

# Create EC2 and Load Balancer

This is a simple EC2 with an apache web server installed. The webserver is accessible behind a load balancer.

## Requirements

This demo requires the VPC with pub to priv cidr installed from service catalog.

- transit gateway from SC for ocelot (Mulesoft is not required for this doc but is a separate gateway)

## Ocelot

Setup ocelot to proxy traffic. Add the route name to Ocleot and route ownership group.

- route name: [nginx-mike.velocity.ag](#)
- route ownership: We use API DSS but your devloper group will be different.

Add a host name, such as `https://nginx-mike.fargate.722540083300.cloud.bayer.com`

- host `http://<application-name><.fargate.722540083300><.cloud.bayer.com>`
  - application-name can be whatever you want
  - 722540083300 is the AWS account number
  - always use .cloud.bayer.com

## Hosts



Ocelot takes HTTP requests and proxies to one or more [approved backend hosts](#). Here you define how to reach those hosts from Ocelot, either by hostname or IP. The host name requires a protocol and the port is optional. You should omit the path or make it '/'.

ADD

Change your hosts to match the format `<protocol>://<hostname or ip>/`

Examples:

- `http://myapp.cf.local/`
- `http://10.0.10.208:8080/`

SAVE

CANCEL

Ocelot takes HTTP requests and proxies to one or more approved backend hosts. Here you define how to reach those hosts from Ocelot, either by hostname or IP. The host name requires a protocol and the port is optional. You should omit the path or make it '/'.

## Ocelot Route key

A route key has been generated for your route: `nginx-mike.velocity.ag`

Please save this key, as it will not be available again. If you lose this key you will need to generate a new one.

Ocelot will send this key with all routed requests in the `ocelot-route-key` header. At your target, you can use this key to validate that the call did originate from a valid route definition in Ocelot.

Save key from ocelot `569d7b80-74d9-11ed-bf2e-9fefebdc0ebb`

## Install EC2

Install EC2. I use `Ubuntu 20.04` and attach the VPC SG.

The VPC SG is typically listed as `default` but verify, when there are multiple VPCs in the account. The inbound and outbound rules for this SG allow all traffic. This will allow the ALB SG created by the `fargate-infra` product to connect to our EC2.

## Apache

Install apache webserver in the EC2 bootstrap script

```
#!/bin/bash
```

```
sudo apt update -y
sudo apt install -y apache2 nano
sudo systemctl start apache2
sudo systemctl enable apache2 ## enable apache to load on boot
```

```
yum update -y
yum install httpd -y
systemctl start httpd
systemctl enable httpd
```

Connect to EC2 secure session manager. Make sure apache is running.

```
bash
sudo -u ubuntu -i

## sudo systemctl status apache2 ## check web server
## hostname -I ## copy and check addressess in browser
## curl -4 icanhazip.com ## Icanhazip tool, which should give you your public IP address
## sudo ufw status ## Verify http traffic is allowed
```

## Security Groups

Virtual firewall to allow EC2 and LB to proxy through Ocelot

1. Default VPC SG

2. Create SG to allow proxy Ocelot/Mulesoft/Akana

- Outbound rules All traffic All Protocol , All Ports and Destination `0.0.0.0/0`
- Inbound rules require port 443
  - Mulesoft CIDR `10.70.200.0/21`

- Ocelot CIDR 10.62.21.0/24
- Akana CIDR 192.168.0.0/24
- This SG ID = sg-0be806ad6114041cd

Open EC2 and add SG sg-0be806ad6114041cd to network interface. Now our EC2 has two SGs.

Details
Security
Networking
Storage

▼ Security details

IAM Role  
[mon-default-instance-role](#)

Security groups  
[sg-080d19b5f2b62a5ab \(default\)](#)  
[sg-067845c651b95e98e \(ec2 ocelot security group\)](#)

The inbound rules of the EC2 should look like this.

▼ Inbound rules

| <input type="text" value="Filter rules"/> <span>&lt; 1 &gt;</span> |                        |            |          |                                      |   |              |
|--|------------------------|------------|----------|--------------------------------------|---|--------------|
| Name   | Security group rule ID | Port range | Protocol | Source                               | Security groups                           | Description  |
| –  | sgr-036af30802a4f41ea  | All        | All      | <a href="#">sg-080d19b5f2b62a5ab</a> | <a href="#">default</a>                   | –            |
| –  | sgr-041cc279580ddfb87  | 80         | TCP      | <a href="#">sg-0be806ad6114041cd</a> | <a href="#">ec2 ocelot security group</a> | Allow Ocelot |

## Create Target Group

Add the "instances" for target type.


# Specify group details

Your load balancer routes requests to the targets in a target group and performs health checks on the targets.

## 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.

- Target group name apache-target-group
- Add VPC in account

Register target to complete this step.

# Load Balancer

Edit listener

Listeners

Network mapping

Security

Monitoring

Integrations

Attributes

Tags

Listeners (1)

A listener checks for connection requests on its port and protocol. Traffic received by the listener is routed according to its rules.

Q

Search

Protocol:Port

ARN

Security policy

Default SSL cert

HTTPS:443

ARN

ELBSecurityPolicy-2016-08

\*.fargate.722540083300.cloud...

Select "Manage Rules" to edit the Listener

Insert Rule

| RULE ID   | IF (all match)   | THEN   |
|---|--|--|
| <div>1</div> <div>A rule ID (ARN) is generated when you save your rule.</div> | <div>Host header...</div> <div><div>is</div><div>apache-mike.fargate.722540083300.cloud.bayer.com</div><div>×</div></div> <div><div>or</div><div>Value</div><div>×</div></div> | <div>1. Forward to...</div> <div>Target group : Weight (0-999)</div> <div><div>Select a target group</div><div>▼</div><div>apache-target-group</div></div> <div><div>1</div><div>×</div></div> |