F5 NGINX Plus Ingress Controller as an API Gateway for Kubernetes

Presented by:

Brian Gautreau (SE - US Army)

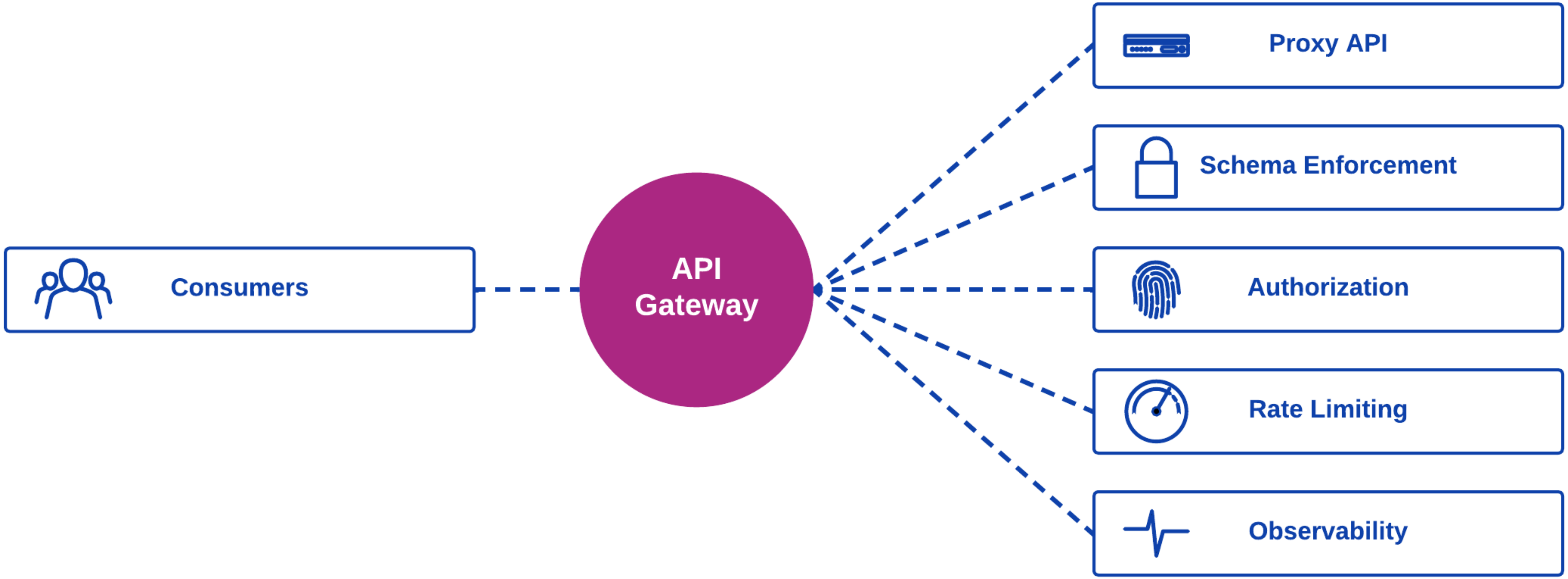
MARCH 19-20, 2024

Lab Documentation

https://clouddocs.f5.com/training/community/nginx/html/class11/class11.html

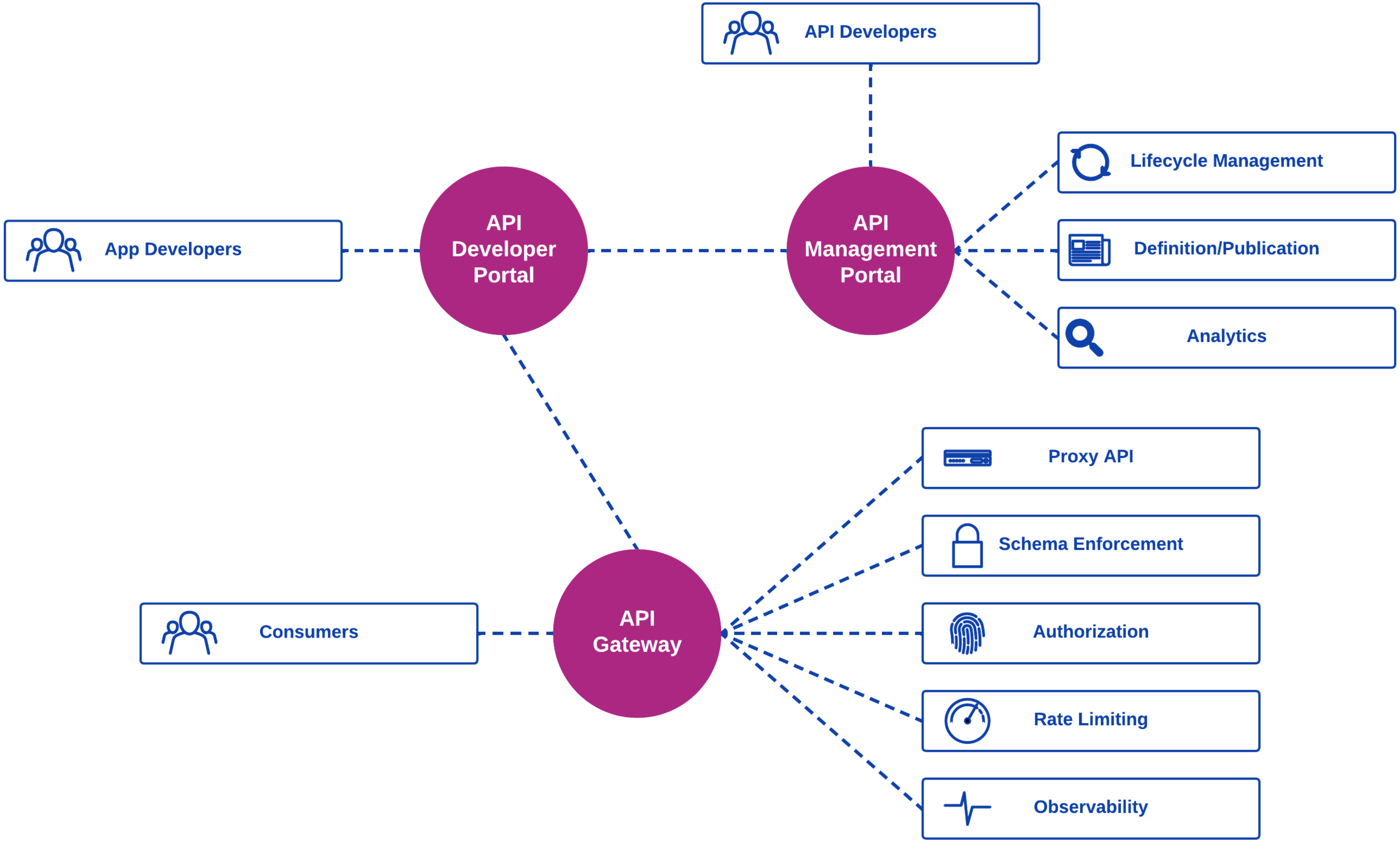
What is an Application Programming Interface (API) Gateway?

A collection of common performance, security, and visibility services required to put an API in production



What about *API Management?*

Every API needs an API Gateway, few require API Management



Common API Formats?

REST APIs are still the most common

API Format	Transport	Data Format	Features
SOAP	Many	XML	RPC, hierarchically structured, security built in, transport independent, relatively complex
REST	**HTTP**	**JSON, XML**	**HTTP methods (eg GET, POST) to query and update data, multiple URLs represent the resources, schemas supported with OpenAPI spec**
gRPC	HTTP/2	Protocol Buffers	RPC, strongly typed, schema based, high performance, microservices
graphQL	HTTP, WebSockets	JSON	Query language for APIs, schemas define static types, get exactly what you ask for, single endpoint (/graphql), no versioning, client-server

