Best Practices for Deploying Alteryx Server on AWS

May 2020

Notices

This document is provided for informational purposes only. It represents AWS's current product offerings and practices as of the date of issue of this document, which are subject to change without notice. Customers are responsible for making their own independent assessment of the information in this document and any use of AWS's products or services, each of which is provided "as is" without warranty of any kind, whether express or implied. This document does not create any warranties, representations, contractual commitments, conditions or assurances from AWS, its affiliates, suppliers, or licensors. The responsibilities and liabilities of AWS to its customers are controlled by AWS agreements, and this document is not part of, nor does it modify, any agreement between AWS and its customers.

Contents

Abstract

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 analytics platform automation (APA), which unifies analytics, data science, and business process automation in one, end-to-end platform.

This whitepaper discusses how to run Alteryx Server on AWS and provides an overview of the AWS services that relate to Alteryx Server. It also includes information on common architecture patterns and deployment of Alteryx Server on AWS. The paper is intended for information technology professionals who are new to Alteryx products and are considering deploying Alteryx Server on AWS.

Introduction

Alteryx Server provides a scalable platform that helps create analytical insights and empowers analysts and business users across your organization to make better data- driven decisions. Alteryx Server provides:

- Data blending
- Predictive analytics
- Interactive visualizations
- An easy-to-use drag-and-drop interface
- Support for a wide variety of data sources
- Data governance and security
- Sharing and collaboration

Alteryx Server is an end-to-end analytics platform for the enterprise, used by thousands of customers around the world. For details on how customers have successfully used Alteryx on AWS, see the Alteryx + AWS Customer Success Stories.

Alteryx Server

Alteryx Server consists of six main components: Designer, Scheduler, Controller, Worker, Database, and Gallery. Alteryx Server has a core-based subscription licensing model.

For pricing, reference Product Pricing.

Each component is discussed in the following sections.

Designer

The Designer is a Windows software application that lets you create repeatable workflow processes. Designer is installed by default on the same instance as the Controller. You can use other installations of the Designer (for example, on your workstation) and connect it to the Controller using the controller token.

Scheduler

The Scheduler lets you schedule the execution of workflows or analytic applications developed within the Designer.

Controller

The Controller orchestrates workflow executions, manages the service settings, and delegates work to the Workers. The Controller also supports the Gallery and handles APIs for remote integration. The Controller has three key parts: **authentication, controller token**, and **database drivers**, which are described as follows.

Authentication

Alteryx Server supports local authentication, Microsoft Active Directory (Microsoft AD) authentication, and SAML 2.0 authentication. For short-term, trial, or proof-of-concept deployments, local authentication is a reasonable option. However, in most deployments, we recommend that you use Microsoft AD or SAML 2.0 to connect your user directory.

> Note: Changing authentication methods requires that you reinstall the Controller.

For deployments of Alteryx Server on AWS where you have chosen Microsoft AD, consider using AWS Directory Services. AWS Directory Services enables Alteryx Server to use a fully managed instance of Microsoft AD in the AWS Cloud. AWS Microsoft AD is built on Microsoft AD and does not require you to synchronize or replicate data from your existing Active Directory to the cloud (although this remains an option for later integration as your deployment evolves over time). For more information on this option, see AWS Directory Service.

Controller Token

The controller token connects the Controller to Workers and Designer clients to schedule and run workflows from other Designer components. The token is automatically generated when you install Alteryx Server. The controller token is unique to your server instance and administrators must safeguard it. You only need to regenerate the token if it is compromised. If you regenerate the token, all the Workers and Gallery components must be updated with the new token.

Drivers

Alteryx Server communicates with numerous supported data sources, including databases such as Amazon Aurora and Amazon Redshift, and object stores such as Amazon Simple Storage Service (Amazon S3). For a complete list of supported sources, see **Data Sources** on the Alteryx Technical Specifications page.

Successfully connecting to most data sources is a simple process because the Controller has a network path to the database and proper credentials to access the database with the appropriate permissions. For help with troubleshooting database connections, see the Alteryx Community and Alteryx Support pages.

Each database requires you to install the appropriate driver. When using Alteryx Server, be sure to configure each required database driver on the server machine with the same version that is used for Designer clients. If a Designer client and the Alteryx Server do not have the same driver, the scheduled workflow may not complete properly.

Worker

The Worker executes workflows or analytic applications sent to the Controller. The same instance that runs the Controller can run the Worker. This setup is common in smaller scale deployments. You can configure separate instances to run as Workers for scaling and performance purposes. You must configure at least one instance as a Worker—the total number of Workers you need is dependent on performance considerations.

Database

The persistence tier stores information that is critical to the functioning of the Controller, such as Alteryx application files, the job queue, gallery information, and result data. Alteryx Server supports two different databases for persistence: MongoDB and SQLite. Most deployments use MongoDB, which can be deployed as an embedded database or as a user-managed database. Consider using MongoDB if you need a scalable or highly-available architecture. Note that most scalable deployments use a user-managed MongoDB database. Consider using SQLite if you do not need to use Gallery and your deployment is limited to scheduling workloads.

Gallery

The Gallery is a web-based application for sharing workflows and outputs. The Gallery can be run on the Alteryx Server machine. Alternatively, multiple Gallery machines can be configured behind an Elastic Load Balancing (ELB) load balancer to handle the Gallery services at scale.

Options for Deploying Alteryx Server on AWS

Requisite Knowledge & Services

To successfully deploy Alteryx Server on AWS, you will need to be knowledgeable with the following services:

- Windows server administration
- SAML or Active Directory knowledge (dependent on authentication choice)
- SMTP / Email configuration
- SSL to enable HTTPS (optional)
- Familiarity with MongoDB

Deployment of Alteryx Server requires the following AWS services:

- IAM
- VPC
- EC2

Optional AWS services (with rationale) include the following:

- AWS Auto Scaling (high availability)
- Route53 (DNS)
- CloudWatch (monitoring)
- CloudTrail (installation audit)
- RDS (sample databases)
- S3 (log export)
- Simple Email Service (Gallery notifications, including password reset emails, if using built-in authentication)
- AWS Certificate Manager (private CA for ALB deployments or IWA)

Alteryx Server is contained as a Microsoft Windows Service. It can run easily on most Microsoft Windows Server operating systems.

