A10 LIGHTNING ADC

Architecture
TABLE OF CONTENTS

INTRODUCTION 3
   Deployment Architecture 3
   A10 Harmony Controller 4
   A10 Lightning ADC 5
   A10 Harmony APIs 5
   A10 Harmony Portal 5
   Communication between Harmony Controller and Lightning ADC 6

MODELING APPLICATION TRAFFIC FLOW 7
   Application 8
   Domain Endpoint 8
   Service Endpoint 8
   Smart Flow 8
   Policies and Other Configuration Items 8
   Provider and Tenant 8

ENABLING TRAFFIC FLOW 9

UNDERSTANDING THE FUNCTION OF LIGHTNING ADC 9
   Traffic Optimization 9
   Traffic Segmentation 10
   Application Server Monitoring 11
   Enhanced Availability 11
   Application Security 12

FEATURES PROVIDED BY THE CONTROLLER 13
   Central Management 13
   Auto-Scaling 13
   Consolidated Insights 13
   Traffic and Security Analytics 13
   Alerting 14
   Keeping Threat Intelligence Updated 14
   Multi-tenancy 14

DEPLOYING LIGHTNING ADC 14
   Impact on Enterprise Perimeter IP Firewall 15

DEPLOYING AND OPERATING THE HARMONY CONTROLLER 15

SUMMARY 16

DISCLAIMER
This document does not create any express or implied warranty about A10 Networks or about its products or services, including but not limited to fitness for a particular use and non infringement. A10 Networks has made reasonable efforts to verify that the information contained herein is accurate, but A10 Networks assumes no responsibility for its use. All information is provided “as-is.” The product specifications and features described in this publication are based on the latest information available; however, specifications are subject to change without notice, and certain features may not be available upon initial product release. Contact A10 Networks for current information regarding its products or services. A10 Networks’ products and services are subject to A10 Networks’ standard terms and conditions.
INTRODUCTION

The A10 Lightning™ Application Delivery Controller (ADC) works with A10 Harmony™ Controller to provide a cloud-native solution to optimize the delivery and security of applications and services running over public, private clouds and hybrid clouds.

It provides innovative, elastic application load balancing capabilities, including traffic management with content-switching, application security and analytics for cloud applications. Lightning ADC is purpose-built for containers and microservices-based application architectures and elegantly integrates with DevOps processes.

For organizations embracing the cloud and application centricity, A10 Lightning ADC increases operational efficiency, offloads application administrators from cumbersome tasks, reduces risk and helps meet compliance requirements.

DEPLOYMENT ARCHITECTURE

A10 Lightning ADC is purpose-built to serve not just traditional Web applications, but also modern microservices and container-based applications.

The solution offers a highly scalable, software-defined distributed architecture with a separation of control and data planes. This allows the A10 Lightning ADC data plane elements to be lightweight and deployed close to, or embedded within, the application environment.

Figure 1: A10 Lightning ADC is unique from traditional application delivery controllers (ADC) in that it separates control and data planes to help reduce management overhead and infrastructure costs.
Lightning ADC has four major architectural components. Unlike traditional ADCs, Lightning ADC has a split architecture in which the data plane and the management and control plane are separated.

Data plane elements can be replicated and scaled independently of the management plane. Having a single point of control and management not only reduces cost but also significantly reduces management complexity.

**A10 HARMONY CONTROLLER**

The A10 Harmony Controller provides centralized management, policy configuration, monitoring, control, a big data repository and an analytics engine. It manages clusters of software-based A10 Lightning ADC instances that implement and enforce policies. The solution provides a single point of management to the administrators for policy configuration, as well as information consumption in a distributed deployment environment.

The controller is a scalable, microservices-based application that is delivered as a SaaS by A10, or may be deployed within an organization’s private cloud. The complete controller subsystem may be deployed on a number of virtual machines (VM) or in Docker containers. The containers are internally managed by a container orchestration system making the system highly available and scalable.

*Figure 2: The A10 Harmony Controller is a cloud-native, microservices-based application.*
A10 Lightning ADC

A10 Lightning ADC is a compact, efficient full-proxy that front-ends cloud applications and microservices to execute Layer 4-7 application delivery policies.

The A10 Lightning ADC core that handles HTTP-parsing, load-balancing and other basic functions is the same as the popular open-source webserver NGINX. A10 not only enhanced NGINX to suit distributed deployments and central managements, but also included many modules for adding functionalities like metrics collection, distributed traffic rate-limiting and data security.

A10 Lightning ADCs are typically deployed as a cluster in the network where the application servers are running and communicate with the controller over a secure messaging infrastructure. Although A10 Lightning ADC instances are stateless and fully managed by the controller, they are completely independent when it comes to application traffic-handling. Application traffic never hits controller.

