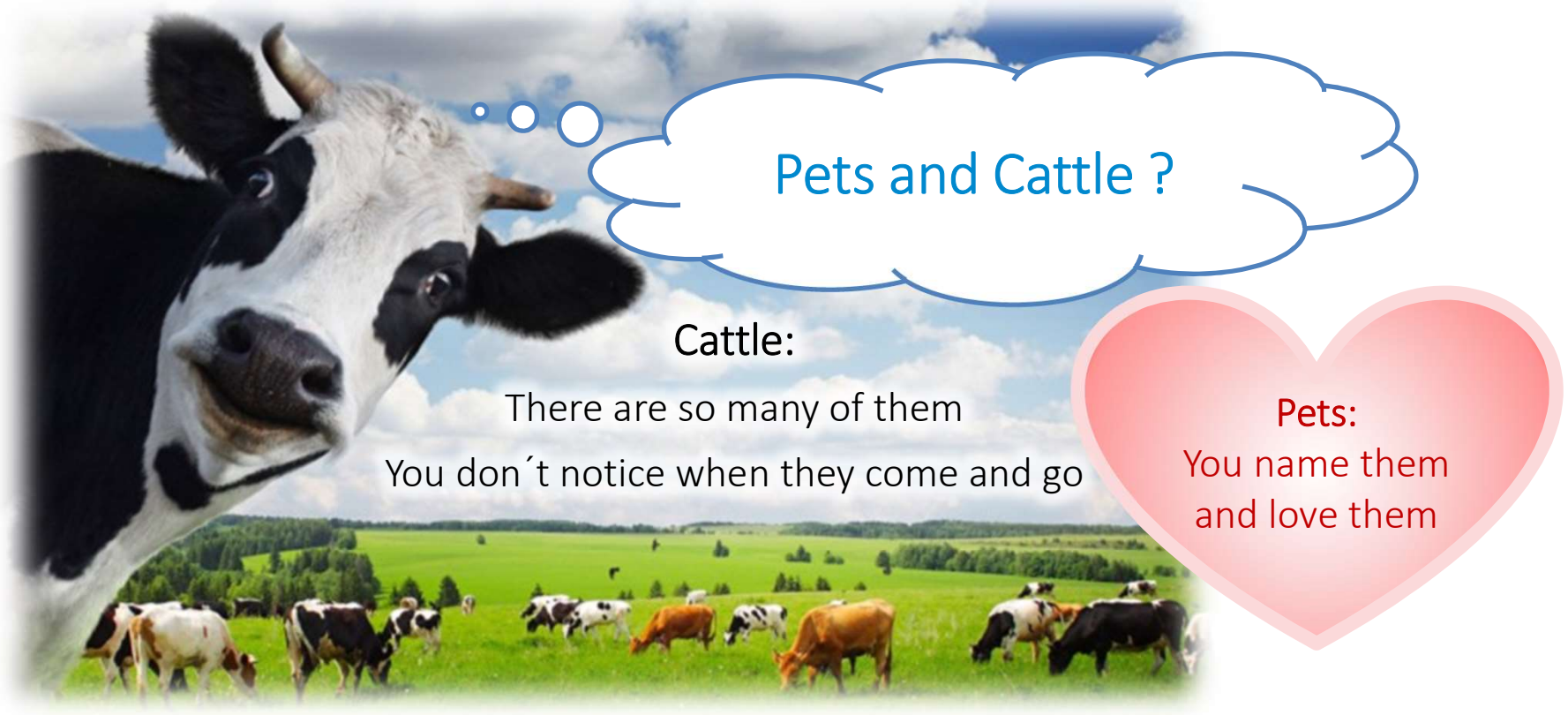

Barracuda
TECHSUMMIT19
BARRACUDA TECHNICAL SUMMIT
Barracuda CloudGen Firewall
CloudGen Firewall Firmware 8.x
Azure CGF Automation

# What Is Infrastructure as Code (IaC) ?

# What Is Infrastructure as Code (IaC) ?



# How to Manage a Herd of Cattle ?


Zero downtime

Blue-Green Deployment

Disaster recovery all the time

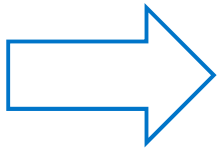
# CGF Automation Focus

- Continuously expanding the REST API
- Automate Firewall ruleset
- Automate VPN connectivity
  - Auto VPN
  - Virtual WAN
- Lifecycle management API

# Two Simple Steps

## Deploy CGF in Azure (+ Rest API)

- Challenges
  - Cost optimization
  - Process automation
  - Public cloud - Zero Touch CGFs
  - …

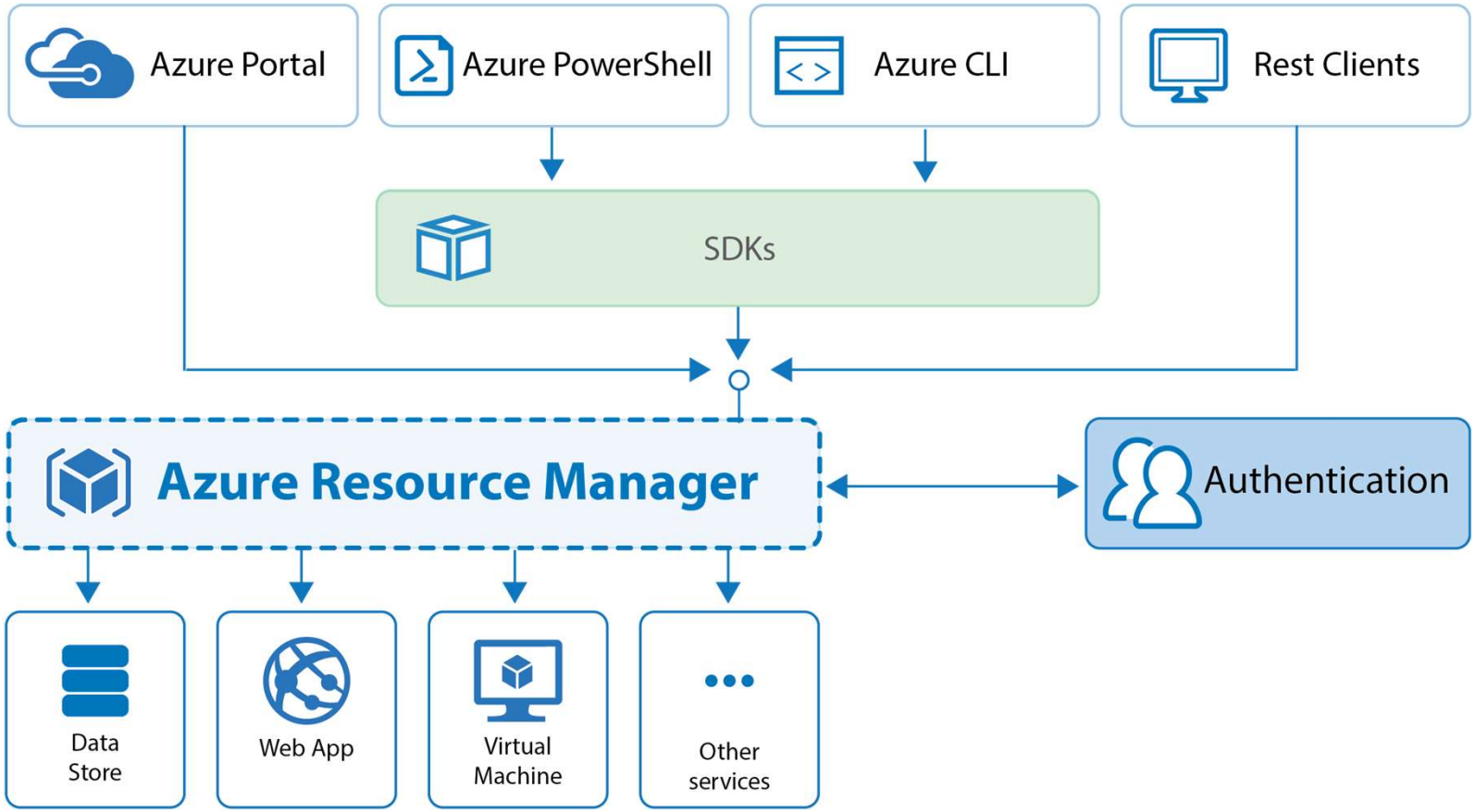


- CGF Rest API
  - Create Rest user
  - Create network objects
  - Create access rules
  - Create AutoVPN
  - …..

# Cooking Instructions

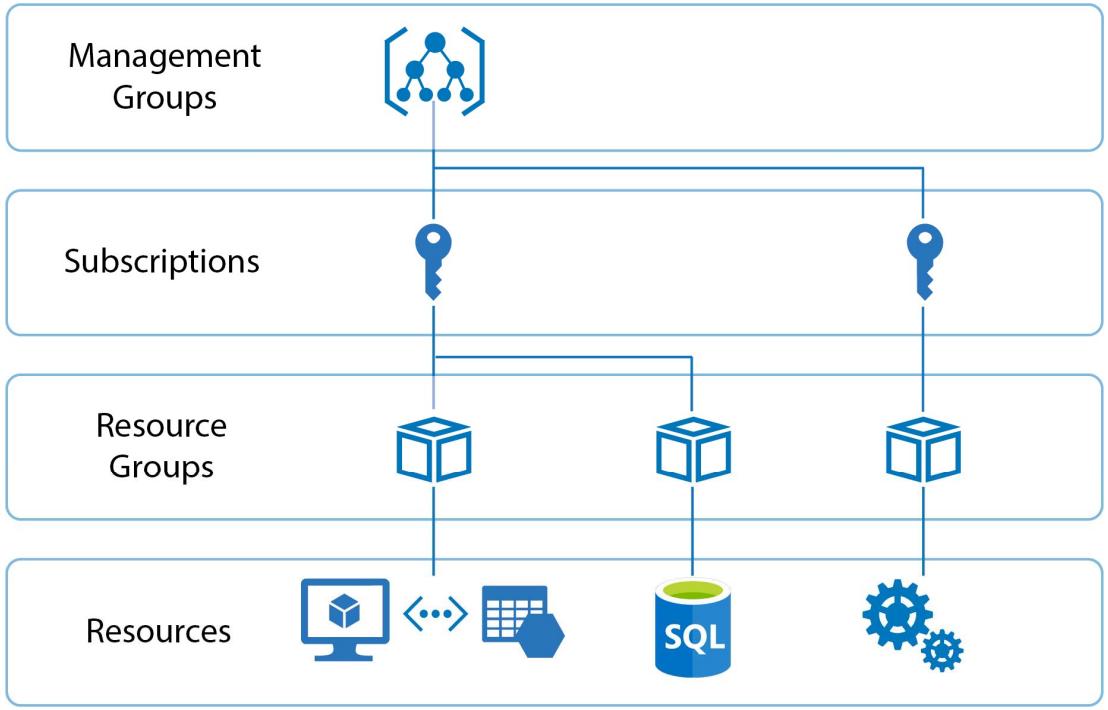
- Azure
  - ARM template deployment
  - Cloud Shell bash/PowerShell
  - Ansible
- CGF
  - GitHub/Barracuda Networks Deployment
  - Rest API
  - AutoVPN

# Basics - Azure Resource Manager



# Azure Scopes

Azure provides four levels of scope:



# ARM Template Deployment

- Benefits
  - Human-readable, file based in JSON (JavaScript Object Notation)
  - The same template usable with different parameters
  - Automation script
- Configuration Tools
  - Azure portal
  - Azure CLI
  - PowerShell
  - Visual studio