> Note: To install Alteryx Server on AWS, you will need an AWS account and an Alteryx Server license key. If you do not have a license key, trial options for Alteryx Server on AWS are available through AWS Marketplace.

You can install the Alteryx Server components into a multi-node cluster to create a scalable enterprise deployment of Alteryx Server:

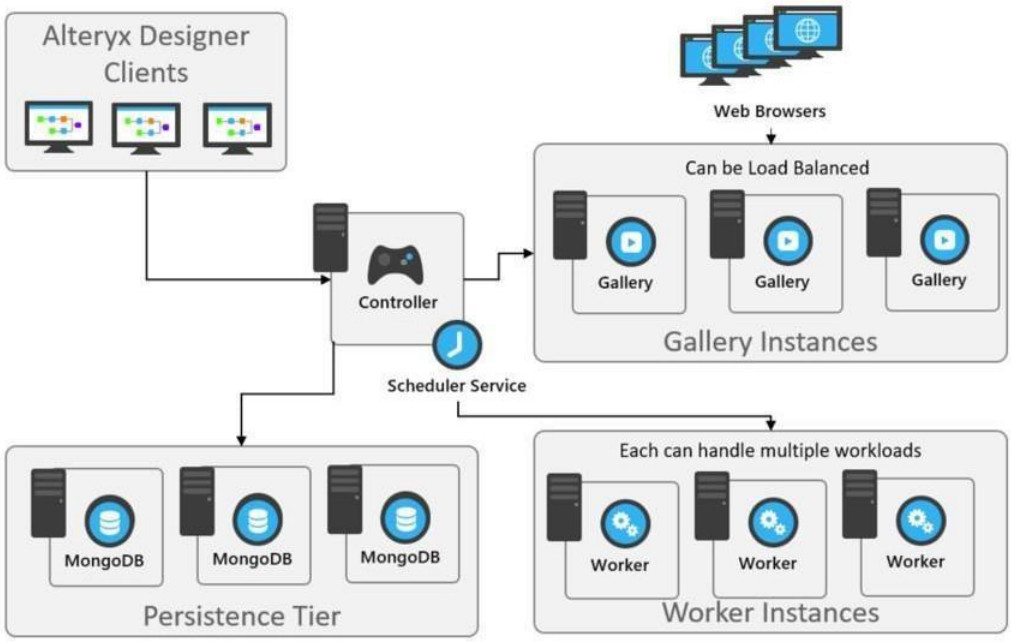

Figure 1: Scalable enterprise deployment of Alteryx Server

Alternatively, you can install Alteryx Server in one self-contained EC2 instance:

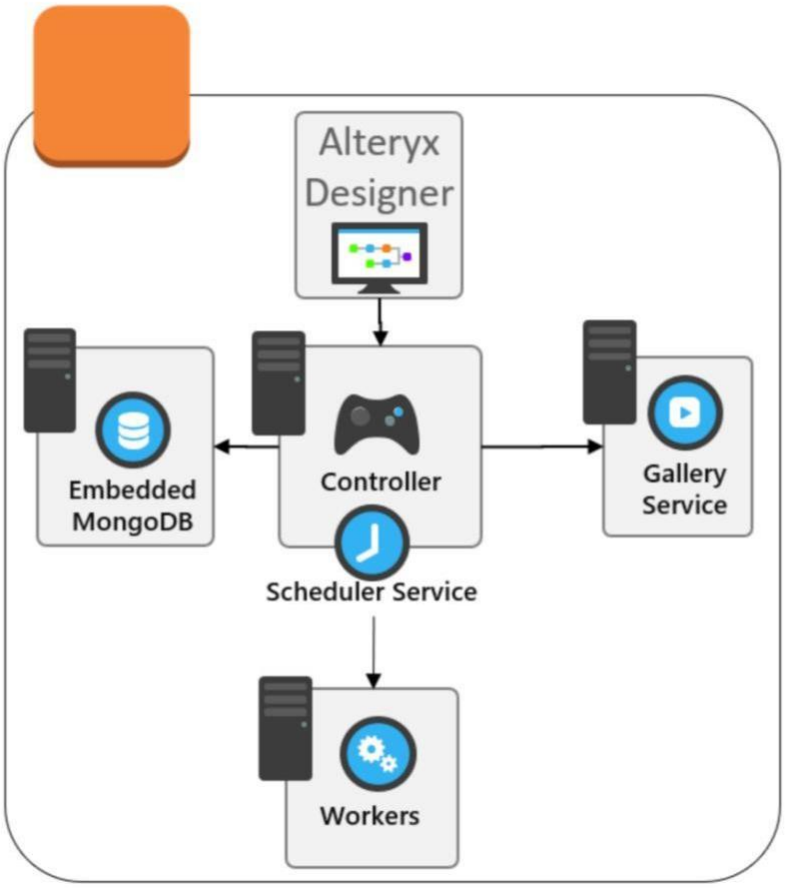

Figure 2: Deployment of Alteryx Server on a single EC2 instance

The following sections discuss how to deploy Alteryx Server on AWS from the most complex deployment to the simplest deployment.

Enterprise Deployment

The following architecture diagram shows a solution for a scalable, enterprise deployment of Alteryx Server on AWS.