Once configuration and policies are delivered to Lightning ADCs, application traffic is never disrupted, even when the link between the controller and Lightning ADC fails.

However, during the time of such link failure, Lightning ADCs don’t receive new configuration changes from controller and don’t deliver traffic metrics to controller. When the link is re-established, Lightning ADC automatically synchronize its configuration from the controller.

A10 Harmony APIs

A10 Harmony Controller features RESTful APIs that allow complete programmatic control over the ADC. The APIs not only automate administrative tasks, but also retrieve data from the system for better traffic intelligence and decision-making.

For automating the workflows of regular continuous integration and deployment (CI/CD), A10 Harmony APIs can be used with popular DevOps tools, including Chef, Puppet, Ansible and SaltStalk.

To achieve better orchestration control, Lightning ADCs may be launched via the deployment automation tools. In addition, APIs can be utilized to build specific Blue/Green deployment use cases, fine-tune policy or alter policies based on analytics feedback.

A10 Harmony Portal

A10 Harmony Portal is a Web-based API client developed using A10 Lightning APIs. This provides easy-to-use visual configuration controls for all major workflows and drastically simplifies the user experience.
COMMUNICATION BETWEEN HARMONY CONTROLLER AND LIGHTNING ADC

Like all other ADCs, Lightning ADC experiences traffic between the management interface (e.g., portal or API clients) and controller. This traffic primarily sets up policies and collects analytics data.

However, because of the split architecture, communication between the controller and Lightning ADCs is also significant. This traffic not only maintains ADC health and keeps configurations in sync with the controller, but also includes the metrics collected by ADCs about the traffic and application servers. These metrics feed into the rich analytics engine at the controller and are used as part of a feedback loop for policy optimization.

To enable these traffic flows, connection is always initiated by Lightning ADC. There is never a requirement to open incoming ports in the IP firewall when the Lightning ADC is installed within an enterprise network.

During launch, the API Server URL and Cluster ID are configured into the Lightning ADC. When the Lightning ADC starts, it connects to the API server for authentication and collects information about other microservices within the controller.

Next, Lightning ADC creates a persistent connection to a message bus for configuration synchronization, and another persistent configuration with metrics collector to push the collected metrics to the analytics subsystem.
**MODELING APPLICATION TRAFFIC FLOW**

Application traffic traverses only the Lightning ADC and not the controller. As the application DNS name is mapped to the ADC IP addresses, clients make TCP connections with Lightning ADC and send HTTP/S requests.

Being a full proxy, Lightning ADC fully terminates the TCP connection and opens separate persistent connections with application servers. The HTTP request is sent separately to application servers and the response is forwarded back to the requesting client. In the middle, all policies are appropriately applied to requests and responses.

Lightning ADC follows an application-centric view — as opposed to a network-centric view — of traditional ADCs. Hence, the configuration flow is the same as the traffic flow at the application layer.

In practice, an application and its microservices can be architected in various ways. The Lightning ADC configuration model is designed to accommodate many different scenarios.

*Figure 3: Hierarchical configuration model of A10 Lightning ADC.*
APPLICATION
The top-level application container represents the logical application abstraction. Consider this a group of microservices offering a logical functionality that makes sense to the application users. Each of these microservices represent a subset of the application. A tenant (account) may have any number of applications.

DOMAIN ENDPOINT
A group of Fully Qualified Domain Names (FQDNs) is represented as a domain endpoint. For various business reasons, it may be required to offer the same functionality on multiple domains. All these domains may be configured into a single domain endpoint. Since all domains offer the same functionality, they share the same infrastructure as well.

SERVICE ENDPOINT
A service is identified by a set of application servers and a segment of traffic served by these servers. All servers in a service offer the exact same functionality and share the load. Different services are created to serve each traffic segment. This enables content-based switching for the application traffic.

SMART FLOW
The next level of content-based traffic segmentation is provided by Smart Flows for granular policy application.

POLICIES AND OTHER CONFIGURATION ITEMS
Lightning ADC configuration gives users full control over their application traffic. Knobs and controls are available at each level to manage, customize, secure and optimize traffic, as needed.