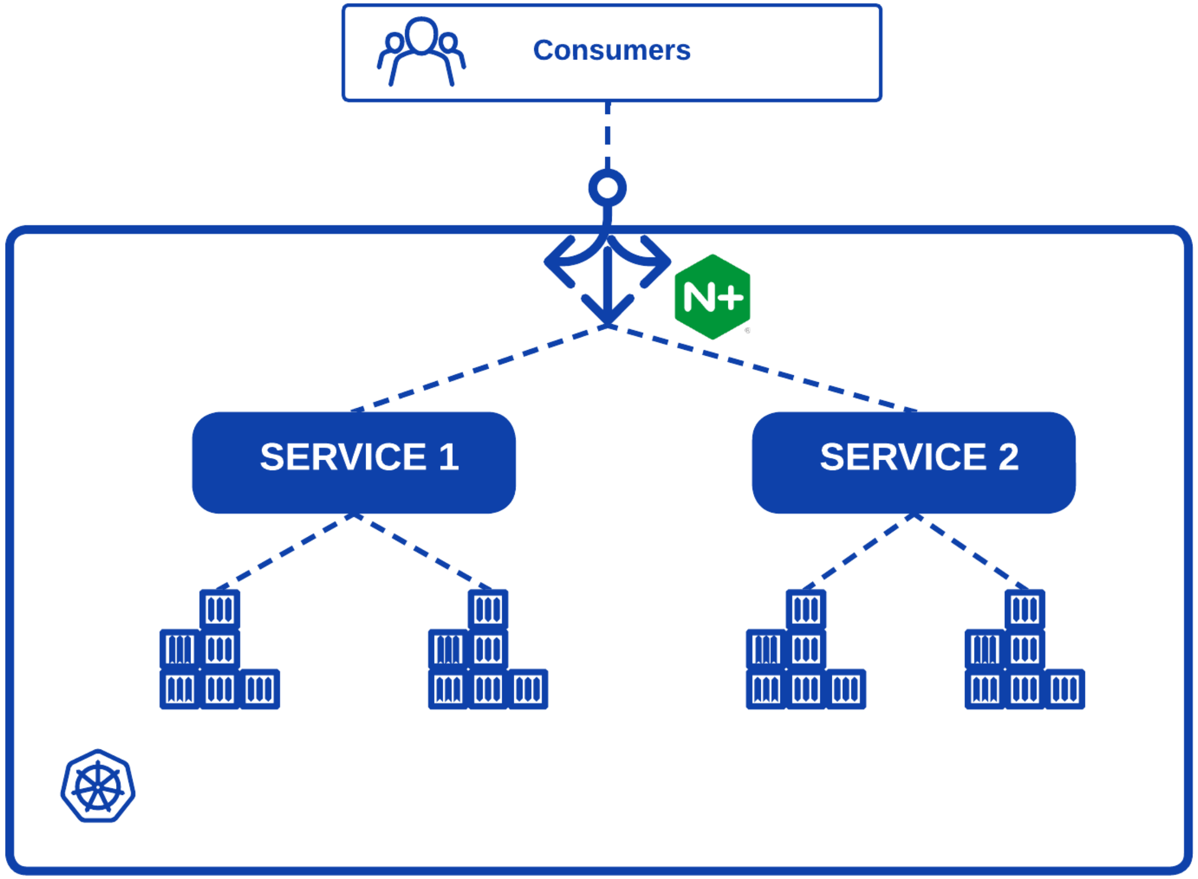
API Gateway Deployment Models

NGINX Plus Ingress is the natural home for many API Gateway use cases

- Standalone physical or virtual proxy such as F5 BIG-IP or NGINX Plus

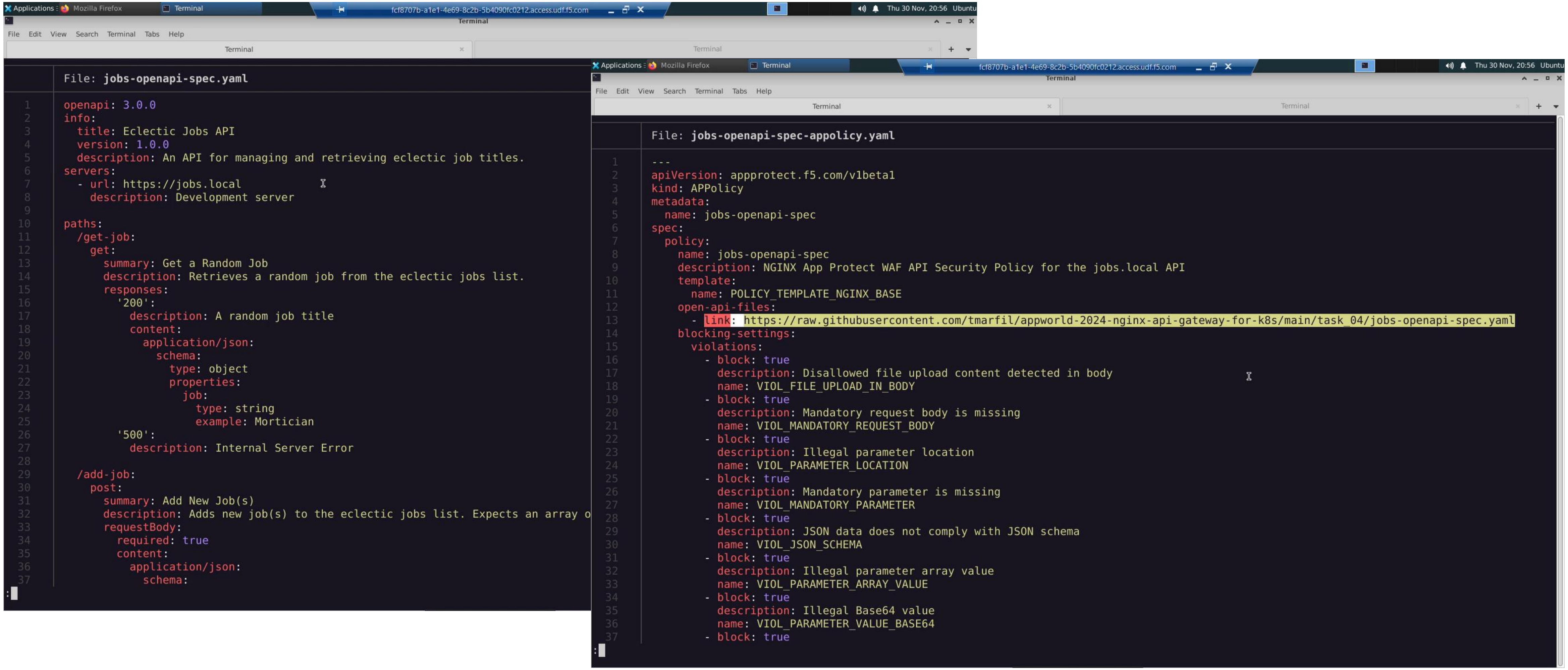
- SAAS such as F5 Distributed Cloud (XC)

- Kubernetes-hosted alongside the microservices that make up an API



OpenAPI Schema Enforcement

Prevent abuse. Ensure all API requests conform to a predefined schema.


Applications
Mozilla Firefox
Terminal
fcf8707b-a1e1-4e69-8c2b-5b4090fc0212.access.udf.f5.com
Thu 30 Nov, 20:56
Ubuntu
File Edit View Search Terminal Tabs Help