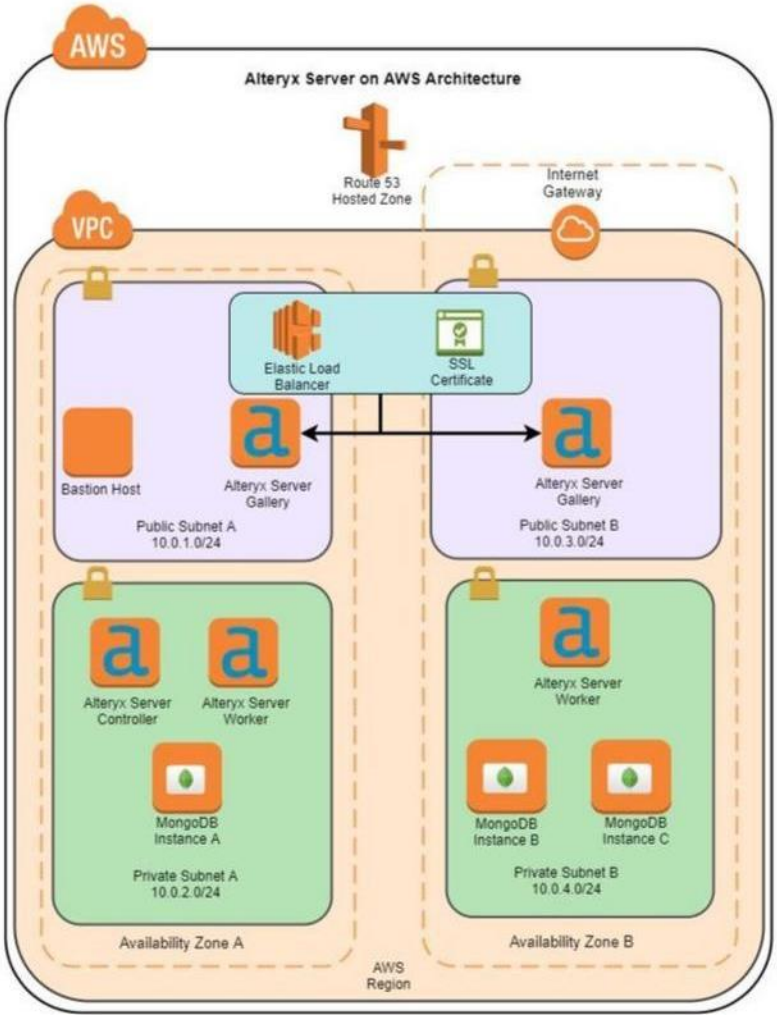

Figure 3: Alteryx Server architecture on AWS

The following high-level steps explain how to create a scalable enterprise deployment of Alteryx Server on AWS:

> Note: To deploy Alteryx Server on AWS, you will need the controller token to connect the Controller to Workers and Designer clients, the IP or DNS information of the Controller for connection and failover if needed, and the user-managed MongoDB connection information.

1. Create an Amazon Virtual Private Cloud (VPC) or use an existing VPC with a minimum of two Availability Zones (called Availability Zone A and Availability Zone B).
2. Deploy a Controller instance in Availability Zone A. Document the controller key and connection information for later steps.

> Note: It is possible to use an Elastic IP address to connect remote clients and users to the Controller, but we recommend that you use AWS Direct Connect or AWS Managed VPN for more complex, long-running deployments. VPC peering connection options and Direct Connect can enable private connectivity to the Controller instance, as well as a predictable, cost-effective network path back to on-premises data sources that you may wish to expose to the Controller.

3. Create a MongoDB replica set with at least three instances. Place each instance in a different Availability Zone. Document the connection information for the next step.
4. Connect the MongoDB cluster to the Controller instance by providing the MongoDB connection information in the Alteryx System Settings on the Controller.
5. Deploy and connect a Worker instance in Availability Zone A to the Controller instance in the Availability Zone A subnet.
6. Deploy and connect a Worker instance in Availability Zone B to the Controller instance in the Availability Zone A subnet.
7. Deploy and connect more Workers as needed to support your desired level of workflow concurrency. You can have more than one Worker in each Availability Zone but be aware that each Availability Zone represents a fault domain. You should also consider the performance implications of losing access to Workers deployed in a Availability Zone.
8. Create an ELB load balancer to handle requests to the Gallery instances.
9. Deploy Gallery instances and register with the ELB load balancer. Be sure to deploy your Gallery instances in multiple Availability Zones.
10. Connect the Gallery instances to the Controller instance.
11. Connect the client Designer installations to the Controller instance using either the Elastic IP address or the optional private IP (chosen in Step 2), then test workflows and publishing to Gallery.
12. (Optional) Deploy a Cold/Warm Standby Controller instance in another Availability Zone or AWS Region. Failover is controlled by changing the Elastic IP address (if deployed in the same VPC) or DNS name to this Controller instance.

Deploy Alteryx Server with Chef

You can use AWS OpsWorks with Chef cookbooks and recipes to deploy Alteryx Server. For Alteryx Chef resources, see cookbook-alteryx-server on GitHub.

Deploy a Windows Server EC2 instance and install Alteryx Server

You can deploy an Amazon Elastic Compute Cloud (Amazon EC2) instance running Windows Server and then install Alteryx Server. You can download the install package here.

Make sure that you deploy an instance with the recommended compute size (at least 8 vCPUs), Windows operating system (Microsoft Windows Server 2008R2 or later), and available Amazon Elastic Block Store (Amazon EBS) storage (1TB).

Deploy an Amazon EC2 Instance from the Alteryx Server AMI

You can purchase an Amazon Machine Image (AMI) from Alteryx through AWS Marketplace and use it to launch an Amazon Elastic Compute Cloud (Amazon EC2) instance running Alteryx Server. You can find the Alteryx Server offering on AWS Marketplace.

> Note: You can try one instance of the product for 14 days. Please remember to turn your instance off once your trial is complete to avoid incurring charges.

You have two options for launching your Amazon EC2 instance. You can launch an instance using the Amazon EC2 launch wizard in the Amazon EC2 console or by selecting the Alteryx Server AMI in the launch wizard. Note that the fastest way to deploy Alteryx Server on AWS is to launch an Amazon EC2 instance using the Marketplace website.