# ARM Template JSON

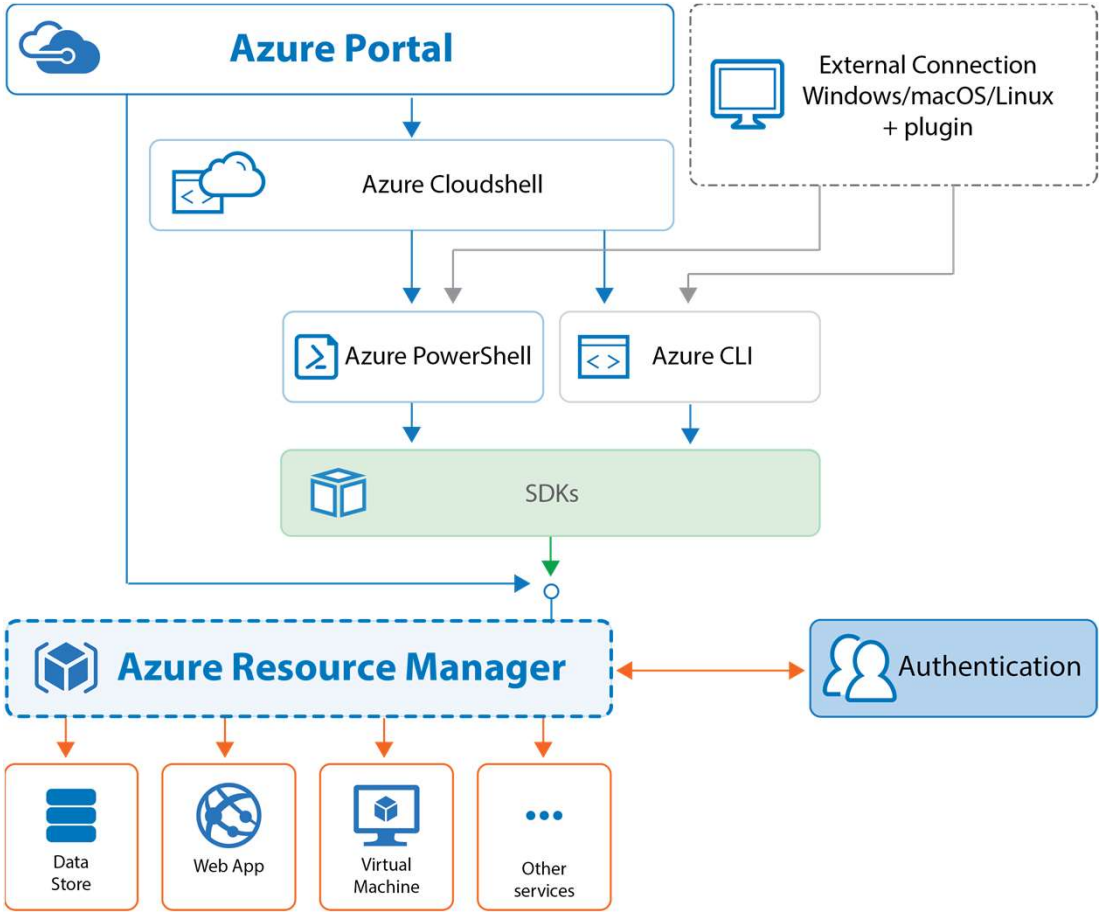
Every JSON file contains four main sections:

- Parameters
- Variables
- Resources
- Output

azuredeploy.json

```
1   {
2     "$schema": "https://schema.management.azure.com/schemas/2015-01-01/deploymentTemplate.json#",
3     "contentVersion": "1.0.0.0",
4     "parameters": {
97    "variables": {
154   "resources": [
715   "outputs": {
729   }
```

# ARM Template Deployment



# Azure Cloud Shell

- Runs on a temporary host: “Bash or PowerShell”
  - Provided on a per-session, per-user basis
  - Times out after 20 minutes without interactive activity
- Permissions set as a regular Linux user in Bash
- Requires an Azure file share to be mounted
  - Persists $HOME using a 5-GB image held in your file share
  - Uses the same Azure file share for both Bash and PowerShell

https://docs.microsoft.com/en-us/azure/cloud-shell/overview

# Azure Command-Line Interface (CLI)

- Perfect tool for building custom automation
- Supported platforms:
  - Get started by running it in Cloud Shell
  - Installation package for:

    **Windows 10 / Ubuntu 16.04+ / macOS**
- Recommended CLI documentation:

https://docs.microsoft.com/en-us/cli/azure/?view=azure-cli-latest

https://docs. microsoft.com/en-us/azure/virtual-machines/azure-cli-arm-commands

# Troubleshoot Deployment Errors

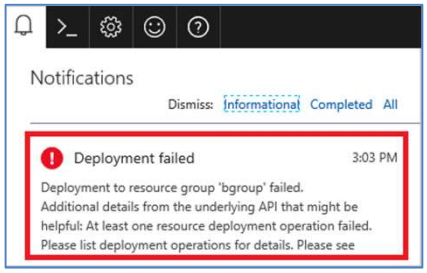
- Azure PowerShell:

```
(Get-AzResourceGroupDeploymentOperation -DeploymentName
exampledeployment -ResourceGroupName
examplegroup).Properties.statusMessage
```

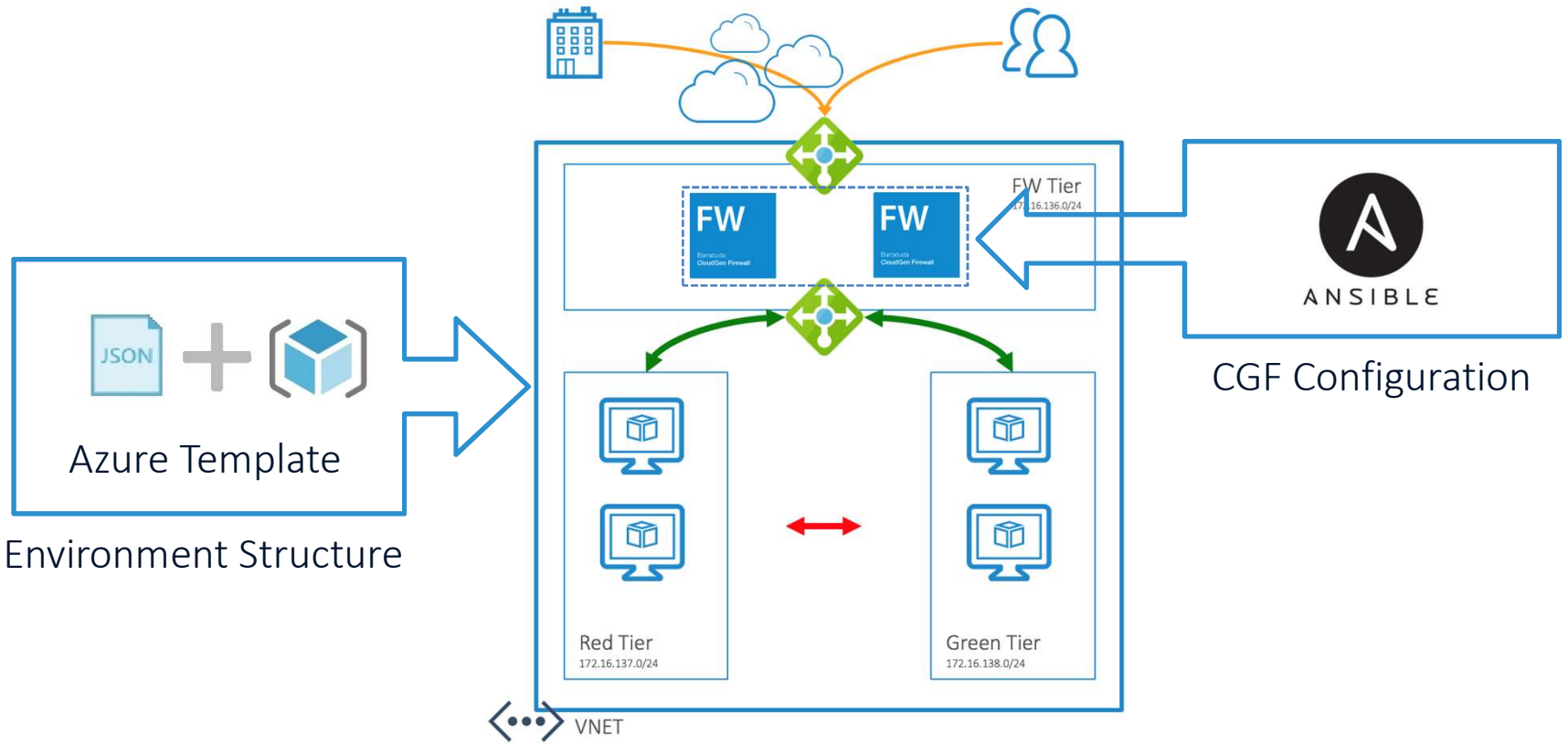
- Azure CLI:

```
az group deployment operation list --name exampledeployment
-g examplegroup --query "[*].properties.statusMessage"
```

- Portal:

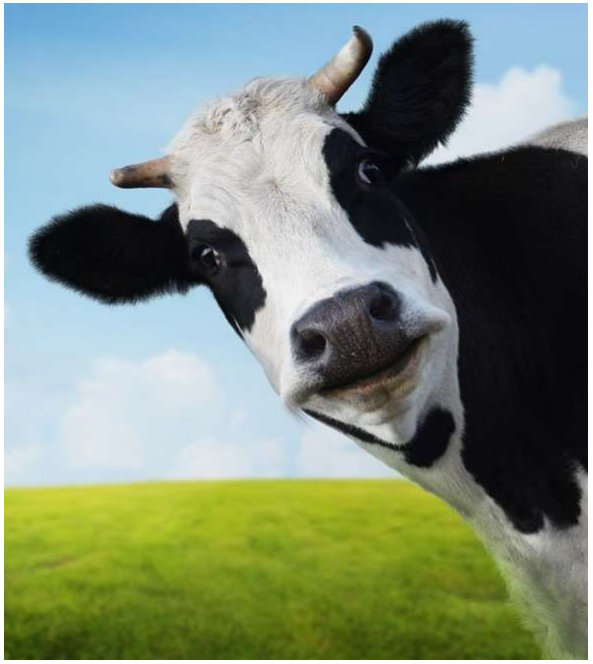


# Azure Template + Ansible



# Ansible

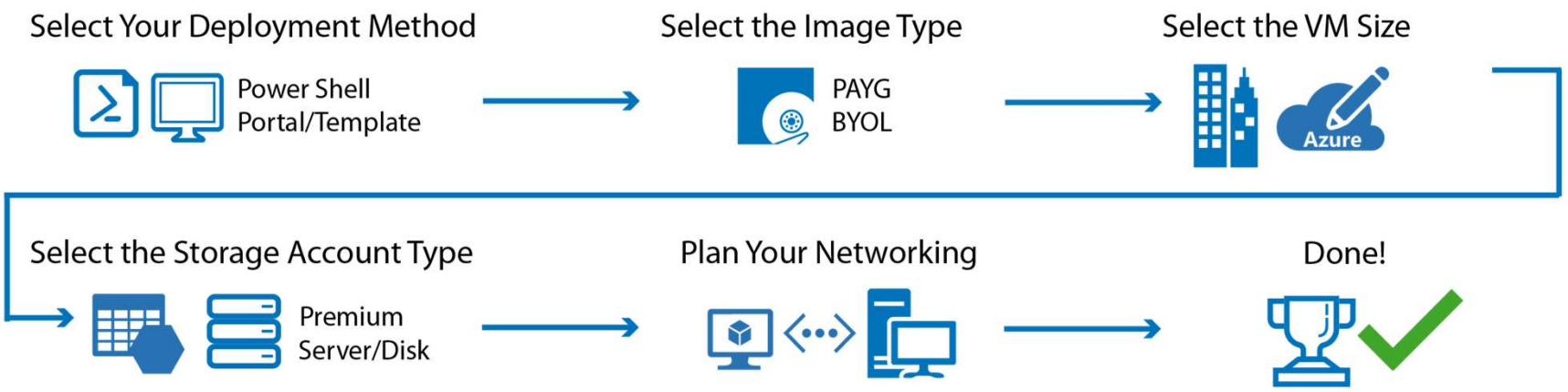
- Why Ansible?
  - Most simple automation tool
  - Human-readable automation files (YAML)
  - No special coding skills needed
- Automation of nearly everything:
  - Cloud provisioning
  - Configuration management
  - Application deployment
  - Intra-service orchestration



