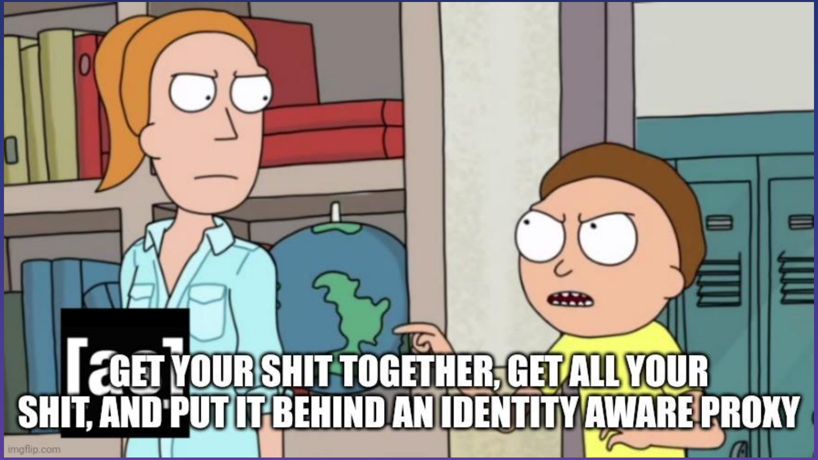
setting up an identity aware proxy

what is an identity aware proxy?

- an identity-aware authentication and authorization layer in front of your web applications
- roles/access policies are managed by an identity provider instead of your applications

why do we want one of these?

- additional layer of protection in front of your application
- users must be authenticated, authorized, and validated with an identity provider before they're able to access your application
- gets us closer to a zero trust model
