Integrations Hand-on

Linode 环境准备

三台虚拟机： OS 皆为 Ubuntu 22.04

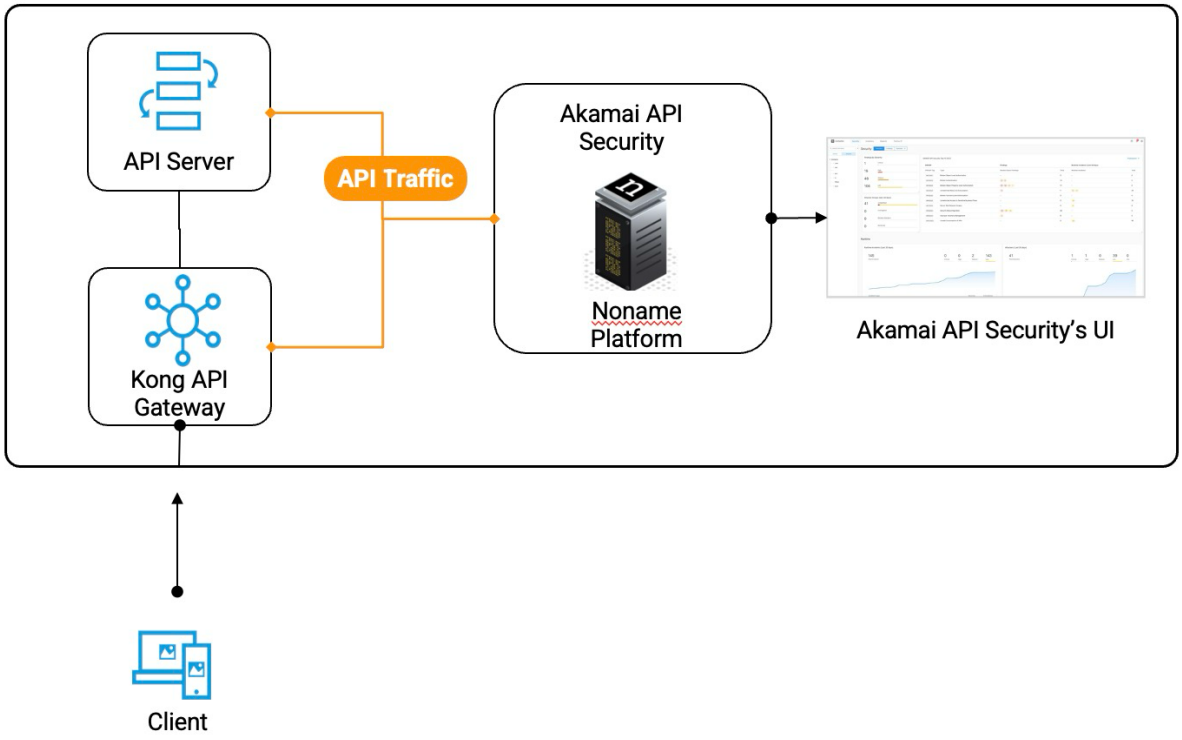
（OS 可使用 Ubuntu or Rocky， 但以下文件皆以 Ubuntu 指令为范例）

1. API Server （基本资源即可）
2. Kong API Gateway （基本资源即可）
3. Noname （建议 16GB 以上）

Linodes

Label ↑	Status ↓↑	Plan ↓↑
api_server	Running	Dedicated 4 GB
kong	Running	Dedicated 4 GB
Noname	Running	Linode 16 GB

架构



情境一：API 需通过 Kong API Gateway 访问

使用 Kong plugin integration 收集经过 Kong 的 API Traffic

情境二：客户直接访问 API Server

使用 Server Sensor integration 直接收集 API 主机上的 Traffic

Firewall 开通

Client -> API Server (22, 3000 ,8000)

Client -> Kong API Gateway (22 ,8000)

Client -> Noname (22 ,443)

Kong API Gateway -> API Server (3000 ,8000)

Kong API Gateway -> Noname (443)

API Server -> Noname (443)

API Server 部署 方法一 (CLI)

安装套件 Fastapi

```
apt updateapt-get install python3-pip
pip install fastapi uvicorn
```