# CGF

## Deployment Methods

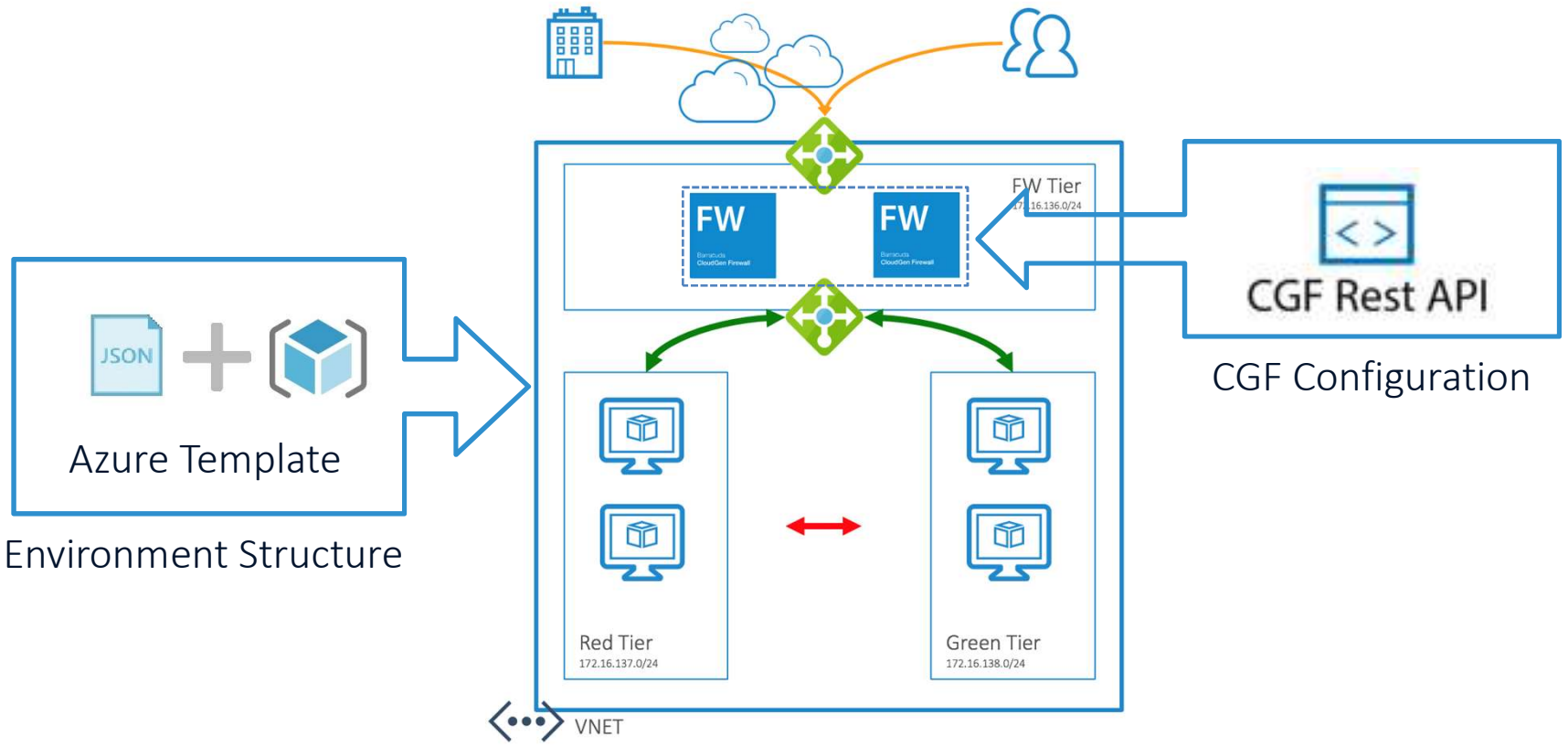
# Planning Your CGF Deployment



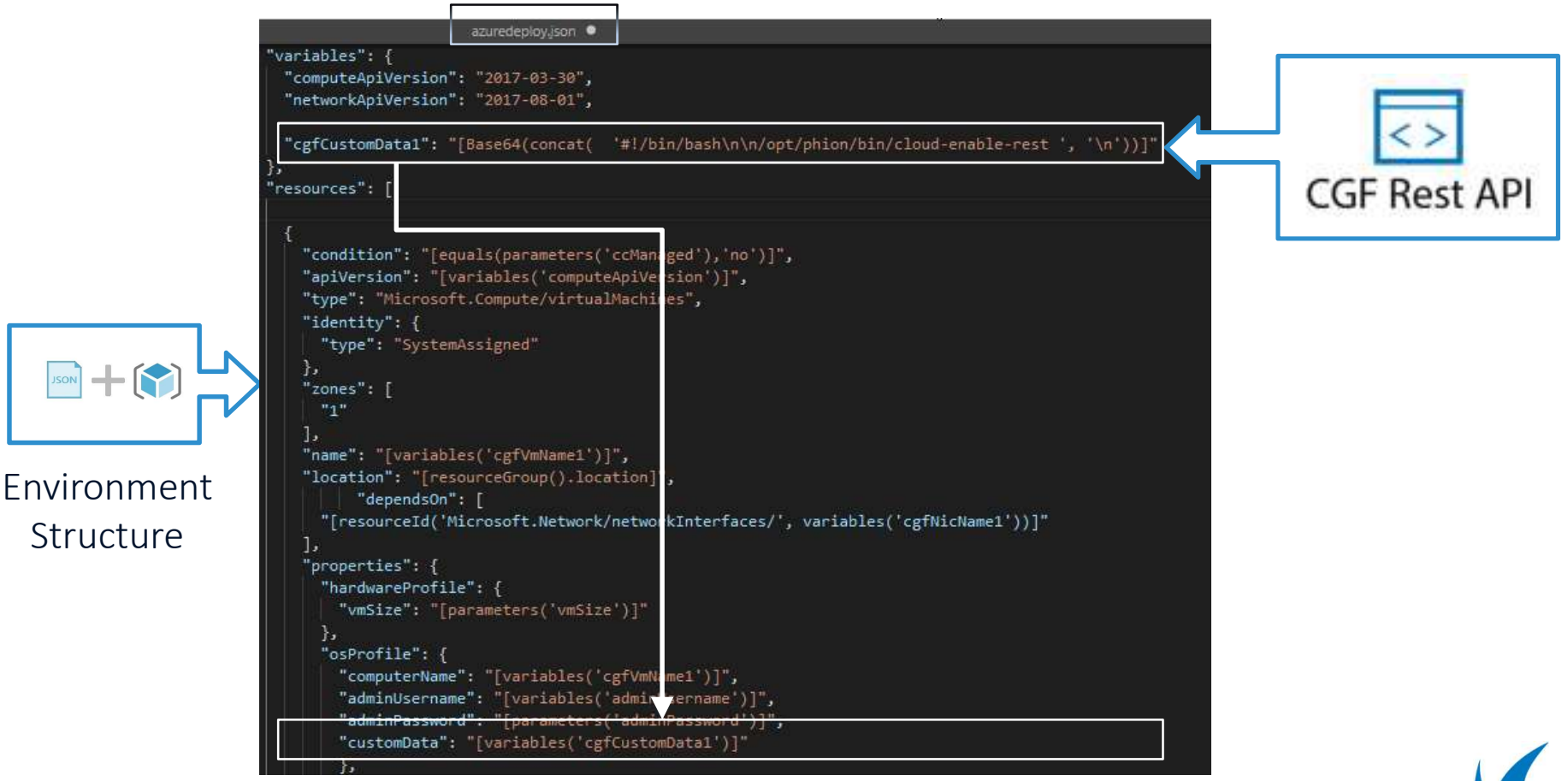
# Deployment Options


- Web portal
  - Solution templates available
- Azure templates
- Azure PowerShell
- VHD disk images via Barracuda Download Portal
  - For older firmware versions
- https://github.com/barracudaNetworks