  - "don't trust anyone until they've been verified"


[as]
GET YOUR SHIT TOGETHER, GET ALL YOUR SHIT, AND PUT IT BEHIND AN IDENTITY AWARE PROXY
imgflip.com

how do we do it?

- choose your identity provider (IdP)
- set up a load balancer in front of your application
- configure your load balancer to either:
  - use OpenID Connect (OIDC) with your IdP
  - use AWS Cognito configured with your IdP with one of the following:
    - SAML
    - OIDC
    - Google
    - Facebook
    - Amazon
    - Apple

onward!

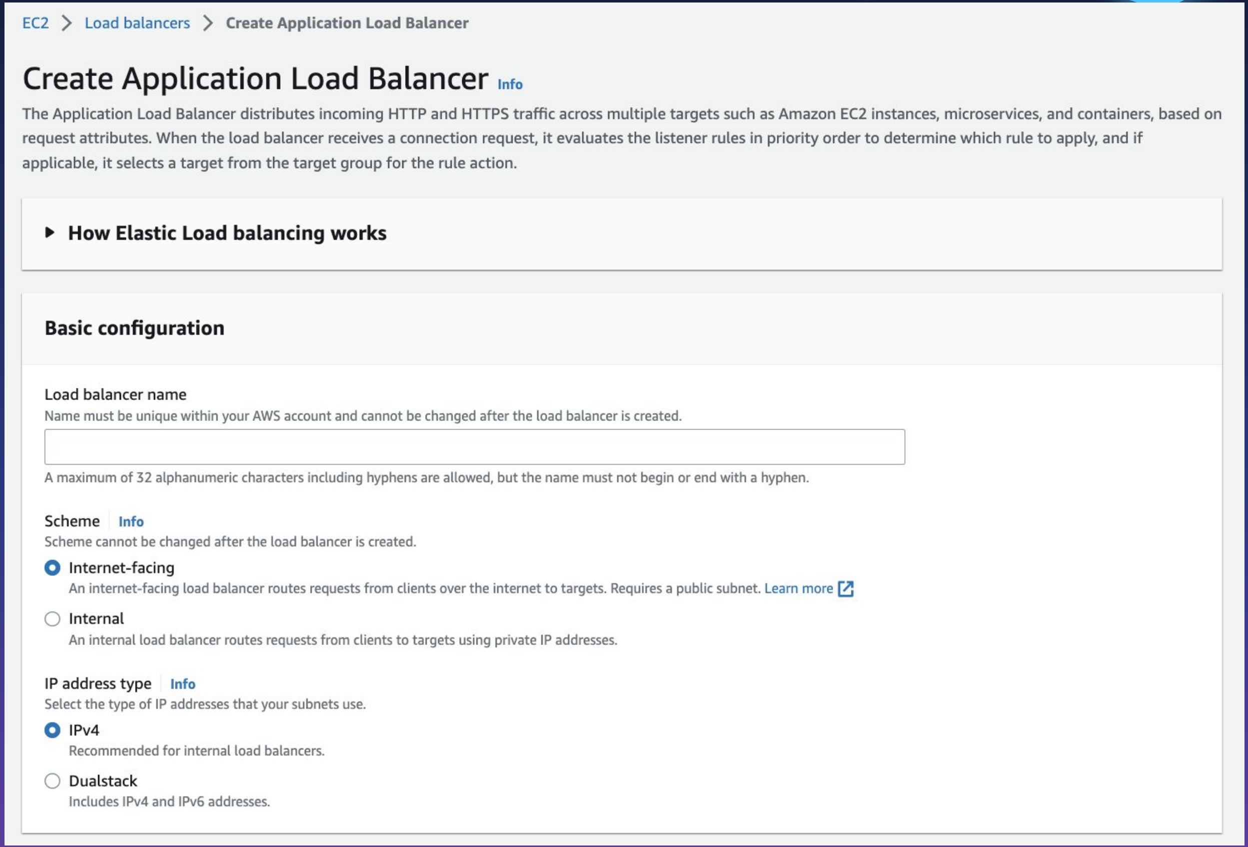
let’s set one up :)