File: **jobs-openapi-spec.yaml**

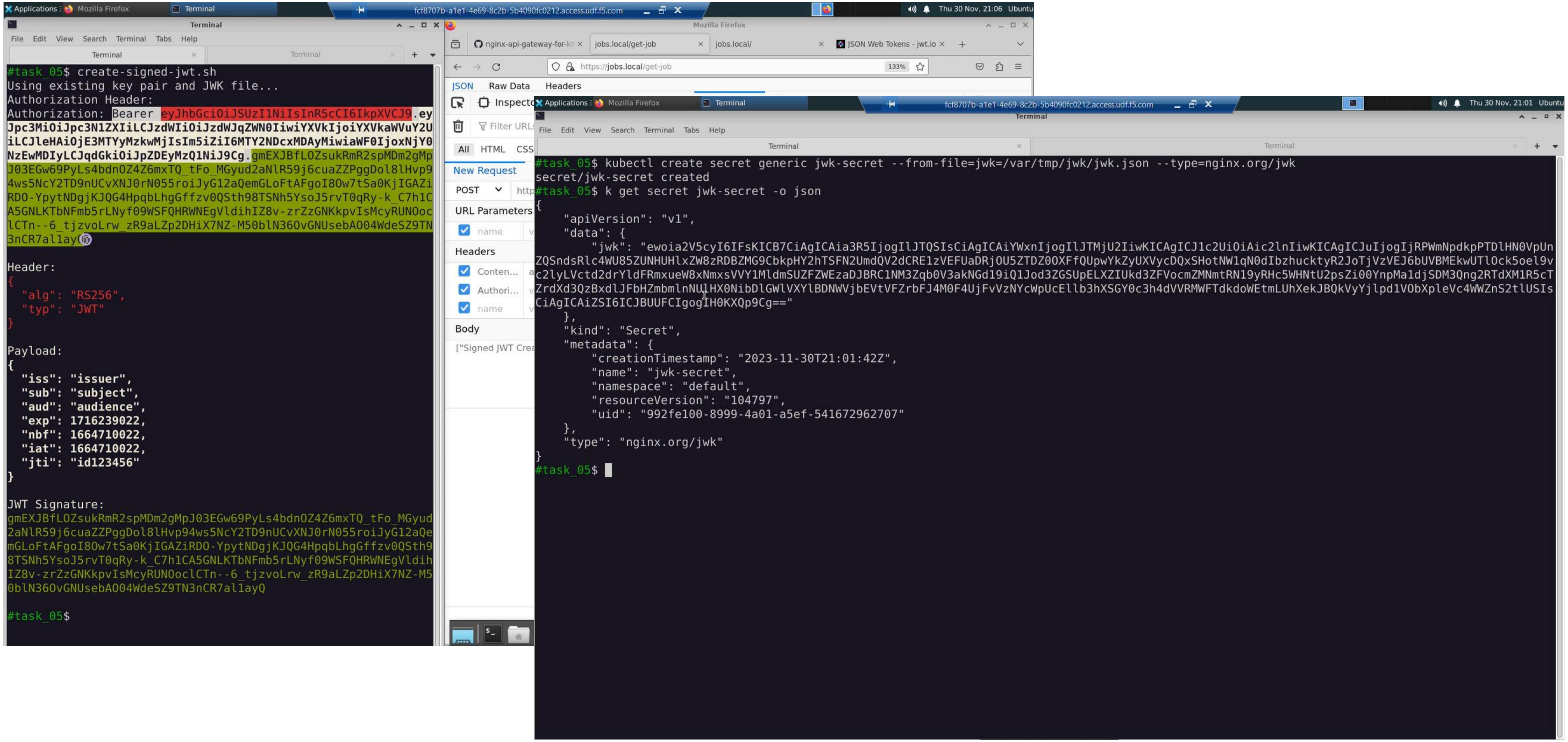
```
openapi: 3.0.0
info:
  title: Eclectic Jobs API
  version: 1.0.0
  description: An API for managing and retrieving eclectic job titles.
servers:
  - url: https://jobs.local
    description: Development server

paths:
  /get-job:
    get:
      summary: Get a Random Job
      description: Retrieves a random job from the eclectic jobs list.
      responses:
        '200':
          description: A random job title
          content:
            application/json:
              schema:
                type: object
                properties:
                  job:
                    type: string
                    example: Mortician
        '500':
          description: Internal Server Error

  /add-job:
    post:
      summary: Add New Job(s)
      description: Adds new job(s) to the eclectic jobs list. Expects an array o
      requestBody:
        required: true
        content:
          application/json:
            schema:
```