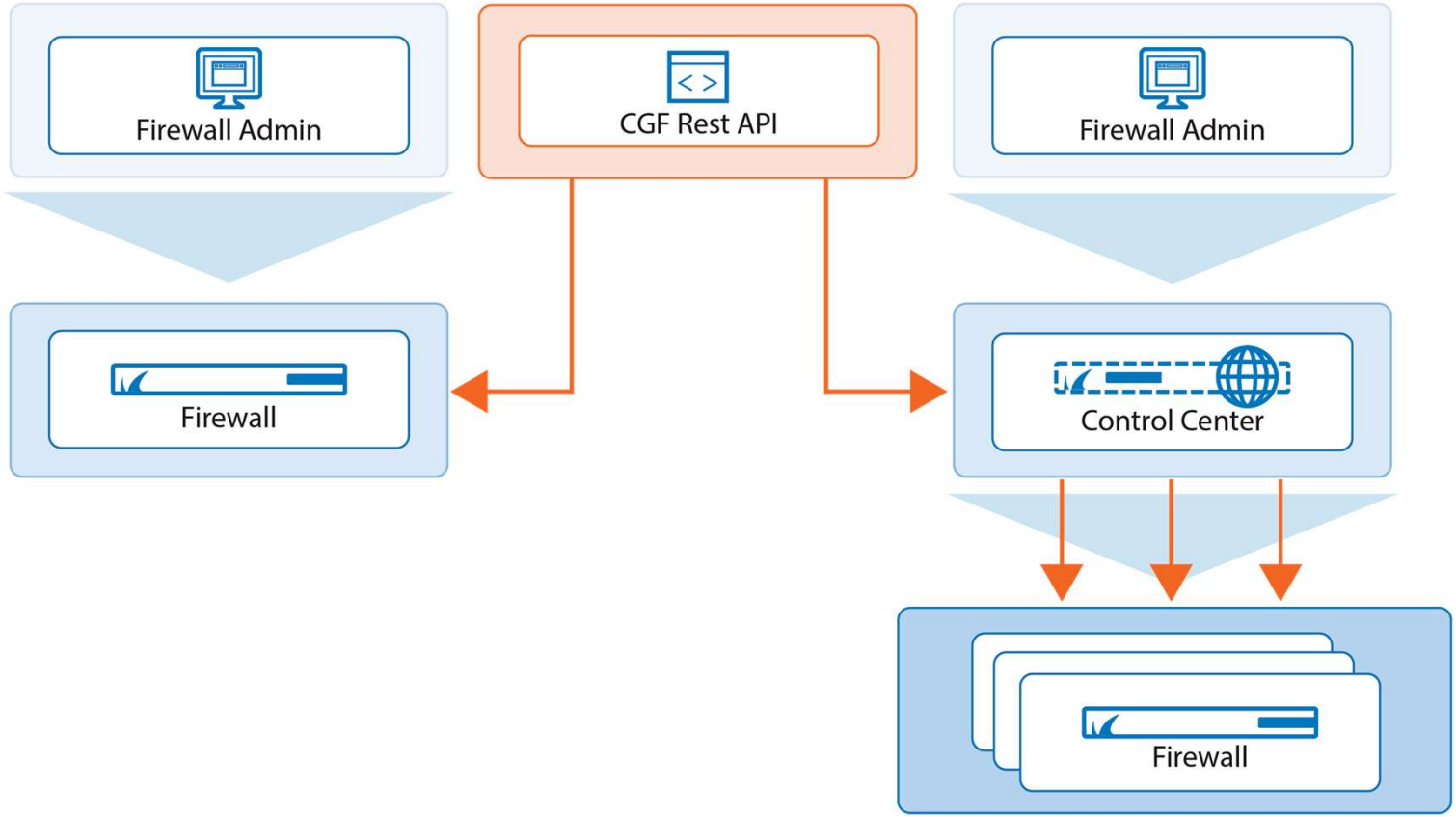
# Azure Template + CGF REST API



# CGF Azure Template + CGF REST API

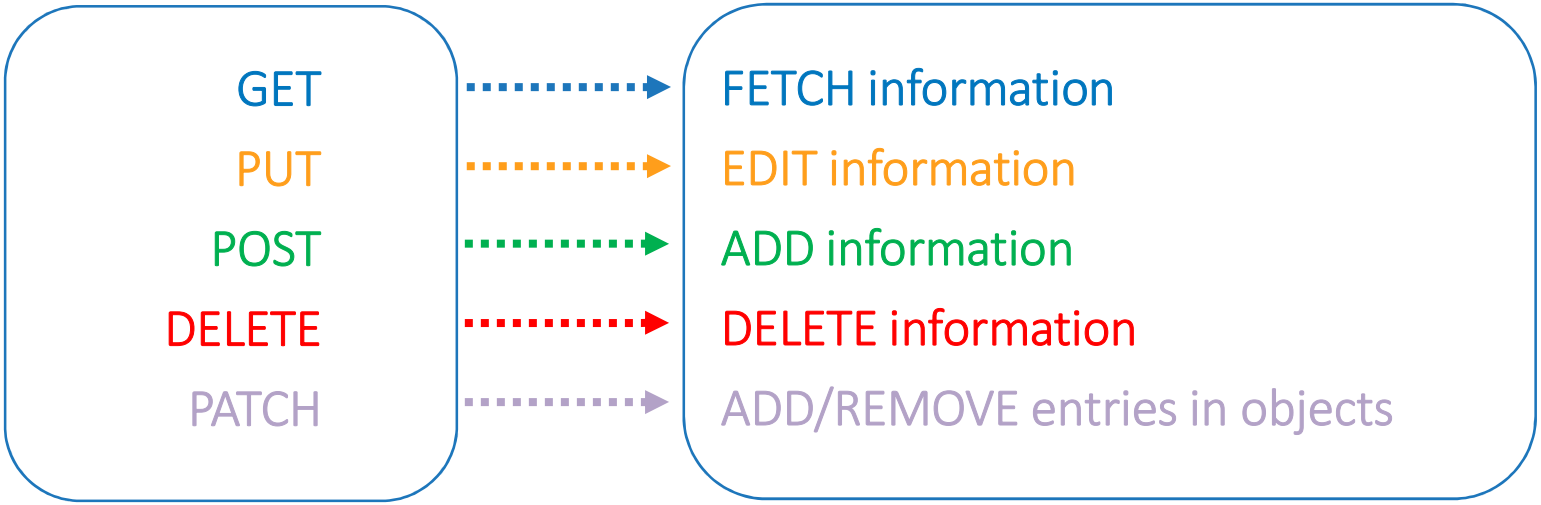


# CGF Configuration



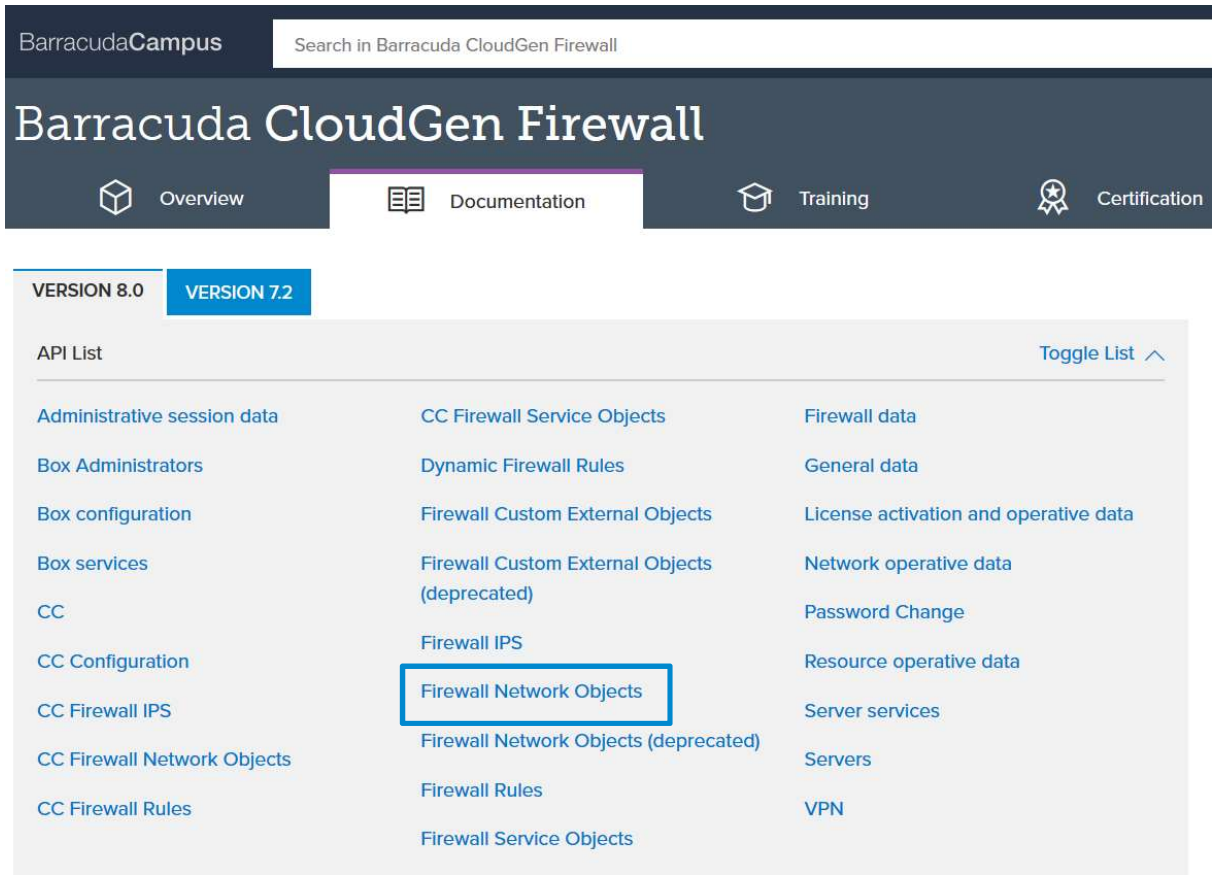
# HTTP Methods – API Objects Actions

- HTTP methods
- API objects actions



# Rest API – Developer Documentation

- GET
- POST
- PUT
- PATCH
- DELETE

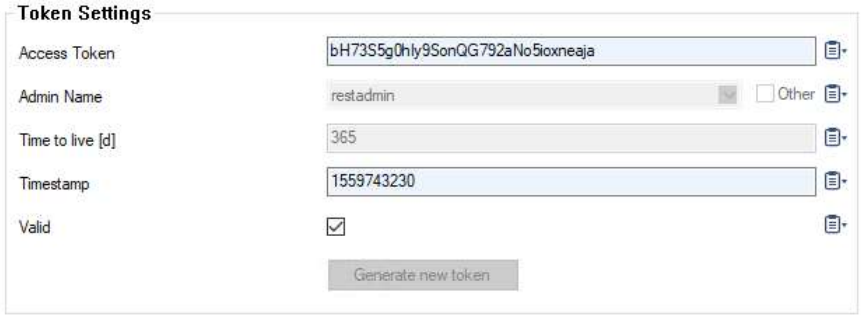


# CGF Rest API

- Be sure to select the correct REST-API command
- Do you want to create a NW-Object in:
  - Host Firewall
  - Forwarding Firewall

# CGF Rest API Config – Step by Step

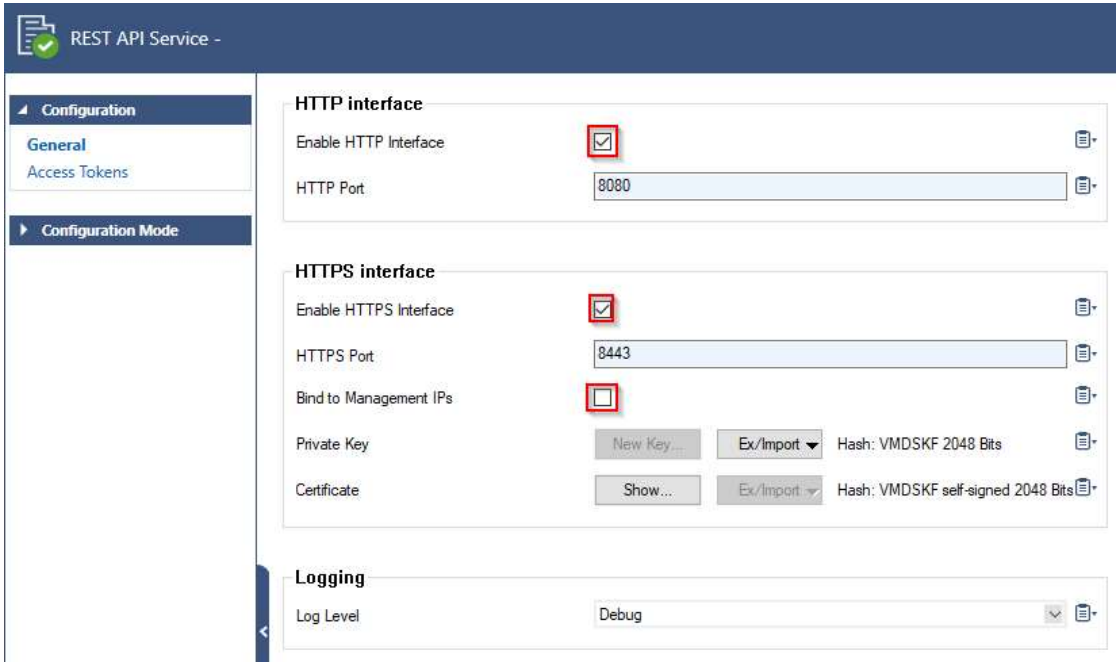
- Create a new administrator
  - Administrators > New (restadmin/AssignedRoles: Manager/….default)
- Create a new REST API certificate
  - Infrastructure Services/REST API Service > New Key+Certificate
- Create a new REST API token for the new administrator
  - Infrastructure Services/REST API Service > New Access Token linked to restadmin
  - COPY this access token