To launch Alteryx Server using the Marketplace website:

1. Navigate to AWS Marketplace.
2. Select **Alteryx Server**, then select **Continue to Subscribe**.
3. Once subscribed, select **Continue to Configuration**.

4. Review the configuration settings, choose a nearby Region, then select **Continue to Launch**.
5. Once you have configured the options on the page as desired, select **Launch**.
6. Go to the Amazon EC2 console to view the startup of the instance.

   It can be helpful to note the Instance ID for later reference. You can give the instance a friendly name to find it more easily and to allow others to know what the instance is for. Click inside the Name field and enter the desired name.
7. Navigate to the instance Public IP address or Public DNS name in your browser. Enter in your email address and take note of the token at the bottom:

   Your token will be specific to your instance. If you selected the Bring Your Own License image, a similar registration will appear and prompt you for license information.

   After selecting your server instance and clicking **Connect**, you will be guided through using Remote Desktop Protocol (RDP) to connect to the Controller instance of Alteryx.

   Once connected, you can use your AWS instance running Alteryx Server. The desktop contains links to the Designer and Server System Settings.
8. Start using Alteryx Server. See Alteryx Community for more information on how to use Alteryx Server and Designer.

Sizing and Scaling Alteryx Server on AWS

When sizing and scaling your Alteryx Server deployment, consider **performance**, **availability**, and **management**.

Performance Considerations

This section covers options and best practices for improving the performance of your Alteryx Server workflows.

> Note: Alteryx Server licensing is based on **physical cores**. AWS EC2 instances are offered with a variety of **vCPU** configurations, where for most instances, two vCPUs are equivalent to one physical core. Functionally, this means that if you have a 4-core Alteryx Server license, you will want to provision an EC2 instance that includes 8 vCPUs.

Scaling Up vs. Scaling Out

You can usually increase performance by scaling your Workers *up* or *out*. To scale *up* you need to relaunch Workers using a larger instance type with more vCPUs or memory, or by configuring faster storage. When scaling up, you should increase the size of all Workers as the Controller does not schedule on specific worker instances by priority and will not assign work to the machine with the most resources. To scale *out* you need to configure additional instances. Both options typically take only a few minutes.

Below are two scenarios that discuss scaling up and scaling out:

Long job queues – If you expect that a high number of jobs will be scheduled, or if you observe that the job queue length exceeds defined limits, then scale out to make sure you have enough instances to meet demand. Scale up if you already have a very large number of small nodes.

Long-running jobs or large workflows – Larger instances, specifically instance types with more RAM, are best suited for long-running workloads. If you find that you have long-running jobs, first examine the query logic, load on the data source, and network path and adjust if necessary. If the jobs are otherwise well tuned, consider scaling up.

This table presents heuristics that can help you determine the number of Workers you need to execute workloads with different run times.

Number of Users	5-Second Workload	30-Second Workload	1-Minute Workload	2+-Minute Workload
Number of Worker Instances				
1-20	1	1	2	3
20-40	1	2	3	4
40-100	2	3	4	5
100	3	4	5	6

Table 1: Number of Worker instances needed to execute workloads with different run times

Consider having your users run some of their frequently requested workflows on a test instance of Alteryx Server of your planned instance size. You can quickly deploy a test instance using the Alteryx Server AMI. These tests will help you understand the number of jobs and workflow sizes that your instance size can handle.

To predict workflow sizes, review your current and planned Designer workflows. In Alteryx benchmark testing, the engine running in Alteryx Designer performed nearly the same as in Alteryx Server when running on similar instance types (see Alteryx Analytics Benchmarking Results). Keep this in mind when determining how long workloads will take to run. You can test workload times without installing Alteryx Server by using the Designer on hardware that is similar to what you would use to deploy Alteryx Server.

Scaling Based on Demand

Many customers find they need to add more Workers at predictable times. For peak usage times, you can launch new Worker instances from the Alteryx Server AMI and pay for them using the pay-as-you-go option. With this model, you pay only for instances you need, for as long as you use them. This is common for seasonal or end-of- month, end-of-quarter, or end-of-year workloads.

You can use an Amazon Elastic Compute Cloud (Amazon EC2) Auto Scaling group with a script to insert the controller token into these new instances to scale additional Worker instances on demand with minimal or no post-launch configuration. Additionally, you can integrate Amazon EC2 Auto Scaling with Amazon CloudWatch to scale automatically based on custom metrics such as the number of jobs queued.

Scaling Alteryx Server to more instances will have licensing implications because it is licensed by cores.