File: **jobs-openapi-spec-appolicy.yaml**

```
---
apiVersion: appprotect.f5.com/v1beta1
kind: APPolicy
metadata:
  name: jobs-openapi-spec
spec:
  policy:
    name: jobs-openapi-spec
    description: NGINX App Protect WAF API Security Policy for the jobs.local API
    template:
      name: POLICY_TEMPLATE_NGINX_BASE
    open-api-files:
      - link: https://raw.githubusercontent.com/tmarfil/appworld-2024-nginx-api-gateway-for-k8s/main/task_04/jobs-openapi-spec.yaml
    blocking-settings:
      violations:
        - block: true
          description: Disallowed file upload content detected in body
          name: VIOL_FILE_UPLOAD_IN_BODY
        - block: true
          description: Mandatory request body is missing
          name: VIOL_MANDATORY_REQUEST_BODY
        - block: true
          description: Illegal parameter location
          name: VIOL_PARAMETER_LOCATION
        - block: true
          description: Mandatory parameter is missing
          name: VIOL_MANDATORY_PARAMETER
        - block: true
          description: JSON data does not comply with JSON schema
          name: VIOL_JSON_SCHEMA
        - block: true
          description: Illegal parameter array value
          name: VIOL_PARAMETER_ARRAY_VALUE
        - block: true
          description: Illegal Base64 value
          name: VIOL_PARAMETER_VALUE_BASE64
        - block: true
```

Authorization

Authorize users at the API Gateway by validating JSON Web Tokens.