# CGF Rest API Config – Step by Step

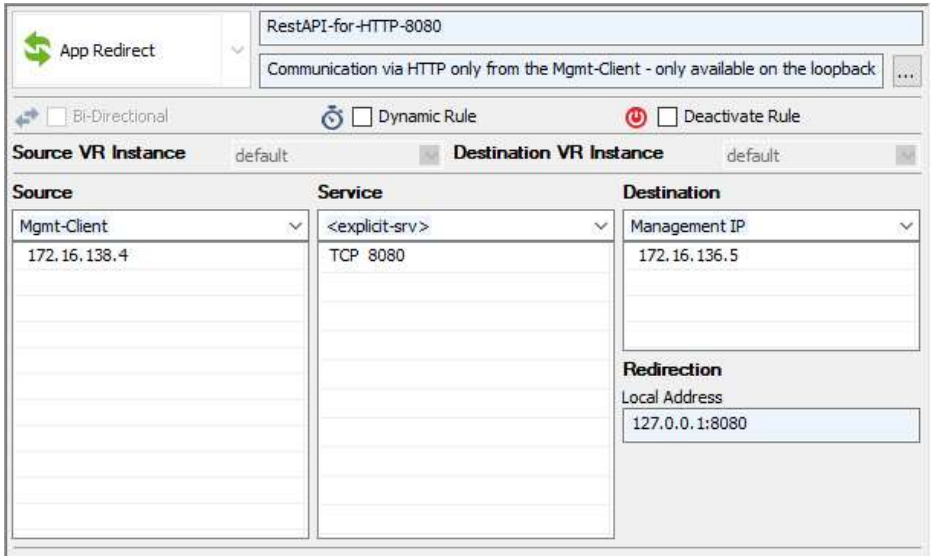
## Enable REST API

- Infrastructure Services/REST API Service



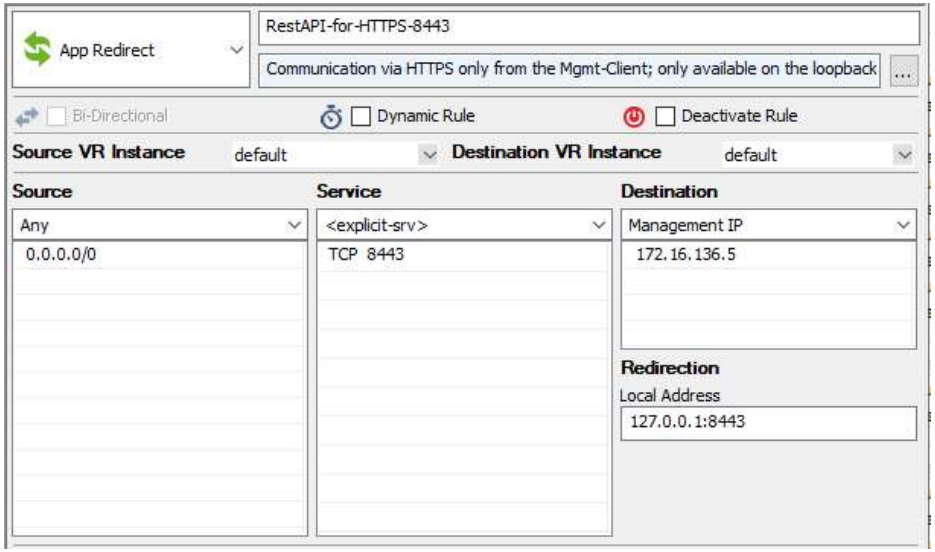
# CGF Rest API Config – Step by Step

- Create a new access rule to allow HTTP traffic to REST API
  - New AppRedirect
    Src.:Client / Srv.:8080 / Dest.:MgmtIPofCGF / Redirect:127.0.0.1:8080



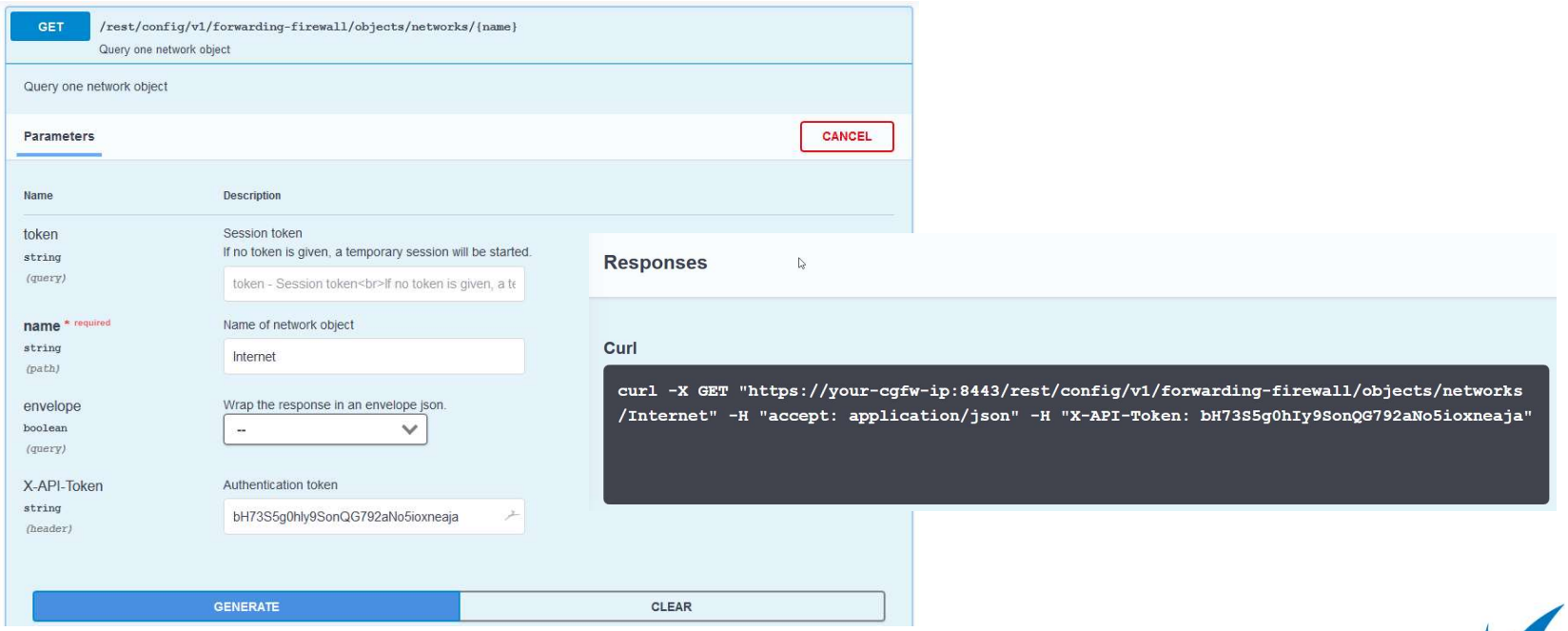
# CGF Rest API Config – Step by Step

- Create a new access rule to allow HTTPS traffic to REST API
  - New AppRedirect
    Src.:Client / Srv.:8443 / Dest.:MgmtIPofCGF / Redirect:127.0.0.1:8443



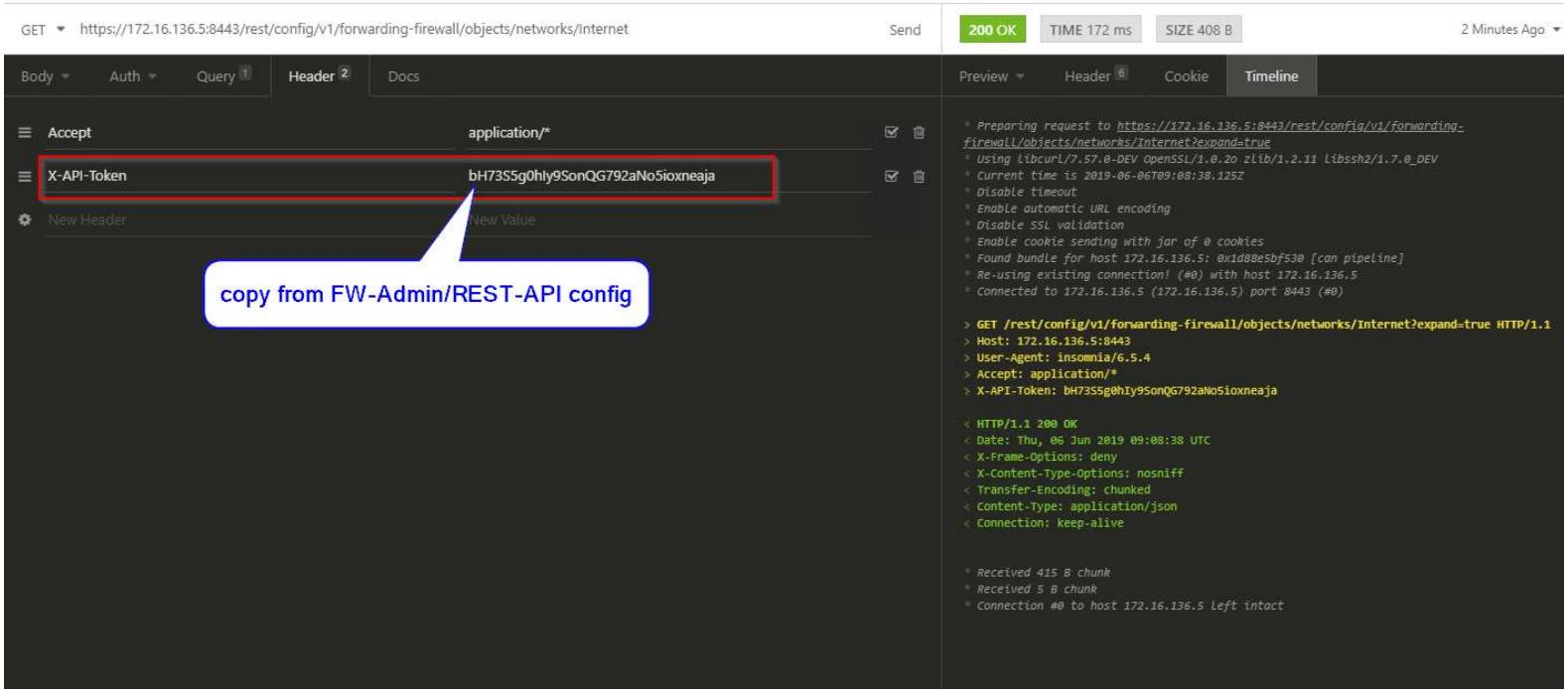
# CGF Rest API Config – Step by Step

- Find examples on Barracuda Campus REST API documentation
  - https://campus.barracuda.com/product/nextgenfirewallf/api#/