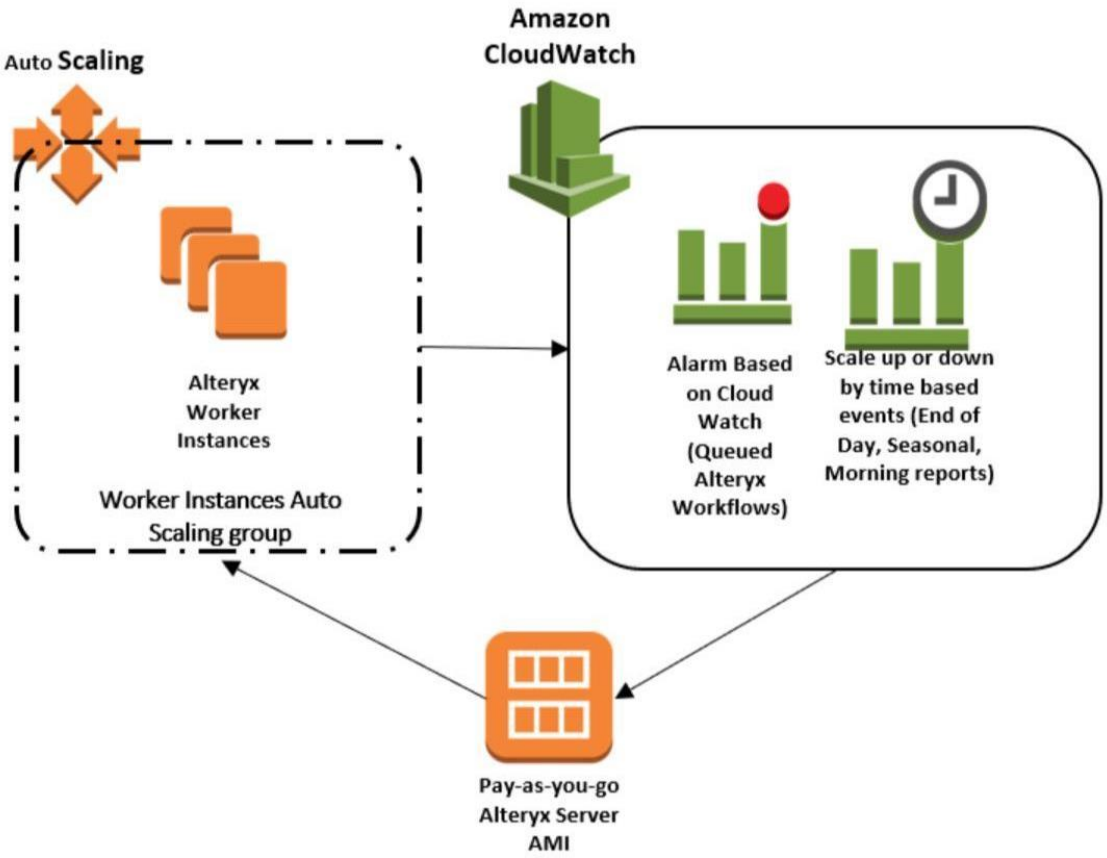

Figure 4: Use Amazon EC2 Auto Scaling and Amazon CloudWatch to scale Worker instances on-demand

You can perform additional scheduled scaling actions with Amazon EC2 Auto Scaling. For example, you can configure an Amazon EC2 Auto Scaling group to spin up instances at the start of business hours and turn them off automatically at the end of the day. This allows Alteryx Server to reduce compute costs while meeting business analytic requirements.

Worker Performance

Workers have several configuration settings. The two settings that are the most important for optimizing workflow performance are **simultaneous workflows** and **max sort/join memory**.

Simultaneous workflows – You have the best starting point for simultaneous workflows when 4 vCPUs are available for each workflow. For example, if an instance has 8 vCPUs, then we recommend that you enable 2 workflows to run simultaneously. This setting is labeled **Workflows allowed to run simultaneously in** the Worker configuration interface. You can adjust this setting as a way to tune performance.

Note: 8 vCPUs = 2 workflows running simultaneously

Max sort/join memory usage – This configuration manages the memory available to workflows that are more RAM-intensive. The best practice is to take the total memory available to the machine and subtract a suggested 4 GB of memory for OS processes. Then, take that number and divide it by the number of simultaneous workflows assigned:

$$\text{Max Sort / Join Memory Usage} = \frac{(\text{Total Memory} - \text{Suggested 4 GBs Operating System Memory})}{\text{\# of Simultaneous Workflows}}$$

For example, for a Worker configured with 32 GB of memory and 16 vCPUs, the recommended number of workflows running simultaneously is 4 because there are 16 vCPUs (1 workflow for every 4 vCPUs). In this example, 4 GB of memory set aside for the OS is subtracted from 32 GB total memory. The remaining number (28 GB) is divided by the number of simultaneous workflows (4), leaving 7 GB. Therefore, the recommended max sort / join memory is 7 GB.

Max Sort / Join Memory Usage for 32 GB Instance and 16 vCPUs = (32 GB – 4 GB) / 4 simultaneous workflows = 7 GB

The following table shows a list of pre-computed values for suggested max sort / join memory.

Instance vCPUs	Suggested Simultaneous Workflows	Total Memory (GB)	OS Memory (constant) (GB)	Suggested Max Sort/Join Memory (GB / Thread)
8	2	16	4	6
16	4	32	4	7
32	8	32	4	3.5
32	8	64	4	7.5
64	16	128	4	7.75

Table 2: Examples of suggested max sort/join memory

Database Performance

Using a user-managed MongoDB cluster allows you to control and tune the performance of the Alteryx Server persistence tier.

Availability Considerations

Except for the Controller, you can scale out the other major Alteryx Server components to multiple instances. Scaling the Worker, Gallery, and Database instances increases their availability, performance, or both. You can create a standby Controller to ensure availability in the event of a Controller issue, instance failure, or Availability Zone issue.

For high availability, you should deploy Worker, Gallery, and Database instances in two or three Availability Zones. Consider deploying instances in more than one AWS Region for faster disaster recovery, to improve interactive access to data for your regional customers, and to reduce latency for users in different geographies.