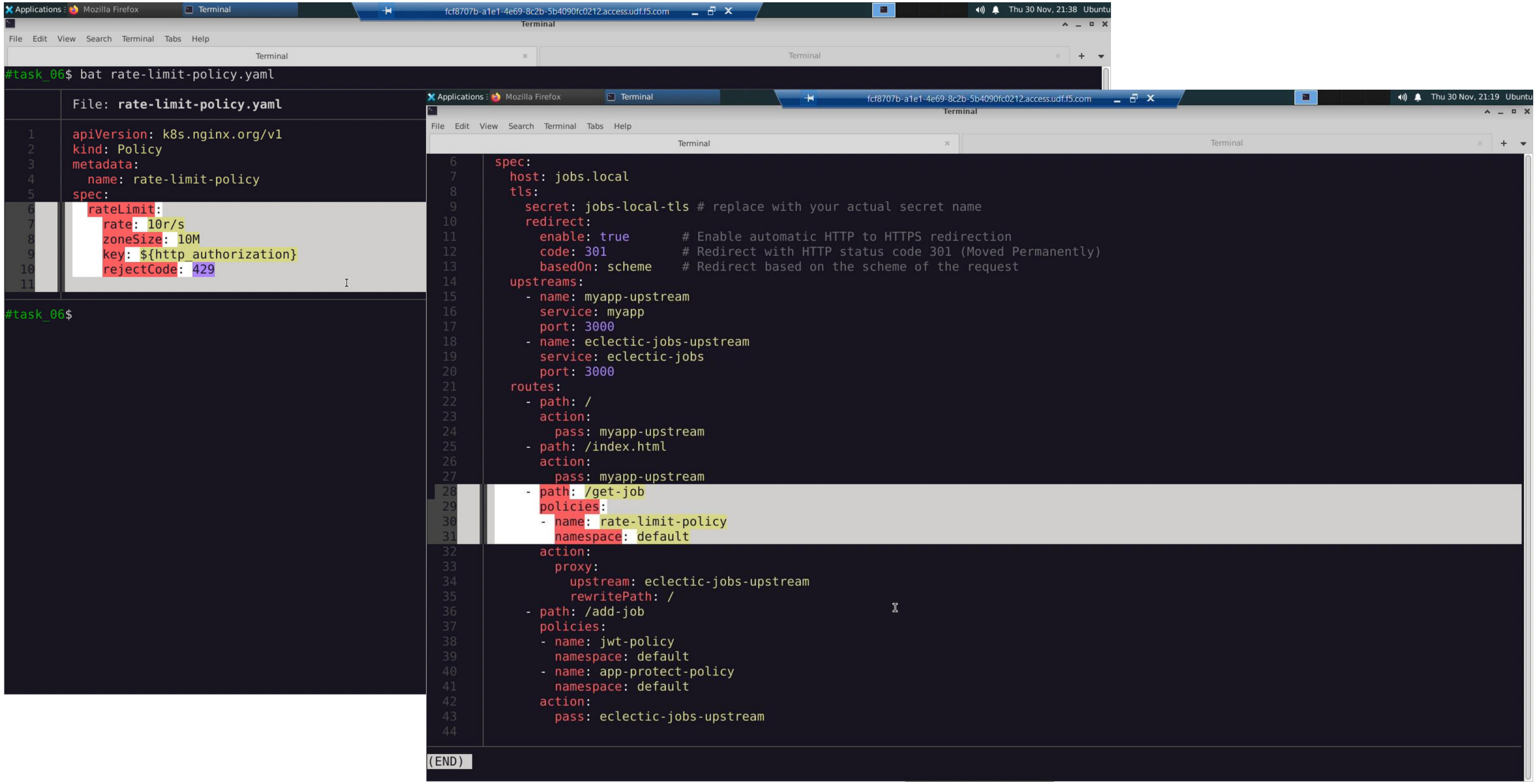


```
#task_05$ create-signed-jwt.sh
Using existing key pair and JWK file...
Authorization Header:
Authorization: Bearer eyJhbGciOiJSUzI1NiIsInR5cCI6IkpXVCJ9.ey
Jpc3MiOiJpc3N1ZXIiLCJzdWIiOiJzdWJqZWN0IiwiYXVkIjoiYXVkaWVuY2U
iLCJleHAiOjE3MTYyMzkwMjIsIm5iZiI6MTY2NDcxMDAyMiwiaWF0IjoxNjY0
NzEwMDIyLCJqdGkiOiJpZDEyMzQ1NiJ9Cg.gmEXJBfLOZsukRmR2spMDm2gMp
J03EGw69PyLs4bdnOZ4Z6mxTQ_tFo_MGyud2aNlR59j6cuaZZPggDol8lHvp9
4ws5NcY2TD9nUCvXNJ0rN055roiJyG12aQemGLoFtAFgoI8Ow7tSa0KjIGAZi
RDO-YpytNDgjKJQG4HpqbLhgGffzv0QSth98TSNh5YsoJ5rvT0qRy-k_C7h1C
A5GNLKTbNFmb5rLNyf09WSFQHRWNEgVldihIZ8v-zrZzGNKkpvIsMcyRUNOoc
lCTn--6_tjzvoLrw_zR9aLZp2DHiX7NZ-M50blN36OvGNUsebAO04WdeSZ9TN
3nCR7allayQ

Header:
{
  "alg": "RS256",
  "typ": "JWT"
}

Payload:
{
  "iss": "issuer",
  "sub": "subject",
  "aud": "audience",
  "exp": 1716239022,
  "nbf": 1664710022,
  "iat": 1664710022,
  "jti": "id123456"
}

JWT Signature:
gmEXJBfLOZsukRmR2spMDm2gMpJ03EGw69PyLs4bdnOZ4Z6mxTQ_tFo_MGyud
2aNlR59j6cuaZZPggDol8lHvp94ws5NcY2TD9nUCvXNJ0rN055roiJyG12aQe
mGLoFtAFgoI8Ow7tSa0KjIGAZiRDO-YpytNDgjKJQG4HpqbLhgGffzv0QSth9
8TSNh5YsoJ5rvT0qRy-k_C7h1CA5GNLKTbNFmb5rLNyf09WSFQHRWNEgVldih
IZ8v-zrZzGNKkpvIsMcyRUNOoclCTn--6_tjzvoLrw_zR9aLZp2DHiX7NZ-M5