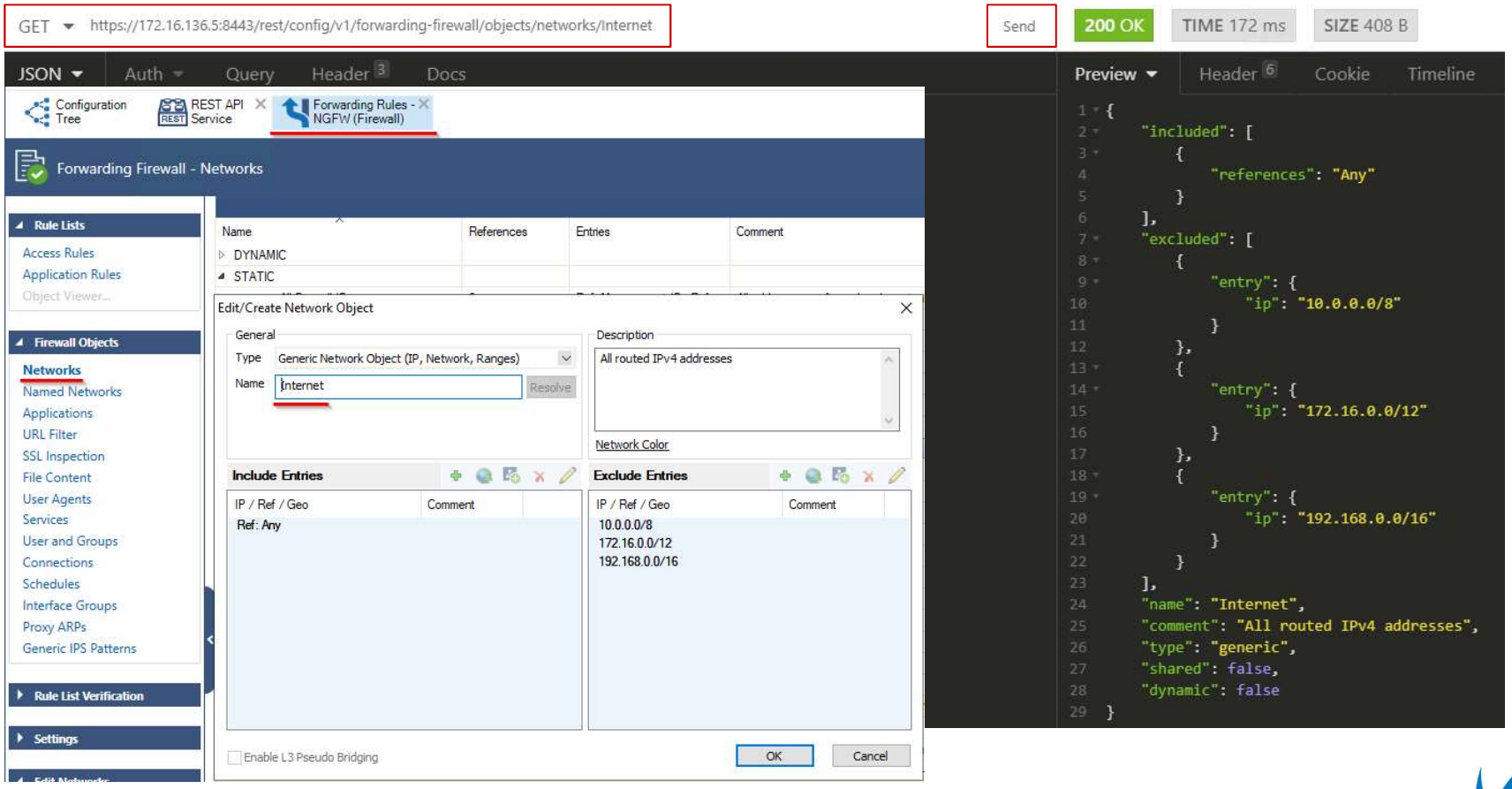


# CGF Rest API Config – Step by Step

## Check REST API with a REST API client, such as Insomnia

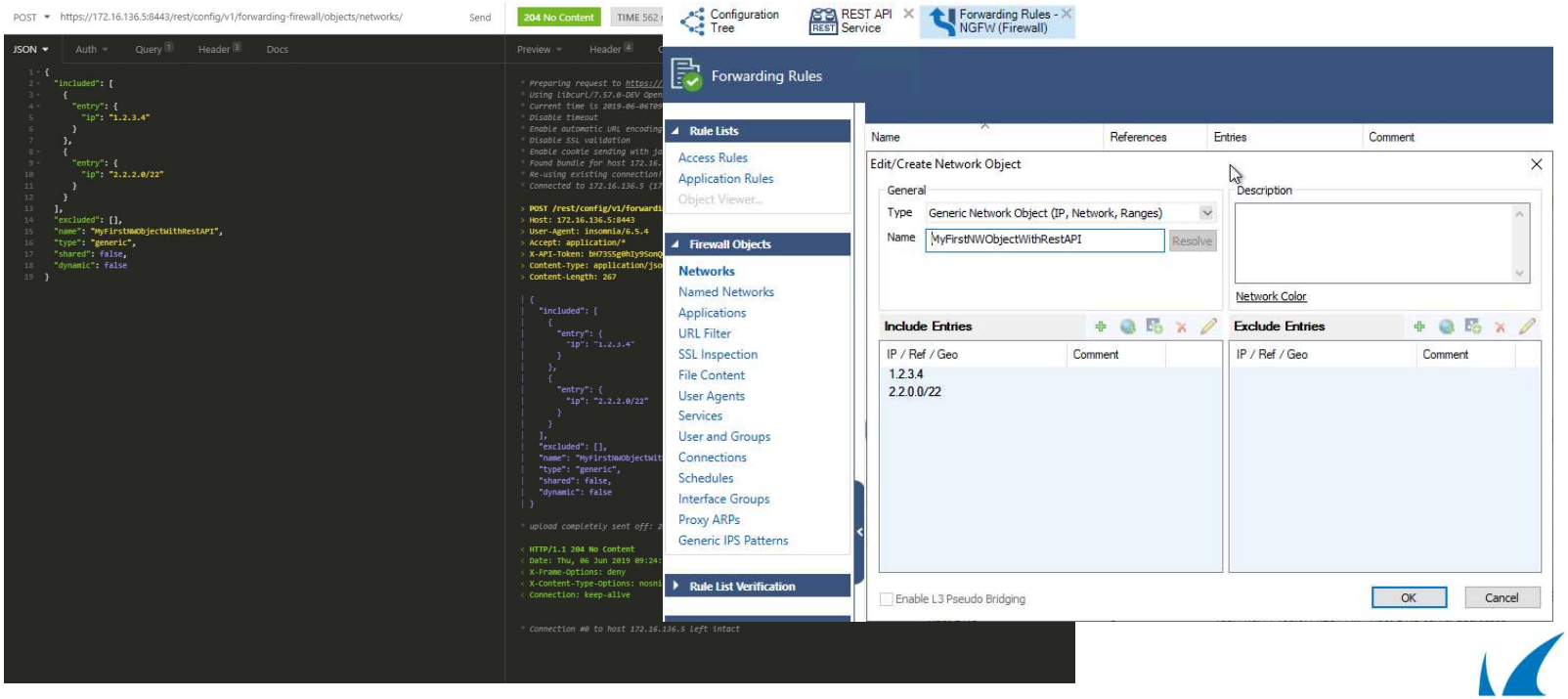


# CGF Rest API Config – Step by Step

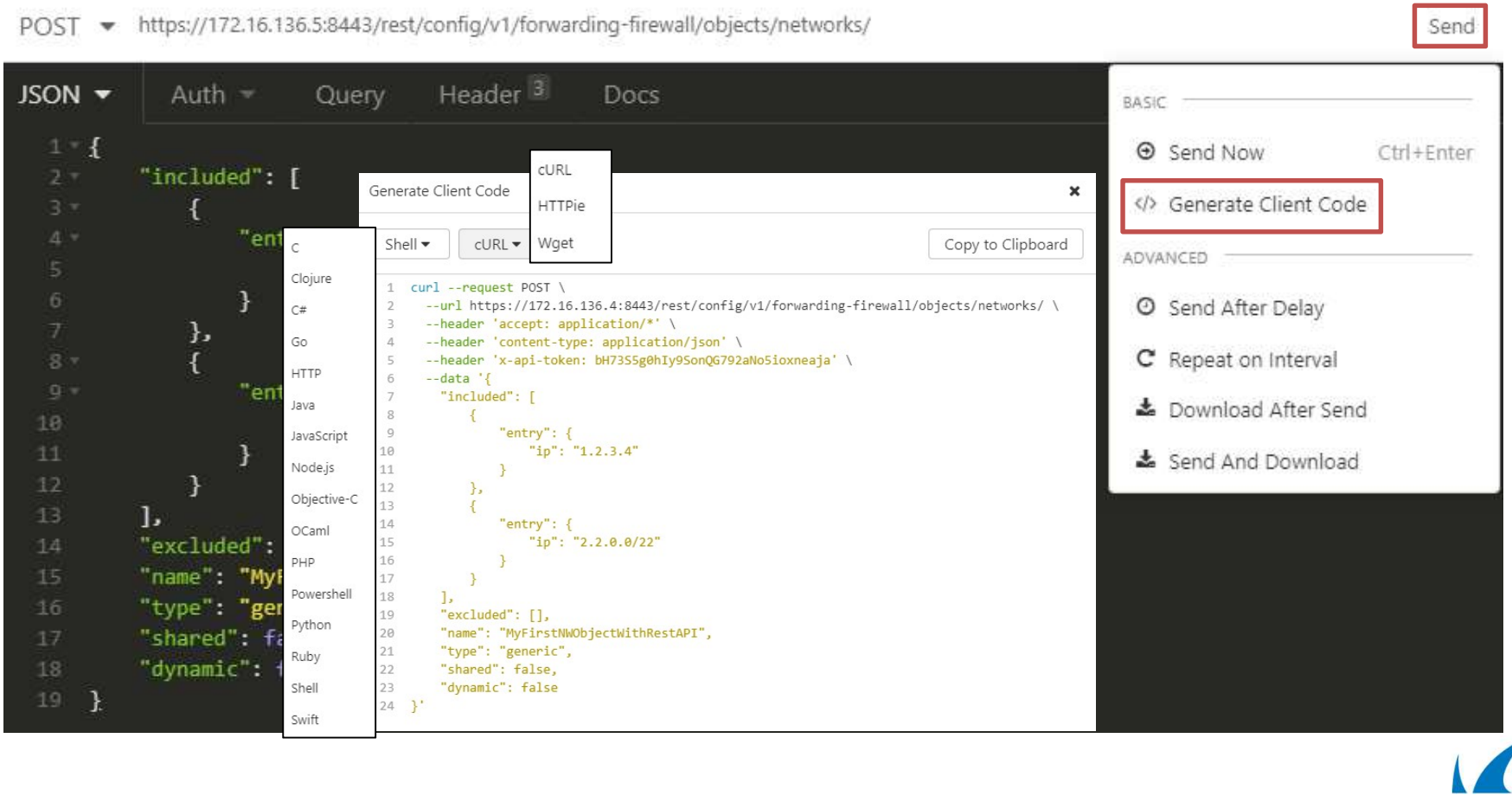


# CGF Rest API Example – Network Object

- Create a NEW network object with REST API
  - `MyFirstNWObjectWithRestAPI: 1.2.3.4|2.2.2.0/22`



# CGF Rest API Example – Network Object

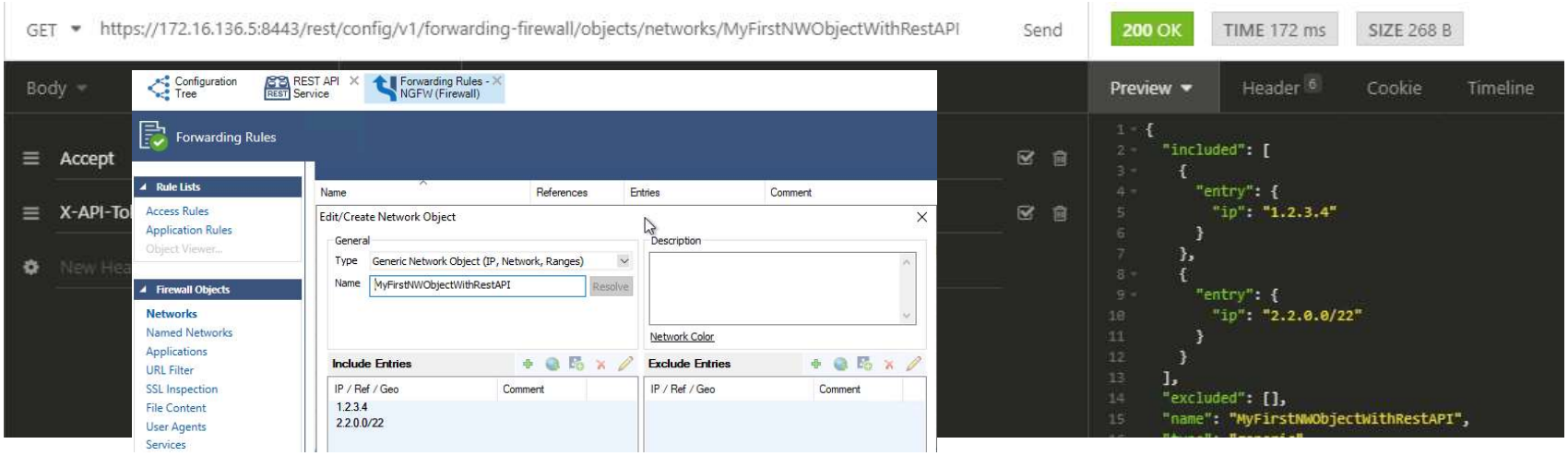


# CGF Rest API Example – Network Object

- CURL examples:
  - curl -X POST "https://your-cgfw-ip:8443/rest/config/v1/forwarding-firewall/objects/networks" -H "accept: */*" -H "X-API-Token: yourRestApiToken" -H "Content-Type: application/json" -d "{\"included\":[{\"entry\":{\"ip\":\"1.2.3.4\"}},{\"entry\":{\"ip\":\"2.2.2.0/22\"}}],\"excluded\":[],\"name\":\"MyFirstNWObjectWithRestAPI\",\"type\":\"generic\",\"shared\":false,\"dynamic\":false}“
  - curl -X POST "https://your-cgfw-ip:8443/rest/config/v1/forwarding-firewall/objects/networks" -H "accept: */*" -H "X-API-Token: yourRestApiToken" -H "Content-Type: application/json" -d "@MyJSON-File"