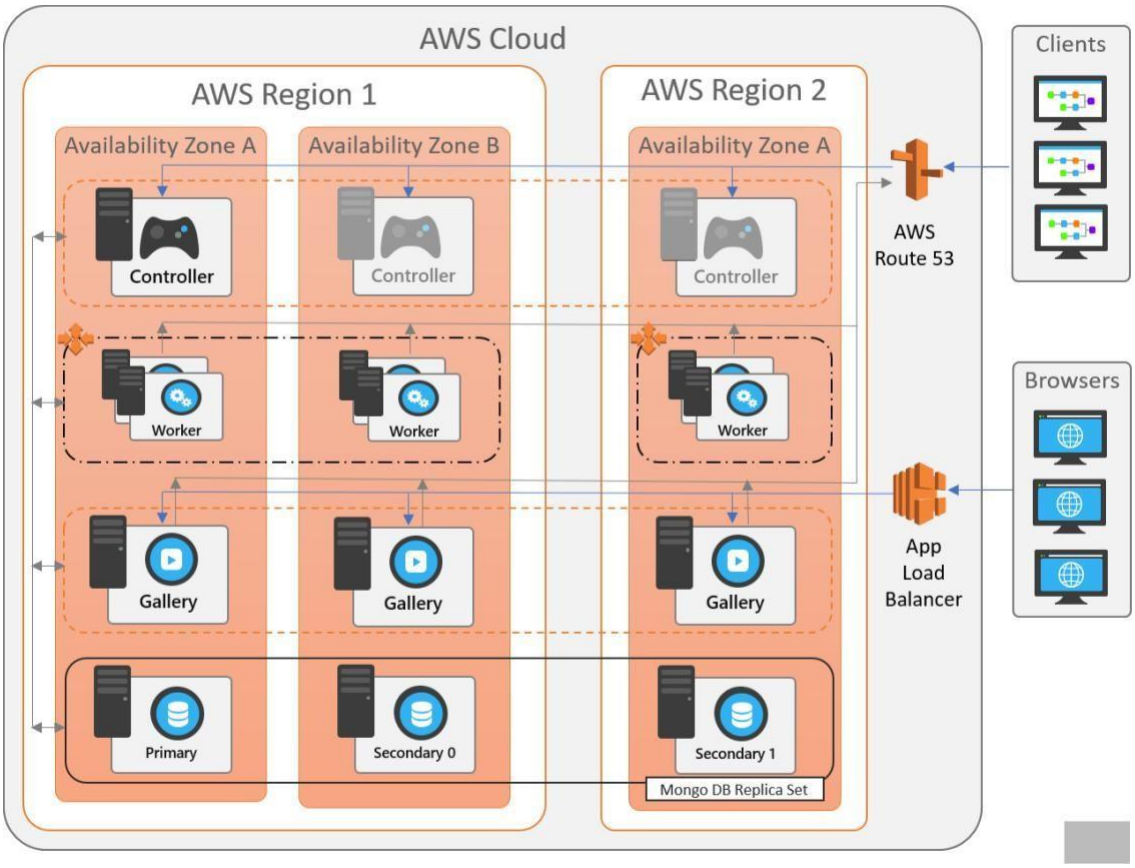

Figure 5: High availability deployment of Alteryx Server on AWS

AWS recommends that you have approximately 3-5 Worker instances, 2-4 Gallery instances behind an ELB application load balancer, and 3-5 Mongo Database instances configured in a Mongo DB replica set for high availability deployments. The worker instances depicted above were created with Amazon EC2 auto scaling. The exact numbers and instance sizes are dependent on costs and the performance sizing specific to your organization.

For multi-Region deployments, ensure that each AWS Region has a Controller instance that can be used with a DNS name (Elastic IP addresses are local to a single AWS Region). We recommend using Amazon Route 53 in an active-passive configuration to ensure there is only one active controller. The passive controllers can be fully configured, but Amazon Route 53 will only route traffic to a passive controller if the active controller becomes unavailable.

Management Considerations

Many of the configurations we discussed allow for more flexible management of Alteryx Server. Control of the persistence tier gives you more options when replicating and backing up the database. Placing the Gallery behind a load balancer allows for easier maintenance when upgrading or deploying Gallery instances. From an operational standpoint, a scaled install gives you more options and less downtime for backups, monitoring, database permissions, and third-party tools.

Remember, scaling Alteryx Server will have licensing implications based on the number of vCPUs in the deployment. You need to license all deployed nodes regardless of function.

Sizing and Scaling Summary

A high-level overview of reasons and decisions for sizing and scaling Alteryx is given in the table below.

Action	Performance Impact	Availability Impact	Management Impact
Controller Scaled Up – Larger Instance Size	Can help increase Gallery performance	No major impact	No major impact
Controller Scaled Out – More Controller Instances	No major impact	Having multiple Controllers requires that one Controller is on cold or warm standby	Requires customized scripts or triggers to automatically failover. You can create these with AWS services such as CloudWatch and SNS.
Worker Scaled Up – Larger Instance Size	Decreased workflow completion times. For best results, use instance types with more memory or optimized memory.	No major impact	No major impact
Worker Scaled Out - More Worker Instances	More concurrent workflows can be run	More resiliency to Worker instance failures	Reduced downtime during maintenance
Gallery Scaled Out - More Gallery Instances	Better performance for more Gallery users	More resiliency to Gallery instance features	Reduced downtime maintenance
User-Managed MongoDB database	More control for tuning and performance	Clustering and replication in MongoDB allow for higher availability	Give you more control over the database, but requires some knowledge about NoSQL databases

Table 3: Scaling actions and impact on performance, availability, and management

When considering Alteryx Server deployment options and which components to scale, it's best to consider your organization's performance, availability, and management needs. For example, your organization may have a few users creating analytic workflows but hundreds of users consuming those workflows via the Gallery. In that case, you might need minimal infrastructure to handle analytic workflows and the database, while the Controller, which aids the Gallery instances, would need to be a larger instance and the Gallery instances would be best served using several instances behind a load balancer.

If you are concerned with data loss, you should create a user-managed MongoDB cluster and make sure that it is backed up regularly to multiple locations.

Operations

This section discusses backup, restore, and monitoring operations.

Backup and Restore

You can use the Amazon Elastic Block Store (EBS) snapshot feature to back up the Controller, Worker, and Database instances. You can use these snapshots to restore data in the event of a failure. It is best to stop the Controller and Database tier before a snapshot. The Gallery is stateless and does not need to be backed up.

For details on how to perform backup and recovery operations if you are using a user- managed MongoDB database, see the MongoDB documentation for Amazon EC2 Backup and Restore.

In a distributed Server, the Gallery, Worker, and external Mongo nodes should be redundant. If the fault implicates a node other than the Controller then spin up a new instance of the component. However, if the fault was on the Controller or the Controller's AZ went offline, then spin up a new Controller VM (preferably using EBS Snapshots to restore a known good configuration – all state, except logs, is kept within the database.)