choose your identity provider

- there are lots to choose from–find what fits your organization and use case
  - price
  - user limits
  - MFA options
  - passwordless options
  - self hosted or managed
  - SLA

create a load balancer

- name
- scheme
- VPC
- security group
- listeners
- SSL certificates



EC2 > Load balancers > Create Application Load Balancer

Create Application Load Balancer Info

The Application Load Balancer distributes incoming HTTP and HTTPS traffic across multiple targets such as Amazon EC2 instances, microservices, and containers, based on request attributes. When the load balancer receives a connection request, it evaluates the listener rules in priority order to determine which rule to apply, and if applicable, it selects a target from the target group for the rule action.

▸ **How Elastic Load balancing works**

Basic configuration

Load balancer name

Name must be unique within your AWS account and cannot be changed after the load balancer is created.

A maximum of 32 alphanumeric characters including hyphens are allowed, but the name must not begin or end with a hyphen.

Scheme Info

Scheme cannot be changed after the load balancer is created.

● Internet-facing

An internet-facing load balancer routes requests from clients over the internet to targets. Requires a public subnet. Learn more

○ Internal

An internal load balancer routes requests from clients to targets using private IP addresses.

IP address type Info

Select the type of IP addresses that your subnets use.

● IPv4

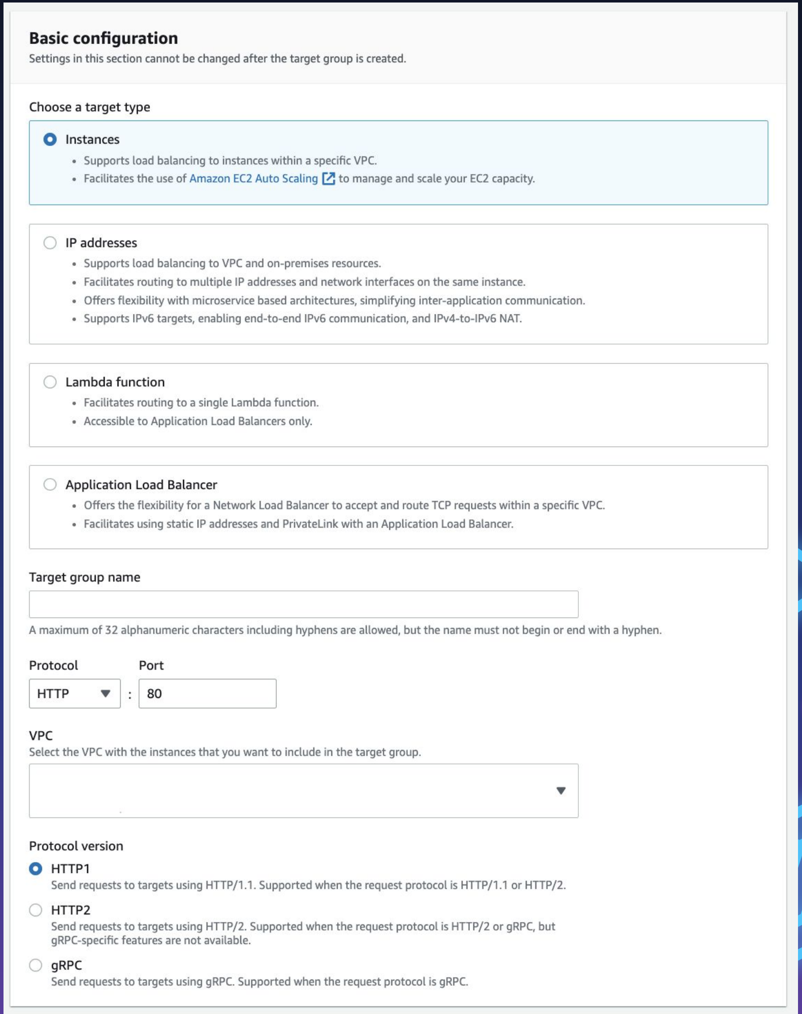
Recommended for internal load balancers.

○ Dualstack

Includes IPv4 and IPv6 addresses.

create a target group

- this will tell the load balancer how to access your application
    - port
    - protocol
    - VPC
    - instances
    - health checks



Basic configuration

Settings in this section cannot be changed after the target group is created.

Choose a target type

● Instances
- Supports load balancing to instances within a specific VPC.
- Facilitates the use of Amazon EC2 Auto Scaling to manage and scale your EC2 capacity.

○ IP addresses
- Supports load balancing to VPC and on-premises resources.
- Facilitates routing to multiple IP addresses and network interfaces on the same instance.
- Offers flexibility with microservice based architectures, simplifying inter-application communication.
- Supports IPv6 targets, enabling end-to-end IPv6 communication, and IPv4-to-IPv6 NAT.

○ Lambda function
- Facilitates routing to a single Lambda function.
- Accessible to Application Load Balancers only.

○ Application Load Balancer
- Offers the flexibility for a Network Load Balancer to accept and route TCP requests within a specific VPC.
- Facilitates using static IP addresses and PrivateLink with an Application Load Balancer.

Target group name

A maximum of 32 alphanumeric characters including hyphens are allowed, but the name must not begin or end with a hyphen.

Protocol: HTTP : Port: 80

VPC

Select the VPC with the instances that you want to include in the target group.