# CGF Rest API Example – Network Object

- Check the NEW network object with REST API
  - curl -k -X GET "https://172.16.136.5:8443/rest/config/v1/forwarding-firewall/objects/networks/MyFirstNWObjectWithRestAPI" -H "accept: */*" -H "X-API-Token: bH73S5g0hIy9SonQG792aNo5ioxneaja"



# Ansible Example

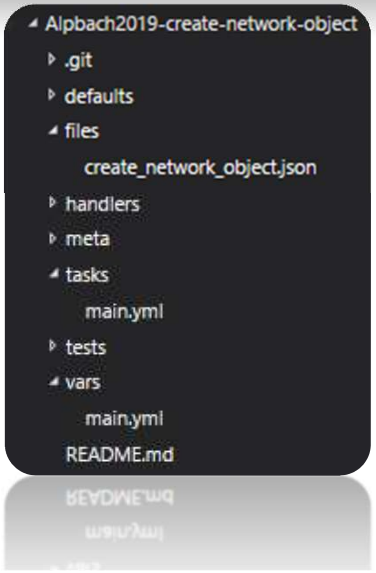
- >ansible-playbook Alpbach2019-create-network-object.yml

Alpbach2019-create-network-object.yml

```yaml
- hosts: localhost
  gather_facts: no
  connection: local
  roles:
    - { role: Alpbach2019-create-network-object}
```

main.yml

```yaml
---
- name: CGF - Create network object
  uri:
    url: "https://{{ CGF.publicIP }}:{{cgf_service_https_port}}/rest/config/v1/forwarding-firewall/objects/networks"
    method: POST
    headers:
      "X-API-Token": "{{ cgf_token.json.token }}"
    validate_certs: false
    body_format: json
    body: "{{lookup('file', 'create_network_object.json') | from_json }}"
    status_code: '204'
    timeout: 10
```




main.yml

```yaml
---
# vars file for CGF-create-user

cgf_service_https_port: "8443"
```

create_network_object.json

```json
{
  "name": "demo2-VNET",
  "comment": "demo2-VNET",
  "type": "generic",
  "shared": false,
  "dynamic": false,
  "included": [
    {
      "entry": {
        "ip": "172.16.136.0/22"
      }
    }
  ],
  "excluded": []
}
```

# Ansible Example

- >ansible-playbook Alpbach2019-create-user.yml

Alpbach2019-create-user.yml

```yaml
- hosts: localhost
  gather_facts: no
  connection: local
  roles:
    - { role: Alpbach2019-create-user}
```

Alpbach2019-create-user
- .git
- defaults
- files
  - user_body.json
- handlers
- meta
- tasks
  - main.yml
- tests
- vars
  - main.yml

main.yml

```yaml
---
#Create REST admin user
- name: Create REST Admin user
  uri:
    url: "https://{{ CGF.publicIP }}:{{cgf_service_https_port}}/rest/config/v1/box/admins"
    method: POST
    user: "{{ cgf_username }}"
    password: "{{ cgf_password }}"
    force_basic_auth: yes
    headers:
      Accept: '*/*'
    validate_certs: false
    body_format: json
    body: "{{lookup('file', 'user_body.json') | from_json
    status_code: '204'
    timeout: 10
```

main.yml

```yaml
---
# vars file for CGF-create-user

cgf_username: "root"
cgf_password: 'CudaL3arner!'
cgf_service_https_port: "8443"
```

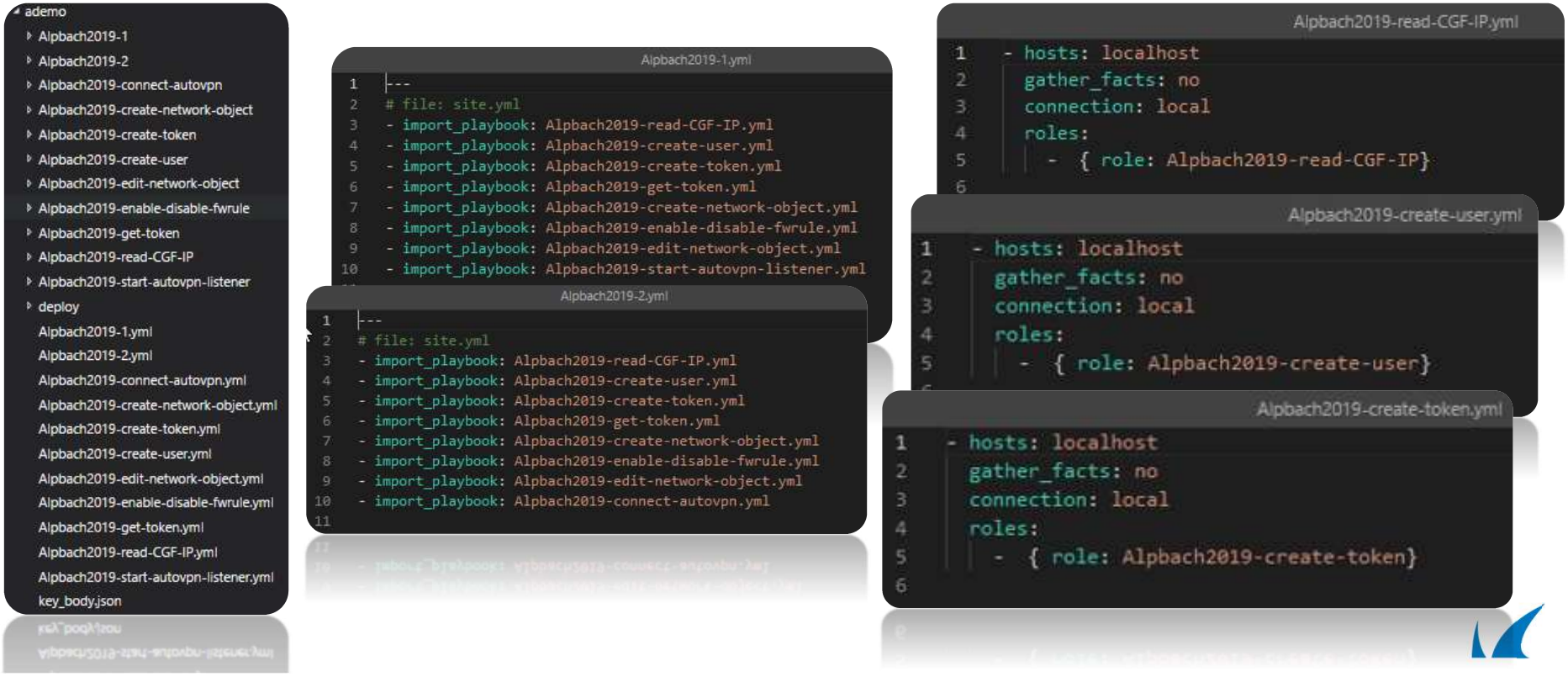
user_body.json

```json
{
    "authenticationLevel": "password",
    "enabled": true,
    "enforcePasswordChange": false,
    "externalLoginName": "restadmin",
    "fullName": "REST Administrator",
    "gracePeriod": 0,
    "loginEvent": "serviceDefault",
    "name": "restadmin",
    "nextForcedChange": 30,
    "password": "R3stAdminPassw0rd!",
    "passwordChangeMode": "forceDifferent",

    "roles": [
        "Manager",
        "Operator"
    ],
    "systemLevelAccess": "standardOsLogin",
    "warningPeriod": 7
}
```

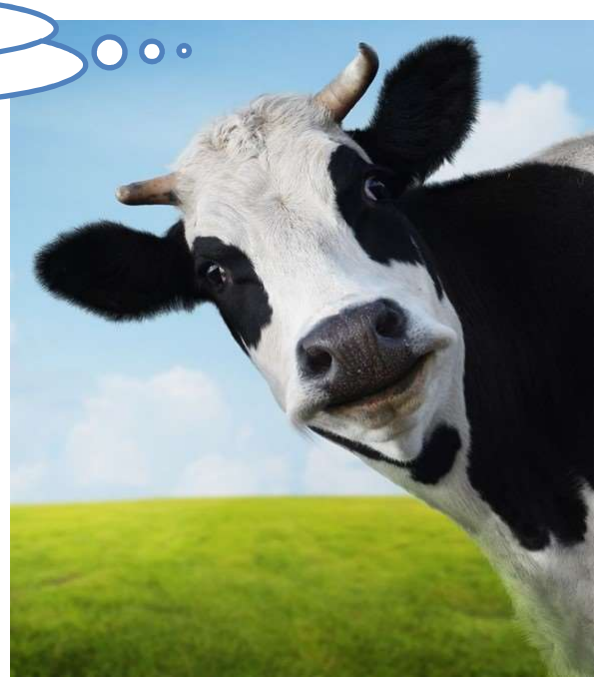
# Ansible Example

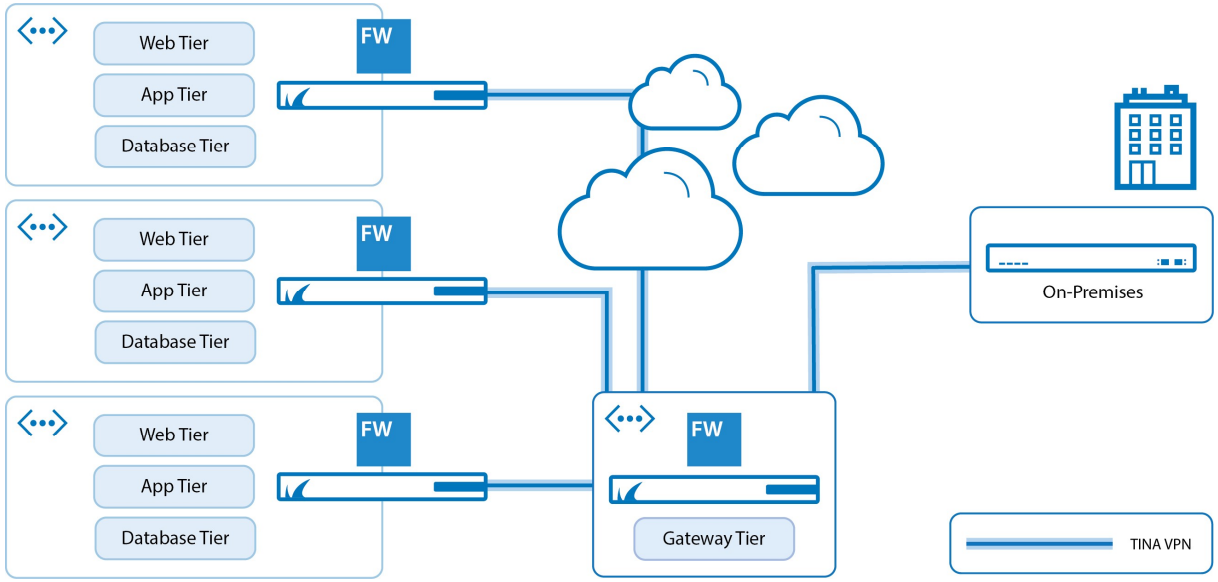
- >ansible-playbook ../Alpbach2019-1.yml



# Multi-VNET with AutoVPN

But how can I automate this?