0blN36OvGNUsebAO04WdeSZ9TN3nCR7allayQ

#task_05$
```

```
#task_05$ kubectl create secret generic jwk-secret --from-file=jwk=/var/tmp/jwk/jwk.json --type=nginx.org/jwk
secret/jwk-secret created
#task_05$ k get secret jwk-secret -o json
{
    "apiVersion": "v1",
    "data": {
        "jwk": "ewoia2V5cyI6IFsKICB7CiAgICAia3R5IjogIlJTQSIsCiAgICAiYWxnIjogIlJTMjU2IiwKICAgICJ1c2UiOiAic2lnIiwKICAgICJuIjogIjRPWmNpdkpPTDlHN0VpUn
ZQSndsRlc4WU85ZUNHUHlxZW8zRDBZMG9CbkpHY2hTSFN2UmdQV2dCRE1zVEFUaDRjOU5ZTDZ0OXFfQUpwYkZyUXVycDQxSHotNW1qN0dIbzhucktyR2JoTjVzVEJ6bUVBMEkwUTlOck5oel9v
c2lyLVctd2drYldFRmxueW8xNmxsVVY1MldmSUZFZWEzaDJBRC1NM3Zqb0V3akNGd19iQ1Jod3ZGSUpELXZIUkd3ZFVocmZMNmtRN19yRHc5WHNtU2psZi00YnpMa1djSDM3Qng2RTdXM1R5cT
ZrdXd3QzBxdlJFbHZmbmlnNUlHX0NibDlGWlVXYlBDNWVjbEVtVFZrbFJ4M0F4UjFvVzNYcWpUcEllb3hXSGY0c3h4dVVRMWFTdkdoWEtmLUhXekJBQkVyYjlpd1VObXpleVc4WWZnS2tlUSIs
CiAgICAiZSI6ICJBUUFCIgogIH0KXQp9Cg=="
    },
    "kind": "Secret",
    "metadata": {
        "creationTimestamp": "2023-11-30T21:01:42Z",
        "name": "jwk-secret",
        "namespace": "default",
        "resourceVersion": "104797",
        "uid": "992fe100-8999-4a01-a5ef-541672962707"
    },
    "type": "nginx.org/jwk"
}
#task_05$
```

