SteelCentral
Aternity SaaS
Architecture

Aternity SaaS v11.10
Overview

This document describes the Aternity SaaS architecture as it is implemented in the Amazon Web Service (AWS) cloud solution.

Aternity SaaS is designed from the ground up to scale both vertically and horizontally. This implementation can scale massively to support hundreds of thousands of monitored devices distributed around the world with high performance and high availability.

This document describes at a high level the AWS features that are used, the different components, the Aternity SaaS architecture, the backup and restore from database, the monitoring and the security mechanisms in place.
Aternity on Amazon Web Services (AWS)

The Aternity SaaS implementation uses the following Amazon features:

- **Locations**: Amazon Regions and Availability Zones (AZ)
- **Compute**: Amazon Elastic Compute Cloud, Amazon Auto Scaling (Amazon EC2)
- **Docker Management**: Amazon EC2 Container Service (Amazon ECS)
- **Storage**: Amazon Elastic Block Store (Amazon EBS)
- **Backup**: Amazon Simple Storage Service (Amazon S3)
- **Replications**: Amazon Machine Images (AMI)
- **Networking**: Amazon Route 53, Amazon VPC
- **Security**: Amazon WAF, Amazon Security Groups
- **Load Balancer**: Amazon ABS/EBS

**Amazon Regions** – Amazon Web Services are available in multiple regions around the globe. Regions consist of one or more **Availability Zones (AZ)**. AZ are distinct locations that are engineered to be insulated from failures in other Availability Zones. In case of disaster the system can be duplicated to a different Availability Zone in the same region (preferably) or different region which is not affected by the disaster event.

**Amazon EC2** – The Amazon Elastic Compute Cloud (EC2) is on-demand computing power (virtual instance) that can be created within minutes from a web-based console.

**Amazon ECS** – **The Amazon EC2 Container Service** is a highly scalable, fast, container management service that makes it easy to run, stop, and manage Docker containers on a cluster of Amazon EC2 instances.

**Amazon AutoScaling** – A feature that helps maintain the health and availability of the Amazon EC2 instance and ensures that you are running the desired number of Amazon EC2 instances.

**Amazon EBS** provides persistent, block-level storage volumes for Amazon EC2 applications within the same availability zone.

**Amazon Machine Images (AMIs)** are preconfigured with operating systems and Aternity application stacks. The AMIs are launched as part of the recovery procedure and reduce the time to install the software.
Amazon S3 is a cloud-based object store that is available through Web services interfaces such as REST and SOAP. It is designed to offer 99.99999999% (11 9s) availability of objects.

For Disaster Recovery (DR), point-in-time snapshots of Amazon EBS volumes of database backups are copied and maintained in Amazon S3 storage, limiting any data loss to that which was created since the last recovery point and recovery time interruption.

Amazon VPC and Route Tables let you provision a logically isolated section of the AWS Cloud where you can launch AWS resources in a virtual network that you define. A route table contains a set of rules that are used to determine where network traffic is directed.

Amazon Route 53 is a highly available and scalable Domain Name System (DNS) web service. Amazon Route 53 gives the ability to failover between multiple endpoints.

Amazon WAF is a web application firewall that helps protect your web applications from common web exploits that could affect application availability, compromise security, or consume excessive resources.
Components

Aternity SaaS is built from a set of loosely-coupled components that are tightly integrated into a highly-scalable solution:

- **Monitored devices (Endpoints)** – Any physical or virtual devices located on the customer site which report the end user experience to Aternity.

- **Oracle Database Server** – Hosts the Oracle Enterprise database used by Aternity to store all historical, transient and configuration data.

- **Cloned Oracle Database Server** – Standby database server that is updated in near real time.

- **Aternity Dashboard Server** – Hosts the Tableau Server instance and the Aternity Dashboard Gateway. The Tableau Server stores and generates Aternity’s dashboards. The Aternity Dashboard Gateway handles the integration between the Tableau Server and the rest of the system, and provides the following functionality:
  - Content deployment – Deploys dashboard layouts, data sources, and schedules on behalf of the Aternity Management Server.
  - Security management – Provides seamless user access to Tableau dashboards, and allows multiple Management Servers to share one Tableau Server instance.
  - Content view – Provides access to dashboards on the Tableau Server.

- **Aternity Management Server** – A core component handling core functions, system management, external integration, user interface, and reporting. All Aternity configuration, administration, and integration are performed centrally from the Management Server’s user interface.

- **Data Warehouse Server** – A core component dedicated to handling the data arriving from the Aggregation Servers, and populating it into Oracle Database according to the specified retention policy.

- **Kafka Servers** – A component that is used for building real-time data pipelines from the Aggregation Servers to different components of the Aternity solution.

- **Raw DB Server** – A core component dedicated to handling part of the raw data arriving from the Aggregation Servers. This component is used to provide a high performance, high availability, and high scalability solution which will replace over time the Data Warehouse server’s role of saving the raw data.

- **Aggregation Servers** – Aggregation Servers are used for the bi-directional communication between the monitored devices and the Management Server, and to aggregate measurements which they pass on to the Management Server. Aggregation Servers typically reside in physical proximity to the devices which report to them.

- **Vertica Database Servers** – Store the performance data from the past 1-2 years in the Vertica format, which is most efficient for displaying in Aternity dashboards. It receives its data from the Aternity ETL Server which translates performance data from the Aternity Oracle Database Server to this server.

- **REST API (Data External)** – REST API is used to load Aternity data directly into Microsoft Excel, Power BI, or other data application which is compatible with the OData format, without going through Aternity’s dashboards.
• **REST API Streaming (Data External)** – REST API is used to continuously pull data from Aternity to external systems.