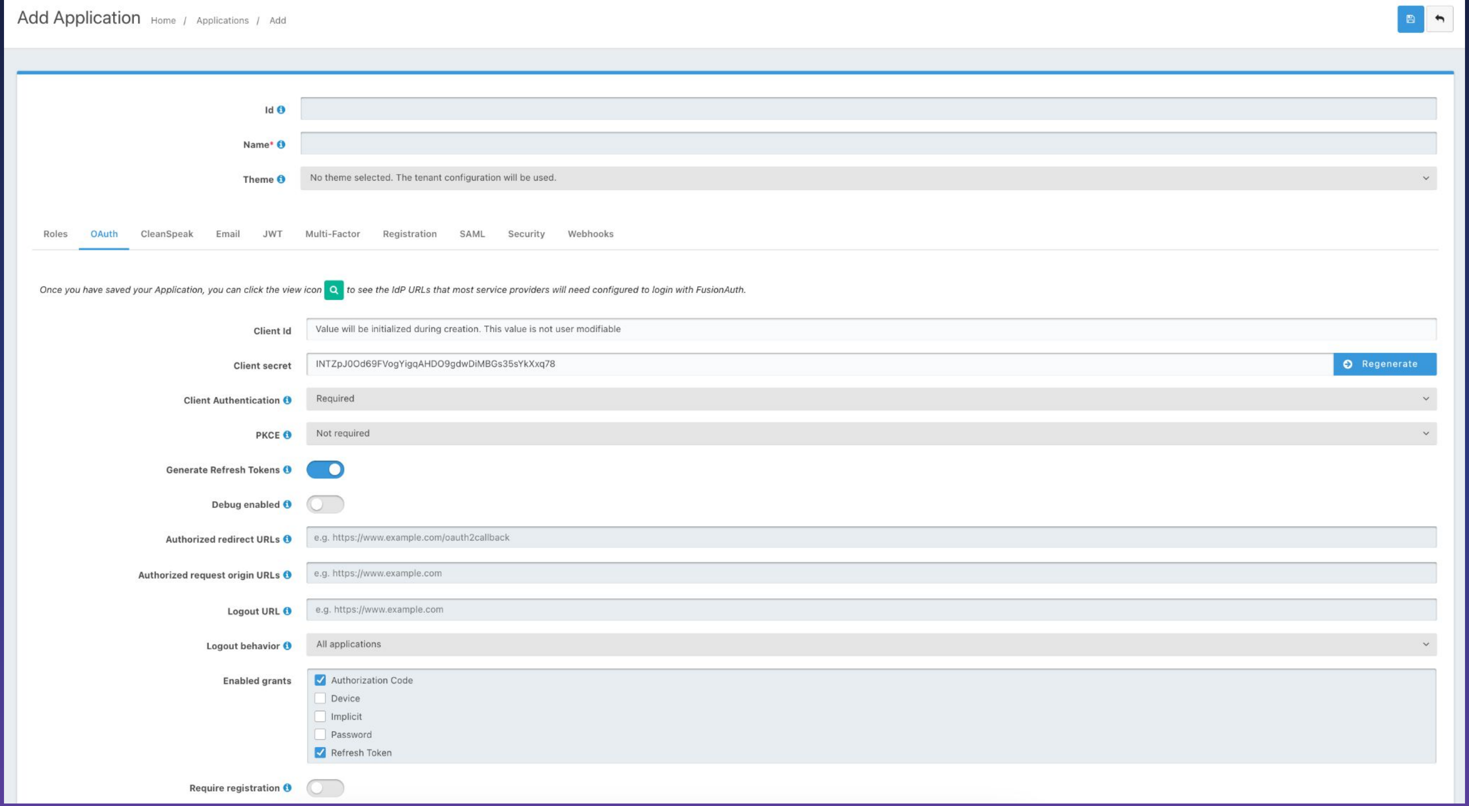
Protocol version

● HTTP1
Send requests to targets using HTTP/1.1. Supported when the request protocol is HTTP/1.1 or HTTP/2.

○ HTTP2
Send requests to targets using HTTP/2. Supported when the request protocol is HTTP/2 or gRPC, but gRPC-specific features are not available.

○ gRPC
Send requests to targets using gRPC. Supported when the request protocol is gRPC.

configure your application in your IdP



Add Application Home / Applications / Add

Id

Name*

Theme: No theme selected. The tenant configuration will be used.

Once you have saved your Application, you can click the view icon to see the IdP URLs that most service providers will need configured to login with FusionAuth.

Client Id: Value will be initialized during creation. This value is not user modifiable