Rate Limiting

Ensure fair use of an API by rate limiting requests per user.



```
#task_06$ bat rate-limit-policy.yaml
       File: rate-limit-policy.yaml

apiVersion: k8s.nginx.org/v1
kind: Policy
metadata:
  name: rate-limit-policy
spec:
  rateLimit:
    rate: 10r/s
    zoneSize: 10M
    key: ${http_authorization}
    rejectCode: 429


#task_06$
```

```
spec:
  host: jobs.local
  tls:
    secret: jobs-local-tls # replace with your actual secret name
    redirect:
      enable: true       # Enable automatic HTTP to HTTPS redirection
      code: 301          # Redirect with HTTP status code 301 (Moved Permanently)
      basedOn: scheme    # Redirect based on the scheme of the request
  upstreams:
    - name: myapp-upstream
      service: myapp
      port: 3000
    - name: eclectic-jobs-upstream
      service: eclectic-jobs
      port: 3000
  routes:
    - path: /
      action:
        pass: myapp-upstream
    - path: /index.html
      action:
        pass: myapp-upstream
    - path: /get-job
      policies:
      - name: rate-limit-policy
        namespace: default
      action:
        proxy:
          upstream: eclectic-jobs-upstream
          rewritePath: /
    - path: /add-job
      policies:
      - name: jwt-policy
        namespace: default
      - name: app-protect-policy
        namespace: default
      action:
        pass: eclectic-jobs-upstream

(END)
```

ENOUGH TALK
SHOW ME A DEMO!!!
memegenerator.net

App
World
2024