Geeko WAF 自动添加IP至Block IPSet

导入包设置

1. 进入Lambda控制台。
2. 在左侧导航栏中选中Layers
3. 创建点击创建Layer，自定义名称如IPy，上传python.zip文件，选择x86_64，运行时选择python3.9，点击创建。

创建函数

1. Lambda控制台，左侧导航栏选择Functions
2. 点击create Function
3. 自定义函数名称，如WAF_autoupdate.
4. 运行时选择Python3.9
5. 架构选择x86_64
6. 其他默认即可，点击创建

上传代码

1. 进入函数页面
2. 上传WAF_Autoupdate.zip代码
3. 页面下拉至底端，在Layers部分，选择add a layer
4. 进入后，选中Custom layers。
5. 在下拉菜单中选择导入包设置中创建的Layer，即IPy，版本号选择1
6. 点击添加
7. 进入Lambda的Configuration界面，点击Permission
8. 在Lambda的IAM中加入AWSWAFFullAccess权限
9. 在Environment Variables界面中，按图示输入参数：

Environment variables (9) Edit

The environment variables below are encrypted at rest with the default Lambda service key.

Key	Value
Customer_blocklist_Name_v4	geeko-v4
Customer_blocklist_Name_v6	geeko-v6
Customer_blocklist_id_v4	d5dd5098-2847-4b1c-a038-5bcc01cd0ec3
Customer_blocklist_id_v6	01beda23-2223-4198-8fcf-a1ed7b360a58
Rule_Name	test-rate-based-limit
WebACL_Id	0a834434-cac8-4d1d-83ed-c9241e7a5f11
WebACL_Name	test-rate-limit
ip_range_v4	14.248.48.0/24,14.248.0.0/16
ip_range_v6	::1

- 其中若不需要ipv6，则可以不填写v6结尾的参数。如果后续需要ipv6，则可以添加此参数，并将lambda代码中被注释部分修改一下即可
- Customer_blocklist_Name_v4: 用户自建的永久block的ipset的名称
- Customer_blocklist_id_v4：用户自建的永久block的ipset的Id，在控制台上可以看到
- Rule_Name：自建的基于速率的rate-based rule的名称
- WebACL_Id：含有rate-based rule的WebACL的Id，控制台上可以看到
- WebACL_Name：含有rate-based rule的WebACL的名称
- ip_range_v4：需要提供的ip地址段，格式如图，以逗号分隔，中间不含空格
- 后续如需修改相应参数在此修改即可，不需要修改代码

10. 点击保存

定时启动Lambda

1. 进入CloudWatch控制台
2. 左侧导航栏选择Events中的Rules
3. 页面中选择Go to Amazon Eventbridge
4. 进入后，选择Create rule
5. 自定义名称，如WAF_auto
6. Rule type选择Schedule，点击下一步
7. Schedule pattern选择at regular rate那一个，rate选为4 minutes（此频率可以根据您的需求进行调整），点击下一步
8. Targate选择Lambda Function，Function选择上一步创建的函数，如WAF_autoupdate
9. 点击下一步至创建