PROVIDER AND TENANT
Harmony Controller makes Lightning ADC a multi-tenant service. Multiple tenant accounts may be created under a provider. Further tenants can create their own applications and Lightning ADC clusters. Having Provider and Tenant as organizational units within Harmony Controller enable agile and self-service IT operations in an enterprise where central IT team acts as providers of application delivery services to the individual business units and application teams. Self-service can be provided to the individual application teams who can efficiently deploy ADC clusters, operate and control the policies of their own applications. Similarly, the provider-tenant model enables Managed Service Provides (MSPs) to serve their customers efficiently.
ENABLING TRAFFIC FLOW

A DNS name is exposed for each Lightning ADC cluster. For traffic to reach Lightning ADCs, the DNS of each domain in the domain endpoint needs to point to the DNS name of the Lightning ADC cluster. Typically, a ‘CNAME’ record is required in the DNS manager (zone file) for this purpose. An 'A' record, which includes IP addresses of all Lightning ADCs, may also be created.

After the DNS change is complete, traffic destined for the application domains flows via the Lightning ADCs. The Lightning ADCs applies policies and other benefits to both the traffic as well as the entire infrastructure.

UNDERSTANDING THE FUNCTION OF LIGHTNING ADC

Lightning ADCs work as a cluster rather than individual units. For an application client, it doesn't matter which Lightning ADC it connects to. In fact, it can connect to a different Lightning ADC every time it makes a new TCP connection to send requests.

TRAFFIC OPTIMIZATION

Delivering responses of every application request in the least amount of time requires multiple optimizations. Lightning ADC not only helps optimize the code by granularly reporting server latency, but also applies various other techniques to optimize overall traffic.

SHARED SESSION STATE

All clustered Lightning ADCs have access to a shared session state. This shared state is stored outside Lightning ADCs in a high-speed data store. Hence, removal of Lightning ADC doesn’t impact the stored state and the newly launched Lightning ADC doesn’t need to be re-built from scratch.

CONNECTION MULTIPLEXING

While serving the traffic, opening and closing TCP connections consumes heavy resources. To improve overall efficiency, Lightning ADC keeps TCP connections with application servers persistent to reduce overhead. All client requests are served over these limited connections. This saves server resources and significantly reduces overall response times.

Figure 4: A10 Lightning ADC performing connection multiplexing on traffic.
**COMPRESSION**

Content compression significantly reduces the bandwidth requirement and makes communication faster. Offloading compressions to Lightning ADCs saves application server resources.

**CACHING**

Inline-caching built into Lightning ADCs not only reduces the load from application servers and their programmers, but also eliminates the requirement of another point product. Caching at Lightning ADC can be enabled for any smaller segment of traffic and it fully obeys cache-control headers set by application servers.

**TRAFFIC SEGMENTATION**

Segment traffic into desired parts to allow admins to treat each segment differently. Segmentation can be done based on any information available in HTTP request. A simple or complex regular expression may be created using information in URL, cookies and HTTP headers for segmenting traffic.

**CONTENT-BASED SWITCHING**

Traffic matching the regular expression may be sent to different server groups. This serves different portions of the application from different sets of servers based on need. For example, serving dynamic content requires an application server, but static content can be served from a Web server. One might want to dedicate servers for specific types of users (e.g., iPhone) or some specific customer with strict SLAs.

**GRANULAR POLICY APPLICATION**

Many times, it is required to apply different policies on different parts of the traffic even if they are served from same server. Admins can do second-level segmentation for granular policy application. For example, protection against SQL injection should only be enabled for the pages that are vulnerable to SQL injection.

**TRAFFIC SPLIT FOR BLUE/GREEN OR A/B OR CANARY DEPLOYMENTS**

Rolling out features to limited sets of users is a strong technique that’s implemented by most large organizations. This not only helps marketing, but also tests the feature on a limited set for rapid improvement. For Lightning ADC users, this capability comes without additional effort.
APPLICATION SERVER MONITORING
As soon as a Lightning ADC cluster is attached to an application, it starts monitoring the application servers that serve traffic for all application services.

IN-BAND MONITORING
In-band traffic monitoring is achieved along with traffic serving. When an application server responds to a request successfully, it is also marked as ‘functional’ by the system. Repurposing regular traffic for monitoring significantly reduces the monitored traffic load on the server, especially at high-traffic times when the server needs it most.

When a server consistently fails to respond to requests, it is marked as ‘down’ and taken out of the load-balancing pool. Out-of-band monitoring continuously checks the down server and brings it back into the load-balancing pool when it responds to pings again.

OUT-OF-BAND MONITORING
When a server is not responding properly, in-band monitoring is not useful to find the server status as it delays the response significantly. Also, when there is less traffic, keeping correct server states and alerting administrators in a timely fashion is not possible with only in-band monitoring. A separate probe is sent from Lightning ADC to check the state of the servers.

By default, this is done by making a TCP connection to the server. However, users can configure a HTTP(s) URL to ping the server as well.