For reference, the estimated time for RTO/RPO are:

Standalone Server deployment with embedded Mongo

- RTO: 2-4 hours
- RPO: 24 hours (assuming a worst-case scenario of restoring from a snapshot)

Distributed Gallery with user-managed Mongo (or Atlas)

- RTO: 5 to 60 minutes
- RPO: 5 to 60 minutes (assuming a fault in the Controller or Controller's AZ, and reflects time to standup a new controller and incorporate into existing architecture)

Recovery Testing is a repeatable (ideally, automatable) subset of the Recovery Process that can be executed to provide confidence that the Recovery Process (perhaps in combination with a specific backup) will result in a functional service replacement. A recommended way to perform testing is to generate a temporary private VPC, spin-up a Mongo node (with its database disk seeded from a production secondary's EBS volume), deploy a Controller and Gallery node connected to the Mongo server, and test that a sample workflow will run.

Audit Logs

You can use Alteryx Server's built-in Audit Log for auditing purposes. It includes information on any changes to Users, Subscriptions, Collections, Credentials, and Workflows. Changes are logged, in addition to, the previous values. More details can be found at Audit Logs. Demonstrating how to back up proprietary Alteryx Server logs to S3, this example PowerShell script can also be used.

Monitoring

AWS provides robust monitoring of Amazon Elastic Compute Cloud (Amazon EC2) instances, Amazon EBS volumes, and other services via Amazon CloudWatch. Amazon CloudWatch can be triggered to send a notification via Amazon Simple Notification Service (Amazon SNS) or email, upon meeting user-defined thresholds on individual AWS services. Amazon CloudWatch can also be configured to trigger an auto-recovery action on instance failure.

You can also write a custom metric to Amazon CloudWatch, for example, to monitor current queue sizes of workflows in your Controller and to alarm or trigger automatic responses from those measures. By default, these metrics are not available from Alteryx Server but can be parsed from Alteryx logs and custom workflows and exposed to CloudWatch using Amazon CloudWatch Logs.

You can also use third-party monitoring tools to monitor status and performance for Alteryx Server. A free analytics workflow and application is available for reviewing Alteryx Server performance and logs. You can get that tool from the Alteryx support community.

Network and Security

This section covers network and security considerations for Alteryx Server deployment.

Connecting On-Premises Resources to Amazon VPC

For Alteryx Server to access your on-premises data sources, connect an Amazon Virtual Private Cloud (Amazon VPC) to your on-premises resources.

In the following figure, the private subnet contains Alteryx Server. You can place all the Gallery services in a public subnet (not shown) for simple access to the internet and users, or you can configure AWS Direct Connect or use VPN to enable a private peering connection with no public IP addressing required. You can also place Gallery instances or Alteryx Server in the private subnets with configuration of NAT Gateway. Scaling, hybrid, or disaster recovery options are also available in this model, with elements of Alteryx Server deployed as needed either on-premises or on AWS.

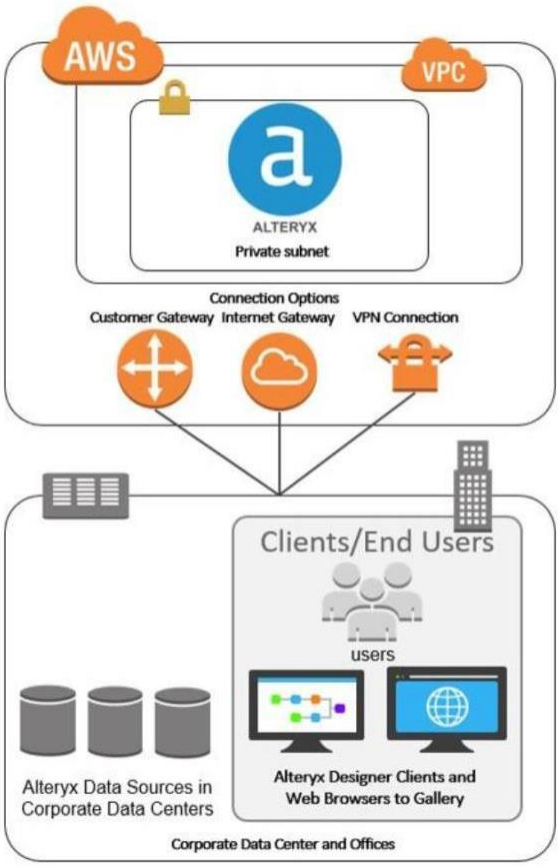

Figure 6: Options for connecting on-premises services to Alteryx Server on AWS

Alteryx Server often uses information stored on private corporation resources. Be aware of the performance and traffic implications of accessing large amounts of data that are outside of AWS.

AWS offers a several solutions to handle this kind of expected traffic. You can provision a VPN connection to your VPC by provisioning an AWS Managed VPN Connection, AWS VPN CloudHub, or a third-party software VPN appliance running on an Amazon EC2 instance deployed in your VPC. We recommend using AWS Direct Connect to connect to private data sources outside of AWS as it provides a predictable, low cost, and high-performance dedicated peering connection. You can also use VPN with Direct Connect to fully encrypt all traffic. This approach fits well into risk and security compliance standards for many corporations.

You may already be using Direct Connect to connect with an existing AWS deployment. It is possible to share Direct Connect and create connections to multiple VPCs, even across AWS accounts, or to provision access to remote regions. While possible, it is not recommended to connect to data sources directly over the internet from a public subnet due to security concerns.

See the Direct Connect documentation for more details on a variety of connectivity scenarios, see the AWS Direct Connect documentation.

Security Groups

When running Alteryx Server on AWS, be sure to check your security group settings when attempting to add a connection to a data source. You will need to customize your security groups based on your needs as some data sources may require specific ports. Refer to the data source documentation on the specific source you are connecting to and the ports and protocols used for traffic.

Port	Permitted Traffic
3389	RDP Access
80	HTTP Web Traffic
443	HTTPS Web Traffic
81	Used Only with AWS Marketplace Offering for Client Connections
5985	Used Only with AWS Marketplace Offering for Windows

Table 4: Security Groups for Alteryx Server

Network Access Control Lists (NACLs)

Amazon VPC and Alteryx Server support NACLs as an optional additional network security component. NACLs are not stateful and tend to be more restrictive, and so they are not recommended for general deployments. They may be useful for organizations with specific compliance concerns or other internal security requirements. NACLs are supported for controlling network traffic that relates to Alteryx Server.