• **Aternity Data Source (Data External)** – The SteelCentral Portal™ offers end-to-end digital experience management by enabling you to view data from several products in the SteelCentral Suite side by side in a single dashboard. The Aternity Data Source is used to provide the Portal a way to view Aternity’s data from the past seven days alongside that of other products.

• **Docker Data Internal Servers** – Containers that are responsible for the following tasks:
  - **Host Resources (HRC) and Installed Applications** tuples are sent from the Aggregation Servers to Kafka servers that manage the tuple queue and are consumed by the HRC component.
  - **Vertica Writer (Data Internal)** – Manages inserting data to the Vertica database.
  - **Vertica Scheduler (Data Internal)** – Schedules jobs on Vertica database.
  - **Remediation Server (Data Internal)** – Manages all remediation actions.
  - **Data Processing (Data Internal)** – Processes host resources tuples and installed applications.
  - **Agent Administrative Pages (Data Internal)** – Manages viewing and operating of the Agent Administration page.
  - **Service Desk Alerts (Data Internal)** – Manages service desk alerts.
  - **Data Anomalies (Data Internal)** – Detects data anomalies.
Aternity SaaS Architecture

Aternity SaaS architecture includes:
- $n$ Aggregation Servers deployed across availability zones
- 1 Management Server
- 1 Tableau Server
- $n$ Vertica Servers (cluster)
- $n$ Raw Database Servers (Cassandra) deployed across availability zones
- 1 Data Warehouse Server
- $n$ Kafka Servers
- 1 Oracle Database Server
- 1 Oracle Standby Database on different availability zone
- 1 Docker Data Internal Server that includes containers for internal processes
- 1 Docker Data External Server that includes:
  - Aternity REST API Server deployed across availability zones using docker and auto scaling
  - Aternity Data Source Server deployed across availability zone using docker and auto scaling
- 1 Docker REST API Synchronization Server that is used to continually pull all necessary information from Aternity
Docker Data Internal Cluster includes containers that are responsible for the following tasks:

- Processing Host Resources tuples
- Processing Installed Applications information
- Managing Remediation actions
- Managing Service Desk Alerts
- Viewing and handling Agent Administration page
- Writing data to Vertica database and scheduling aggregations
- Analyzing data anomalies
Docker Data External includes:

- REST API
- REST API Streaming
- Portal Data Source

The data is exposed to users or external sources (Excel, Power BI, Steel Central Portal).
The information provided by Riverbed is only for purposes of internally evaluating a potential business relationship and is considered by Riverbed to be confidential and proprietary information.
Monitoring Aternity Performance

Aternity SaaS is monitored by a central monitoring system.

The monitors are divided into the following layers:

- **Uptime** – Monitoring the availability of the application 24x7 and sending alerts by phone, SMS, and email
- **Host Resources** – Monitoring the CPU, memory, network, disk space, disk I/O, and latency
- **Performance** – Monitoring the performance of the application to ensure best use of the system. KPI metrics are collected and sent to the central monitoring system for evaluation and trend analysis.
- **Logs** are collected to centralized locations for analysis.

The Aternity Product Operations team proactively performs health checks on the functionality of the system on a regular basis.

Oracle Database Backups & Restores

The Oracle database is backed up using the following methods:

- Clone database that is updated near real time and is located in a different AWS Availability Zone
- Daily Recovery Manager Backup that is saved as an Amazon snapshot on AWS S3

Aternity keeps 7 days data retention.

Database restore is performed periodically on a staging database to ensure it works properly.

Vertica Database

Vertica is used to enhance the performance of dashboards. The Aternity ETL Server copies data from Oracle to the Vertica cluster which is then fed to the Tableau dashboards.
Disaster Recovery

Disaster recovery (DR) is about preparing for and recovering from a disaster. A disaster is described as a significant outage event such as physical damage to a building which affects the ability to use the application for a significant time.

The DR process is used by the Aternity Product Operations team. They analyze alerts sent from the Amazon Cloudwatch and other external monitoring systems, to confirm the outage event is indeed significant.

Aternity has mapped out a solution to provide its customers with a cost-effective system recovery within a reasonable timeframe, where data loss is covered by AWS services and features.

The solution is based on a standby system that is located on a separate AWS Availability Zone.

RTO: Aternity expects to recover from a disaster within 24 hours, for the worst scenario in which the clone database is damaged as well. For the scenario in which the clone database is not damaged, Aternity expects to recover within less than 1 hour.

RPO: Aternity expects that the maximum targeted period in which data might be lost is 24 hours.

Security Concept

The Aternity solution is certified as SOC 2 compliant. The following section describes the main security concept of the Aternity SaaS solution:

- Connection to AWS is done using Multi Factor Authentication (MFA).
- All AWS instances are connected to AWS VPC. Each component has its own subnet.
- Routing tables, NACLs and Security Groups are defined to allow communication only between relevant servers.
- All AWS instances have only internal IP addresses.
- All applications are exposed to the internet behind Amazon Load Balancer and over https.
- REST API, REST API Streaming and Portal Datasource are protected by AWS WAF.
- Database volumes are encrypted.
- All Windows instances are protected using Antivirus and Intrusion Prevention System solutions.
- Active Directory on the SaaS environment is used to allow authentication to servers.
- Access to servers is allowed only via VPN with Multi Factor Authentication (MFA) and only through jump box.
- Servers Hardening is done using group policies.
- Security updates are performed on a monthly basis.
- Any login and fail login to the system are tracked.