ENHANCED AVAILABILITY
Continuous application monitoring is a best practice to ensure server availability. Lightning ADC exposes some configuration parameters that administrators can use to increase application availability. Lightning ADCs learn about the environment dynamically via various collected metrics and automatically selects the appropriate methods to ensure enhanced application availability.

SERVER HOPPING
When a server fails to respond to a request, Lightning ADC attempts the same request to another server. With this method, the client only receives a failure when all servers fail to respond to the request. This increases overall service availability for the clients.

BACKUP SERVER GROUP
To further improve service availability, administrators can configure another server group to serve traffic when all servers in the primary group are down.
REQUEST QUEUING

Lightning ADC monitors servers for health parameters — like number of connections and latency — so the server is never overloaded. In case of traffic burst (when server health starts deteriorating), Lightning ADC places requests in a queue and dispatches them at a rate that servers can handle. This raises requests response times but prevents the service from going down.

APPLICATION SECURITY

Securing applications and traffic from various threats and attacks is of utmost importance. Lightning ADC unifies security with load balancing to provide ease of management, better integration and lower costs.

 Typically, cloud infrastructure providers work on a shared security model. In this model, security across L1 to L3 of the network stack is the responsibility of the infrastructure provider; security of L4 to L7 falls in the application owner’s court.

 Mostly, cloud providers completely control L1 and L2 of network stack, but provide tools and guidance to manage security controls for L3 and L4. Application layer (L7) security is the sole responsibility of the application owners.

 Lightning ADC eliminates the need of yet another point solution. Multiple layers of security are designed to avoid, report, mitigate and prevent attacks.

Figure 5: A10 Lightning ADC protects applications via several built-in security layers that detect and mitigate advanced cyber attacks.

Threat actors and other malicious groups plan attacks by profiling both the environment and its applications. Lightning ADC helps reduce or eliminate meta data and other information that is often used for profiling. This includes deleting or rewriting response headers, response codes and response body.

Lightning ADC also protects sensitive users by blocking malware, encrypting cookies, rewriting response body and blocking responses containing sensitive data.

Lightning ADC provides Web application firewall (WAF) controls to protect against the top-10 OWASP vulnerabilities like SQL injection, command injection, cross-site scripting (XSS), cross-site request forgery (CSRF), function-level access control, remote or local file inclusion, and more.
Elastic architecture and connection multiplexing techniques make the infrastructure resilient to distributed-denial-of-service (DDoS) attacks. DDoS attacks are mitigated by quickly identifying the attackers, configuring access control, and dynamically changing HTTP timeouts and traffic-rate limiting.

The combination of rich and granular security analytics, coupled with automation support, creates a powerful tool for adjusting security postures on-the-fly and combating attackers.

**FEATURES PROVIDED BY THE CONTROLLER**

The controller interfaces between administrators and distributed Lightning ADCs for complete management and control functions. It collects traffic metrics from various sources and presents a composite and correlated view of actionable traffic insights and application analytics that simplifies decision-making.

**CENTRAL MANAGEMENT**

The most important capability of the controller is central management. In an elastic and distributed environment, managing ADCs separately is a monumental pain point.

With A10 Lightning ADCs, maintaining up-to-date configurations is a simple task. While the controller presents singular configuration interfaces to administrators, it puts the intelligence of converting it (especially overall traffic limits) into distributed values and dynamically manages the information via Lightning ADCs.

**AUTO-SCALING**

The controller also continuously monitors the health of Lightning ADCs and the interface of the underlying API infrastructure that helps bring Lightning ADCs up and down.

Wherever supported, the controller also subscribes to the triggers from the underlying infrastructure and acts on them appropriately. For example, the controller may listen to application server-scaling triggers and add or remove them in appropriate server groups.

**CONSOLIDATED INSIGHTS**

Continuous metrics are collected by the controller, processed and presented to administrators in the form of actionable insights. Because of the consolidation and correlation functions of the controller, administrators gain a unified view of the distributed application environment, per defined logical application abstraction.

**TRAFFIC AND SECURITY ANALYTICS**

The controller also runs big-data-style analytics engine to analyze the collected data to detect anomalous trends. Traffic is also examined for possible security threats. These analytics are not only displayed in the portal to provide better context, but also include various recommendations and alerts based on the information.
ALERTING

The alert framework allows administrators to set up alerts on various metrics or customizable fields. Alerts may be defined at granular levels, per the application abstraction modal.

When conditions are matched and alerts are triggered, administrators are delivered notification emails for manual action. Additionally, the alerts are delivered to the configured Web-hook URL for automated, rapid action.