Client secret: INTZpJ0Od69FVogYigqAHDO9gdwDiMBGs35sYkXxq78

Client Authentication: Required

PKCE: Not required

Generate Refresh Tokens: ●

Debug enabled: ○

Authorized redirect URLs: e.g. https://www.example.com/oauth2callback

Authorized request origin URLs: e.g. https://www.example.com

Logout URL: e.g. https://www.example.com

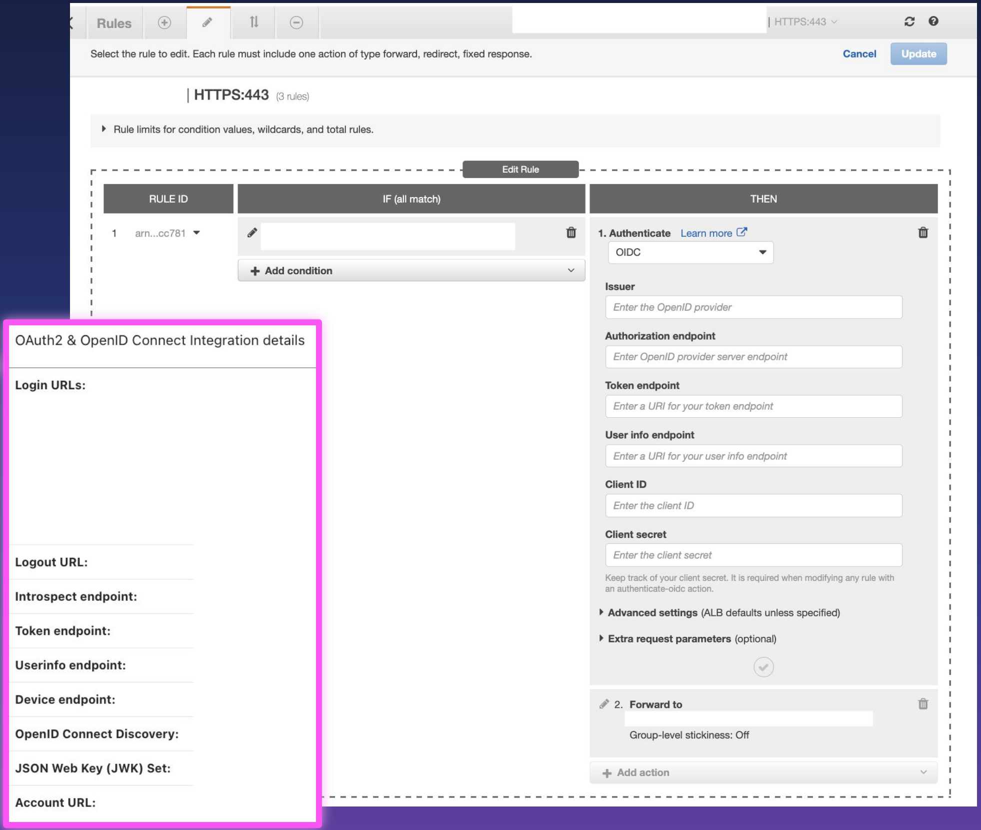
Logout behavior: All applications

Enabled grants:
- ☑ Authorization Code
- ☐ Device
- ☐ Implicit
- ☐ Password
- ☑ Refresh Token

Require registration: ○

configure your listeners

- port 80
  - forward to 443
- port 443
  - OIDC
  - issuer
  - authorization endpoint
  - token endpoint
  - user info endpoint
  - client ID
  - client secret
  - target group
    - this should be pointed at your application instance



Select the rule to edit. Each rule must include one action of type forward, redirect, fixed response.

HTTPS:443 (3 rules)

Rule limits for condition values, wildcards, and total rules.

RULE ID	IF (all match)	THEN
1 arn...cc781	+ Add condition	1. Authenticate Learn more

OIDC

Issuer: Enter the OpenID provider

Authorization endpoint: Enter OpenID provider server endpoint

Token endpoint: Enter a URI for your token endpoint

User info endpoint: Enter a URI for your user info endpoint

Client ID: Enter the client ID

Client secret: Enter the client secret

Keep track of your client secret. It is required when modifying any rule with an authenticate-oidc action.

Advanced settings (ALB defaults unless specified)

Extra request parameters (optional)

2. Forward to

Group-level stickiness: Off

OAuth2 & OpenID Connect Integration details

Login URLs:

Logout URL:

Introspect endpoint:

Token endpoint:

Userinfo endpoint:

Device endpoint:

OpenID Connect Discovery:

JSON Web Key (JWK) Set:

Account URL:

now we have

- an IdP configured with our application
- a load balancer configured to
  - redirect all port 80 requests to port 443
  - authenticate all port 443 requests with our IdP via OIDC
  - forward successfully authenticated users to our application in a target group

takeaways

- there are lots of ways to do this
  - this is an example of doing it with AWS and FusionAuth
  - you can do this with *just* AWS Cognito or similar cloud-native services
    - not as feature-rich, but still 100% doable
- you don’t need to spend a fortune on security to be secure
  - i like to promote FusionAuth 🤷🏻‍♀️

<3

hooray!

infosec.exchange/@shortstack
www.reconinfosec.com