Bastion Host (Jump Box)

In the case that Alteryx Server components are placed in a private subnet, we recommend that a bastion host or jump box is placed in the public subnet with security group rules to allow traffic between the public jump box and the private server. This adds another level of control and helps limit the types of connections that can reach the Alteryx Server. For details on bastion host deployment on AWS, see the Linux Bastion Hosts on the AWS Cloud Quick Start.

Secure Sockets Layer (SSL)

The Gallery component of Alteryx Server is available over HTTP or HTTPS. If you deploy gallery instances in a public subnet, we recommend HTTPS. For information on how to properly configure TLS, see the Alteryx Server documentation.

Technical Support

Alteryx provides customized technical support based on the requirements of each customer. A variety of support tiers and SLAs are available. Details can be found on our Support Page.

Best Practices

The following sections summarize best practices for deploying Alteryx Server on AWS.

Deployment

- Deploy Alteryx Server on an instance that meets the minimum requirements: Microsoft Windows Server 2008R2 (or later), at least 8 vCPUs, and at least 1TB of Amazon Elastic Block Store (Amazon EBS) storage.
- Do not change the Alteryx Server Authentication Mode once it has been set. Changing the Authentication Mode requires that you reinstall. Microsoft Windows Active Directory (Microsoft AD) or SAML 2.0 are the recommended authentication methods.
- The controller token is unique to each Alteryx Server installation, and administrators must safeguard it.
- Be sure to configure each required database driver on the server machine with the same version that is used for designer clients.
- Alteryx Server supports two different mechanisms for persistence: MongoDB and SQLite. Choose MongoDB if you need a scalable or highly-available architecture. Choose SQLite if you do not need to use Gallery and your deployment is limited to scheduling workloads.
- Worker instances, Gallery instances, and user-managed MongoDB instances can be scaled for deployments supporting user groups of 20 or more.
- If you use the pay-as-you-go AWS Marketplace image for test purposes, be sure to note the 14-day trial period and remember to turn your instance off once your trial is complete.
- It is recommended that you use a tagging strategy that defines the role the services plays with regard to Alteryx Server (i.e., ayx-role: controller, ayx-role: worker, ayx-role: gallery). For additional guidance on tagging strategies, reference AWS Tagging Strategies.

Scaling and Availability

- For a more resilient architecture, be sure to scale out worker, Gallery, and persistence instances across multiple Availability Zones. Consider deploying instances across AWS Regions to reduce latency for users in different geographies or to improve access to data.
- Multiple Gallery instances can be configured behind a load balancer to handle the Gallery services at scale.
- When scaling Worker instances, you should increase the size of all Worker instances as the Controller does not schedule on specific worker instances by priority
- A standby Controller can be deployed for failover. AWS tools such as AWS CLI, Amazon Route 53, and Amazon CloudWatch can help automate failover.
- Scaling Alteryx Server to more instances will likely have licensing implications because it is licensed by cores.

Network and Security

- Alteryx Server on AWS commonly process information stored on-premises. Be aware of the potential performance and cost implications of using large amounts of data outside of AWS.
- When using Alteryx Server on AWS, ensure that you check your security group settings when attempting to add a connection to a data source. You will need to customize security groups based on your needs as some data sources may require specific ports. Refer to documentation on the specific database you are connecting to and the ports and protocols used for traffic.
- Amazon VPC and Alteryx Server support NACLs as an optional additional network security component. NACLs may be useful for organizations with specific compliance concerns or other internal security requirements.
- Be sure your Alteryx Designer clients have connectivity to any Controllers you plan to schedule workflows on. This is an easily missed requirement when Alteryx Server is deployed in the cloud.

Performance

- Instance types with a larger ratio of memory to vCPUs will often run Alteryx workflows faster. Consider EC2 memory-optimized instances types, such as the R4 when working to improve performance.
- We recommend two VPCs per simultaneous workflow.
- The user-defined Controller setting max sort/join memory manages the memory available to workflows that are RAM-intensive. The best practice is to take total memory available to the machine and subtract a suggested 4 GBs of memory for OS processes. Then take that number and divide it by the number of simultaneous workflows assigned. For example: 32 GBs – 4 = 28 GBs / 4 simultaneous workflows = 7 GBs max sort/join memory.
- For workflows using geo-spatial tools, use EBS Provisioned IOPS SSD (io1) or EBS General Purpose SSD (gp2) volumes that have been optimized for I/O- intensive tasks to increase performance.
- A standby Controller can be deployed for failover. AWS tools such as AWS CLI, Amazon Route 53, and Amazon CloudWatch can help automate failover.
- Scaling Alteryx Server to more instances will likely have licensing implications because it is licensed by cores.

Conclusion

AWS lets you deploy scalable analytic tools such as Alteryx Server. Using Alteryx Server on AWS is a cost effective and flexible way to manage and deploy various configurations of Alteryx Server. In this whitepaper we have discussed several considerations and best practices for deploying Alteryx Server on AWS.

Please send comments or feedback on this paper to the papers authors or helpfeedback@alteryx.com.

Contributors

The following individuals and organizations contributed to this document:

- Mike Ruiz, Solutions Architect, AWS
- Claudine Morales, Solutions Architect, AWS
- Steve Wagner, Product Manager, Alteryx
- Mark Hayford, Amazon Web Services Architect, Alteryx

Further Reading

For additional information, see the following:

- Alteryx Community
- Alteryx Knowledge Base
- Alteryx Server Install Guide
- Alteryx SSL Information
- Alteryx Documentation

Document Revisions

Date	Description
August 2018	First publication
June 2019	Updated vCPU sizing recommendation
May 2020	Clarified requisite skills and services, auditing, tagging, pricing, support, and RPO/RTO