编辑 API

```
vim main.py
```

```
# main.py
from fastapi import FastAPI

app = FastAPI()

@app.get("/api1")
async def root():
    return {"message": "Hello World"}

@app.get("/api2")
async def root():
    return {"api2": "ok"}
```

保存离开

Run FastAPI

```
uvicorn main:app --reload --host 0.0.0.0 --port 8000
```

验证服务 使用 Client

```
curl http://<api_server_ip>:8000/api1
```

API Server 部署 方法二 (GUI)

安装 Ubuntu GUI

```
apt-get update
apt-get install ubuntu-desktop
systemctl set-default graphical.target
```

安装 Mockoon API Server

```
snap install mockoon
reboot
```

GUI 点选 Mockoon

编辑 API 路由

Kong API Gateway 部署

确认 FW 规则

Kong Server 可以访问 Api Server

```
curl http://<api_server_IP>:8000/api1
```

安装 Kong

（参考 https://docs.konghq.com/gateway/3.4.x/install/linux/ubuntu/?install=oss#installation)

```
bash <(curl -sS https://get.konghq.com/install) -p kong -v 3.4.2
```

是否要分开 Control Plane and Data Plane

```
Switch to using Konnect? (y/n) n
```

设定 Kong 路由一

（参考 https://docs.konghq.com/gateway/3.4.x/get-started/services-and-routes/）

```
curl -i -X POST \
  --url http://localhost:8001/services/ \
  --data 'name=my-service1' \
  --data 'url=http: //<api_server_IP>: 8000/api1'
```

```
curl -i -X POST \
  --url http://localhost:8001/services/my-service1/routes \
  --data 'paths[]=/test1' \
  --data 'name=my-route1'
```

验证服务 使用 Client

```
curl http://<kong_ip>:8000/test1
```

设定 Kong 路由二

```
curl -i -X POST \
  --url http://localhost:8001/services/ \
  --data 'name=my-service2' \
  --data 'url=http://<api_server_IP>:8000/api2'
```

```
curl -i -X POST \
  --url http://localhost:8001/services/my-service2/routes \
  --data 'paths[]=/test2' \
  --data 'name=my-route2'
```

验证服务 使用 Client

```
curl http://<kong_ip>:8000/test2
```

Noname on-prem 部署

Resource: 8 core CPU , 32GB MEM , 256GB Disk

OS 前置作业

```
vim /etc/hosts
```

#::1 （在 ::1 前面加上#）

For RHEL 记得关闭 SELinux

```
vim /etc/selinux/config
```

SELINUX=disabled

下载安装包

```
wget https://files.nonamesecurity.com/RjHylulLtD-noname-active -O noname.active.v3.40.10-lts.tar
```

下载 Activation File（POC 可重复使用）

```
wget http://172.104.171.157/noname.activation_poc_202504
```

Client 下载 License File（POC 每个客户需独立申请）

```
http://172.104.171.157/Workshop-end_19Apr25.lic
```

解压缩

```
mkdir /opt/nonamecp noname.activation /opt/nonametar -zxvf noname.active.v3.40.10-lts.tar -C /opt/noname
```

开始安装

```
cd /opt/noname
./noname_installer.sh --full-installation --activation noname.activation_poc_202504 --
```

admin-email <帐号> --admin-password <密码>

Option:

--active 安装 Active Testing

--force 忽略系统资源检查（仅 Lab 使用，客户 POC 请达到系统要求资源）

安装完成检查

docker service ls

###有安装 Active Testing 需执行

docker service update noname_active_backend --env-add DATABASE_USE_SSL="false"

登录 UI 上传 License

Kong Integration

Noname UI > System > Integrations > Add Integration > Kong

下载 plugin file 並 upload 至 Kong Server

```
unzip noname-security-kong-policy.zip
luarocks install kong-plugin-nonamesecurity-3.2.4-1.all.rock
vim /etc/kong/kong.conf
```

找到并加上 plugins = bundled,nonamesecurity

重启 Kong

```
kong restart
```

启用 plugin

```
curl -X POST http://localhost:8001/plugins --data "name=nonamesecurity"
```

验证服务 使用 Client

```
curl http://<kong_ip>:8000/test1
curl http://<kong_ip>:8000/test2
```

