, and BCC fields.
- Create the subject line.
- Create the body of the message.
- Attach documents.

RUN COMMAND

Using the Run Command building block or event, Alteryx allows for the execution of command line functions through the "Run" shell. Common uses for this command include integration through:

- Batch commands to execute external processes.
- Complex file transfers.
- Custom scripts or other add-ons.

DOWNLOAD

Using the Download building block, Alteryx can integrate with applications that support APIs by retrieving data from a specified URL or uploading data to a system via an HTTP Post method. The Download building block can also download or upload data via FTP/SFTP.

PYTHON

Using the Python building block, Alteryx can execute custom Python code in a Jupyter Notebook style editor. The Python building block includes many common data science packages, including:

- matplotlib
- numpy
- pandas
- scikit-learn
- scipy

alteryx

alteryx | AI PLATFORM
FOR ENTERPRISE ANALYTICS

About Alteryx

Alteryx powers actionable insights with the AI Platform for Enterprise Analytics. With Alteryx, organizations can drive smarter, faster decisions with a secure platform deployable in on-prem, hybrid, and cloud environments. More than 8,000 customers globally rely on Alteryx to automate analytics to improve revenue performance, manage costs, and mitigate risks across their organizations.

To learn more, visit www.alteryx.com.