KEEPING THREAT INTELLIGENCE UPDATED

Discovering vulnerabilities, their exploits and their defenses is an everyday job of hackers and security professionals alike. As a result, threat intelligence changes daily. Keeping this data in sync is critical for mitigating zero-day attacks.

A10 Networks subscribes to these updates. A10 security teams monitor the items applicable for A10 users and publishes relevant updates regularly. The controller facilitates threat intelligence updates from the central repository to the Lightning ADCs.

MULTI-TENANCY

Along with facilitating application management in distributed environments, Harmony Controller can also host multiple service providers and tenants on a single ADS installation. The provider/tenant model allows IT teams to offer ADC-as-a-Service (ADCaaS) to their internal or external customers.

DEPLOYING LIGHTNING ADC

The solution architecture offers versatility and flexibility for deploying Lightning ADCs, as needed by an application environment. However, deploying Lightning ADCs in the same subnet as application servers is the best practice from traffic optimization and security points of view.

Typically, Lightning ADCs are deployed in active-active clusters for high availability and load-sharing. Lightning ADC configurations are fully managed by the controller based on policies configured by administrators. When permitted by infrastructure APIs, the controller also launches Lightning ADC clusters and scales up/down cluster sizes based on traffic volume and other health parameters.

A10 Lightning ADCs may run within a customer infrastructure in public clouds (e.g., AWS, Azure or Google Cloud), private clouds (e.g., OpenStack) or data centers (e.g., VMware). The Lightning ADC instances can run as a VM in any environment or as a Docker container. The controller is capable of simultaneously supporting instant deployments in multiple clouds for expanded scalability, flexibility and choice.
IMPACT ON ENTERPRISE PERIMETER IP FIREWALL

Lightning ADC is involved with two types of communication. One is accepting traffic from application clients and serve them with the response obtained from application servers. The other is to communicate with the controller to keep up-to-date policy configurations and to send various metrics to the controller for analysis.

No change in networking or IP firewall rules is needed to deploy Lightning ADCs; the controller never initiates inbound connections to Lightning ADCs. It is the Lightning ADC that initiates connections to the cloud controller on outgoing TCP ports.

For enhanced security, access to application servers from outside networks should be eliminated. This may be achieved by either blocking IP firewall access or removing public IP address of the servers.

Lightning ADCs should be on public IP address, but servers may be on private IP address. Being in the same IP subnet, Lightning ADCs can access the servers. This may require configuring more than one network on the Lightning ADC instances using the underlying deployment infrastructure. Lightning ADC software adapts to the underlying network and requires no additional configuration.

DEPLOYING AND OPERATING THE HARMONY CONTROLLER

The controller is a highly scalable, multi-tenant system based on a Web-scale, microservices architecture. Each microservice, including metrics collection, analytics and control function, can be independently scaled to meet the requirements of different deployment use cases.

The services run in Docker containers and can be packaged into single or multiple VMs. Internally, the system uses a container orchestration system that makes it simple to deploy and operate, but also ensures high availability for each service.

The controller is delivered as a service by A10, which also operates and offers the controller as a managed service. This significantly reduces the operational and infrastructure costs and complexity for organizations. Users just need an account on the multi-tenant controller and can quickly start using the system.

Even when the SaaS controller is being used, application traffic always remains within the organization’s network and no change is required to an enterprise IP firewall.

Optionally, the controller may be deployed in an organization’s account in public/private cloud or data center.
SUMMARY

A10’s solution with Lightning ADC and Harmony Controller has a unique ADC architecture, where management, control and analytic functions are provided by the central controller. Traffic-handling and policy enforcement is maintained by A10 Lightning ADC.

The A10 Harmony Controller is available as a SaaS and can also be deployed in a customer account.

A10 Lightning ADCs can be deployed as VMs or Docker containers in the networks where application servers are located.

Lightning ADC offers various traffic management, optimization, security, insights and analytics capabilities to its users. The configuration model is application-centric and accommodates various types of application architectures. Lightning ADCs are easy to manage via the A10 Harmony Portal. All the interactions with Harmony Controller may also be automated via A10 Harmony APIs. Overall, this solution provides the best of both on-premise and SaaS worlds – and is the best application delivery solution for application owners.

NEXT STEPS

To learn more about the A10 Lightning ADC, please contact your A10 representative or visit a10networks.com/lightning-adc.

ABOUT A10 NETWORKS

A10 Networks (NYSE: ATEN) is a Secure Application Services™ company, providing a range of high-performance application networking solutions that help organizations ensure that their data center applications and networks remain highly available, accelerated and secure. Founded in 2004, A10 Networks is based in San Jose, Calif., and serves customers globally with offices worldwide.

For more information, visit: a10networks.com or tweet @a10Networks