查看 Noname UI

Server Sensor Integration

Noname UI > System > Integrations > Add Integration > Sensor

选择 Debain

下载 script file 解压缩后用文本编辑器打开文件

将 token 改为以下内容

cat > /tmp/noname_token << 'EOF'

machine us-central1-apt.pkg.dev/projects/noname-artifacts login _json_key_base64

password

ewogICJ0eXBlIjogInNlcnZpY2VfYWNjb3VudCIsCiAgInByb2plY3RfaWQiOiAibm9uYW1lLWFydGl
mYWN0cyIsCiAgInByaXZhdGVfa2V5X2lkIjogIjZhMmEwNGMzYjc0Y2M3ZDc5NjcyZjM1OTNjY
mE0MDIyYmYyYTQ1YmYiLAogICJwcml2YXRlX2tleSI6ICItLS0tLUJFR0lOIFBSSVZBVEUgS0VZLS0
tLS1cbk1JSUV2Z0lCQURBTkJna3Foa2lHOXcwQkFRUZBQVNDQktnd2dnU2tBZ0VBQW9JQkFR
RE1mL09CdFNWSmpBcUtcbmMwRUw1YVBiTlFHZmxjNFp5UWpQL2hhT1lFekdhaU51VVRsZzZK
OW1hVzJ2dFRUaDVBQzg0Kzlhck1kYSswV1Vcbm4zd3RONUEwMlhyMjFBZ2xjdjQ2ZkNIZmR0d
ExlcEFpbzRwenVWZ0pBTTEvUDFDYzBJRXAwVXZ5UTF3VUtMZXVcbms5anhyWXZzRTgwTzlVT
0F0bndZV2ZUMkpGcFNXTXlvdVh0RGlpN1NKTEphblZUcSt5cncxQldBbEo5d2V1MERcbmszTjZv
SE1FNFRIT1ltTFN2ckxJbElPTTAxMWJzcm12TlpadXRybGRFMWJTU2wybWI5QXpHa2xwUk80Y0
kzZEFcbjZiTWh3N0VRY3ZZSlp2ZHVqQmYzV1VGNjZmNU15U1JUVkViUlp2YytjODRHa0p0THAz
RG42Y0FSb0pxZWZucDFcbk52ckNydHRUQWdNQkFBRUNnZ0VBV1FxWFJORlNhSGtrOTg2aU1t
TnRXcHQwVnYrYnkwZG80bU5nZitLM1d5ZElcbndSWjF0TmZrMitzbzJybzhTZVVFV3pxbDgrWE
tNYmU1dnJKSWw4bWJMUi9LdGlGeFlKajIwQkZQa0dUSXJDY2xcbkhDZjROMVlxMmxTeXBXYTB
tMnIvNm16YXRDUVhPaUhXaEptVWV2cVZNNWdWajU5RUlvMlNiK252U3FOZWZuNnJcblkzb0hr
aFY1K1Y0YU5pajkxRk5WZ2VxVXpRK240a0lNSVUydG1rbDdnWkZTTFRiYTRQM1pKVlh3bCtwMF
BsWWpcbmJlcUc3R2pGdGxQTDVzeUNRcWxZY1l0Q0RCSTNCVm0yMnpjSitIbFhtdndUdDJxcGd
XZm9kTlhtR29CVVRYbXJcblBNTE04N00rVGQreDdoSDRyY0Y4ck9ZekNITGxqa3Z5Ukw5WFJsY
kNuUUtCZ1FEcVFJeFN2MEFBUnNIWU4vb09cbkVCaFNyQkNmWTZMSnhoQzIrc3BUL0V2WDNs
akNOWk9JRTBySnhvUWZBd3dDeUZFZFJ1NkxJclRZUDZGSFc3YzZcbko0RHk3YkFOK09CTTM1Tl
F1aFl0V1MwT1crZXZEVCthUkpNdXdvanZ3VEl5RUIxUG9BcWRDbjU3SDZSYjVROWpcbnFqS2V3
Ny8vY1F3S1phLzVDU2J4eGpqdlBRS0JnUURmZkVnaWJjRGt6bkllMlRYay8zZ3pQNlVGRmt3ZUh
BdTJcbmpKeUNlbm9BRGZkMldocGNSYWsrUTlZU01TMk1lZEdWYU1SUHd3WHBMZWM5d1E5RHl
DMmFvMURJeVorak9rVFJcbkdTaGpwZEFiLzUrQmR0NVh4cTNQSWplMWJ5NUFlc25qSVp0MCt

md0FkOWRoOTN2eHh6dWdCUTFEeHZBTElaYkdcbmZxaThMbkdkendLQmdDcDk2dlpYbXRSeU
FWZW5UOFNkUnN3NFNndWZsemZpcGNXTFdJaERpNzVzcHMwVHl0ODBcbklLelpSYjRhWDJOa
DB0VHN1UGlrSXduWEFQcGd1S3ZqM3BBeWJXMGJhQlNKdWI4N0FpclhzaTNZaHo1SnVPL1NcbjR
KL3F5Y3d6ejFjbGRTdXNiRUJ3VnVFZXJFRy9HenVJOUFXNG5HdEpwbFY4Zko4LytjQ0JwSEpCQ
W9HQkFMbTRcbkVFSDhMeloxa1VTdGdTb2lXd05MdXBqMXBKL0k4cnJyTXcwRE51dGdyK3I4a
WJkd2RHcTMrT2Y1VWo3UTVTaGlcbmZmRkN3bUM2c3RZUTM5eVdONyt3Ukd2M2tlQVZoaWE
xMzJpZjNNZkhzQWZvZm5FM1BDMUwzM1IweGx6ekxWN0JcbkNPMDI2UUtDbE14bTIrdGFnK1JC
RG5ycWtZVE9qdnU1T2dQWUhMYmxBb0dCQUpEUEZoUEhKRVQ5aVpmcTNWWUNcbkNFZ1Jy
Wjh5TkQyblpBdVBLb0l6enJjRWsyS3llUis0MndKUFJaWGdTb0NDc2VVYVFuWGdkQ015S21pakp
GQ0Jcbk1xSkhtVEdlZEh1dEszcGwwZGk5WXZUOVhCK2JhZmNGZXR0UTFrU3U3K0lKNmU4WV
VMOWZGN2E0d2d4OTN2VTNcbml5Z25NUjdRbjZpTFNIdnU5eEd1MUZqcFxuLS0tLS1FTkQgUFJ
JVkFURSBLRVktLS0tLVxuIiwKICAiY2xpZW50X2VtYWlsIjogImRlbW8xLXNhYXNAbm9uYW1lLWFy
dGlmYWN0cy5pYW0uZ3NlcnZpY2VhY2NvdW50LmNvbSIsCiAgImNsaWVudF9pZCI6ICIxMDQxN
jQ0MjcxMTI1OTk4MjcyODciLAogICJhdXRoX3VyaSI6ICJodHRwczovL2FjY291bnRzLmdvb2dsZS
5jb20vby9vYXV0aDIvYXV0aCIsCiAgInRva2VuX3VyaSI6ICJodHRwczovL29hdXRoMi5nb29nbGV
hcGlzLmNvbS90b2tlbiIsCiAgImF1dGhfcHJvdmlkZXJfeDUwOV9jZXJ0X3VybCI6ICJodHRwczovL
3d3dy5nb29nbGVhcGlzLmNvbS9vYXV0aDIvdjEvY2VydHMiLAogICJjbGllbnRfeDUwOV9jZXJ0X
3VybCI6ICJodHRwczovL3d3dy5nb29nbGVhcGlzLmNvbS9yb2JvdC92MS9tZXRhZGF0YS94NTA
5L2RlbW8xLXNhYXMlNDBub25hbWUtYXJ0aWZhY3RzLmlhbS5nc2VydmljZWFjY291bnQuY29tIi
wKICAidW5pdmVyc2VfZG9tYWluIjogImdvb2dsZWFwaXMuY29tIgp9Cg==
EOF

Upload 至 API Server 并安装
chmod +x install_script.sh
./install_script.sh

检查 Sensor 状态
systemctl status noname-sensor

編輯 API 增加 api3 與 API4

```
vim main.py
```

```
# main.py
from fastapi import FastAPI

app = FastAPI()

@app.get("/api3")
async def root():
    return {"api3": "加油"}

@app.get("/api4")
async def root():
    return {"api4": "快下課了"}
```

保存离开

Run FastAPI

```
uvicorn main:app --reload --host 0.0.0.0 --port 8000
```

验证服务 使用 Client

```
curl http://<api_server_ip>:8000/api3
curl http://<api_server_ip>:8000/api4
```

查看 Noname UI