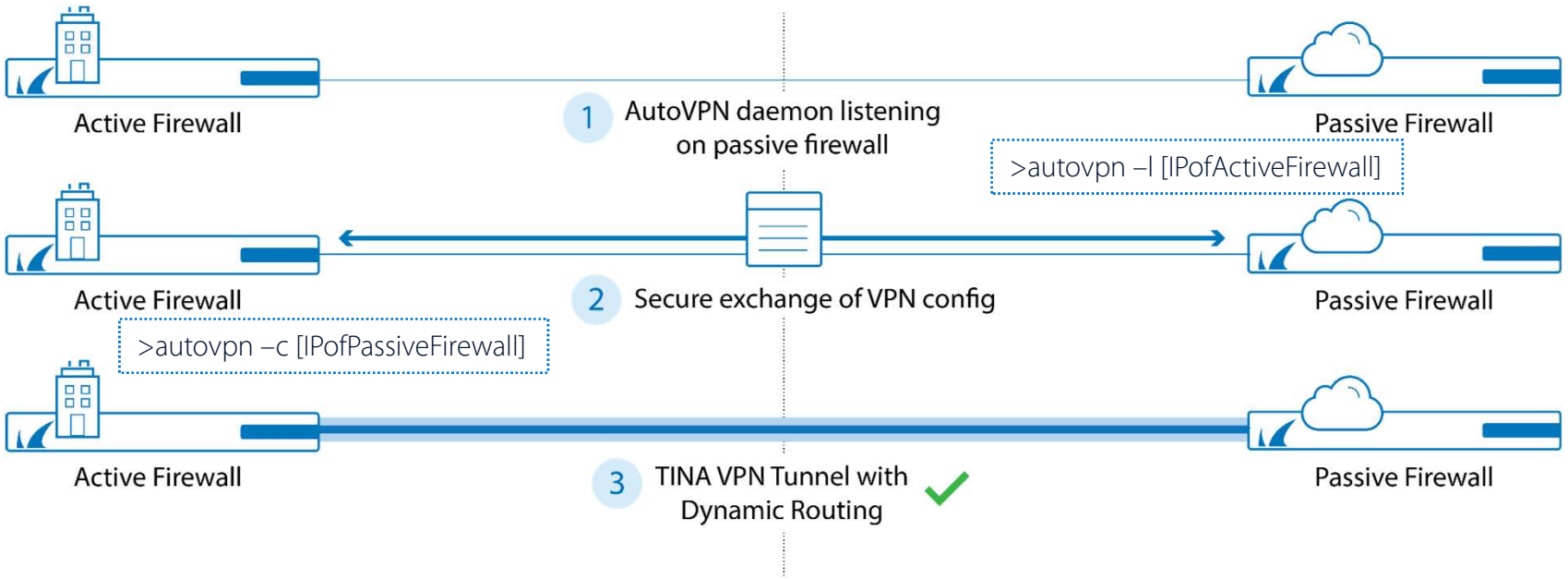
# AutoVPN

- Simplify creating TINA VPN tunnels
- Automation first
  - REST API
  - CLI
- AutoVPN is a configuration tool

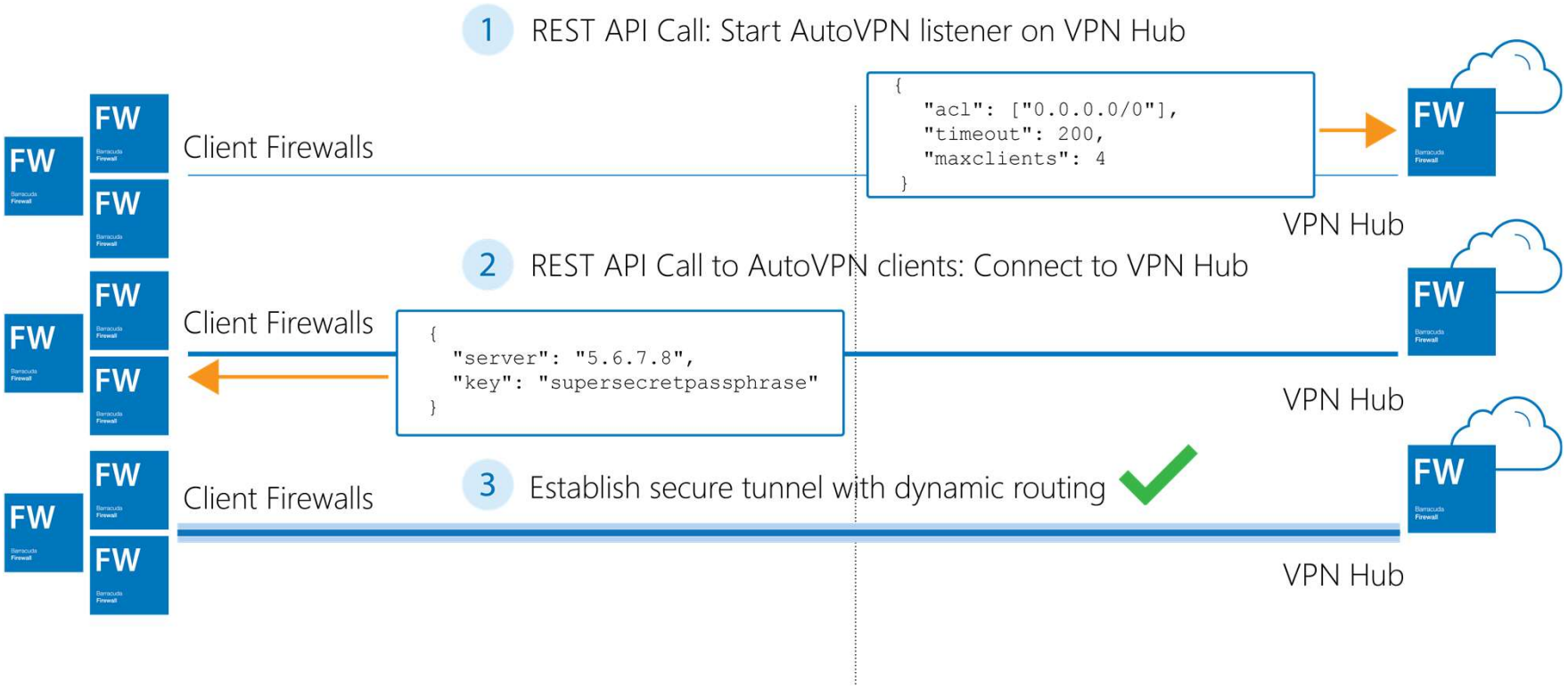
# TINA AutoVPN Features

- Only 8.x platforms are supported: cloud, virtual, and hardware
- AutoVPN uses port 694

# CLI - AutoVPN


Active Firewall
1 AutoVPN daemon listening on passive firewall
Passive Firewall
>autovpn –l [IPofActiveFirewall]
Active Firewall
2 Secure exchange of VPN config
Passive Firewall
>autovpn –c [IPofPassiveFirewall]
Active Firewall
3 TINA VPN Tunnel with Dynamic Routing
Passive Firewall

# REST API with AutoVPN



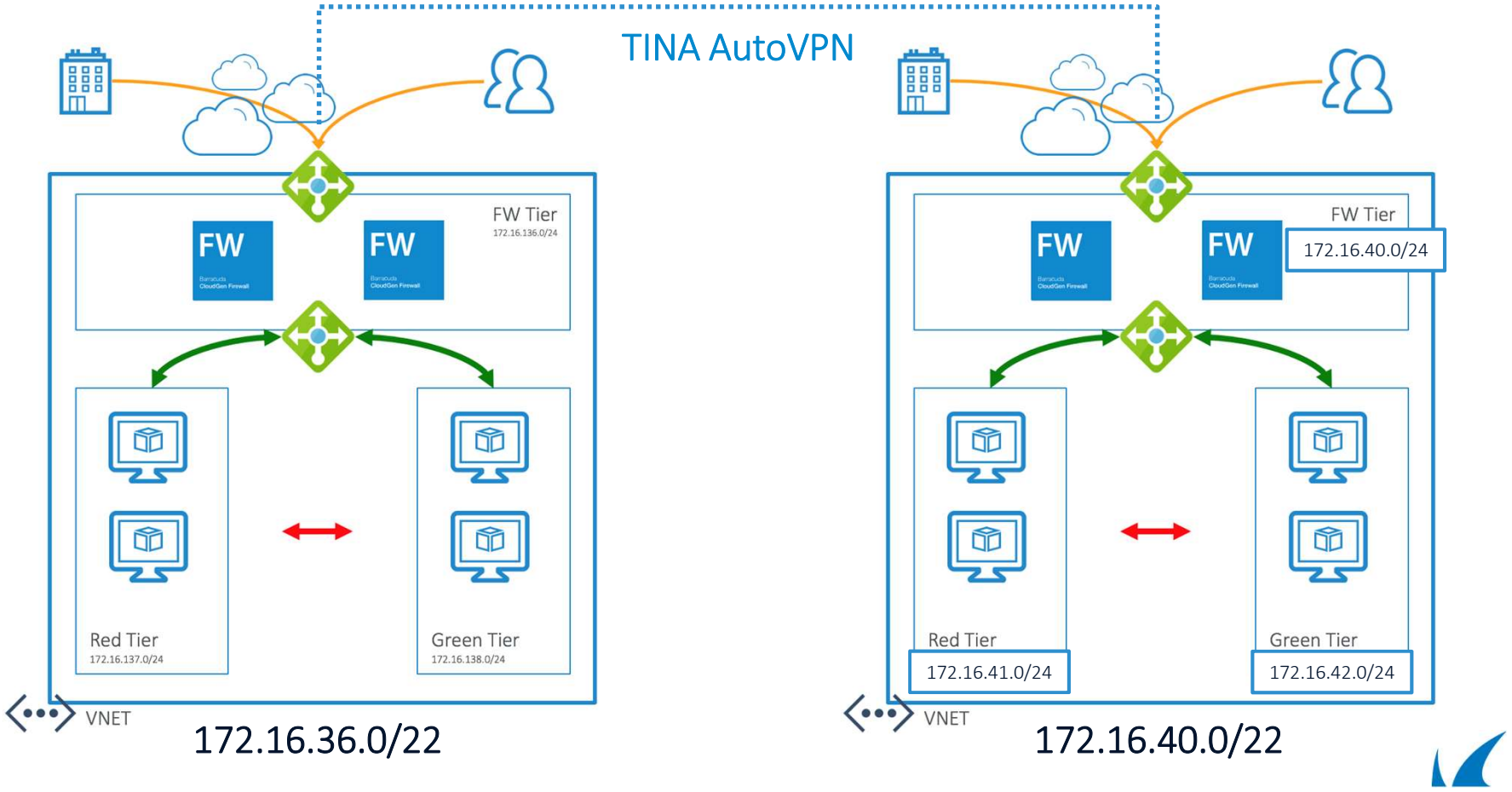
# Technical Details

- Single transport UDP TINA tunnels
- Shared secret used for authentication
- Securing the VPN Hub
  - ACL
  - Number of connecting firewalls
  - Timeout
- REST API documentation on Campus

# Azure CGF Automation

Demo

# Azure Template + Ansible

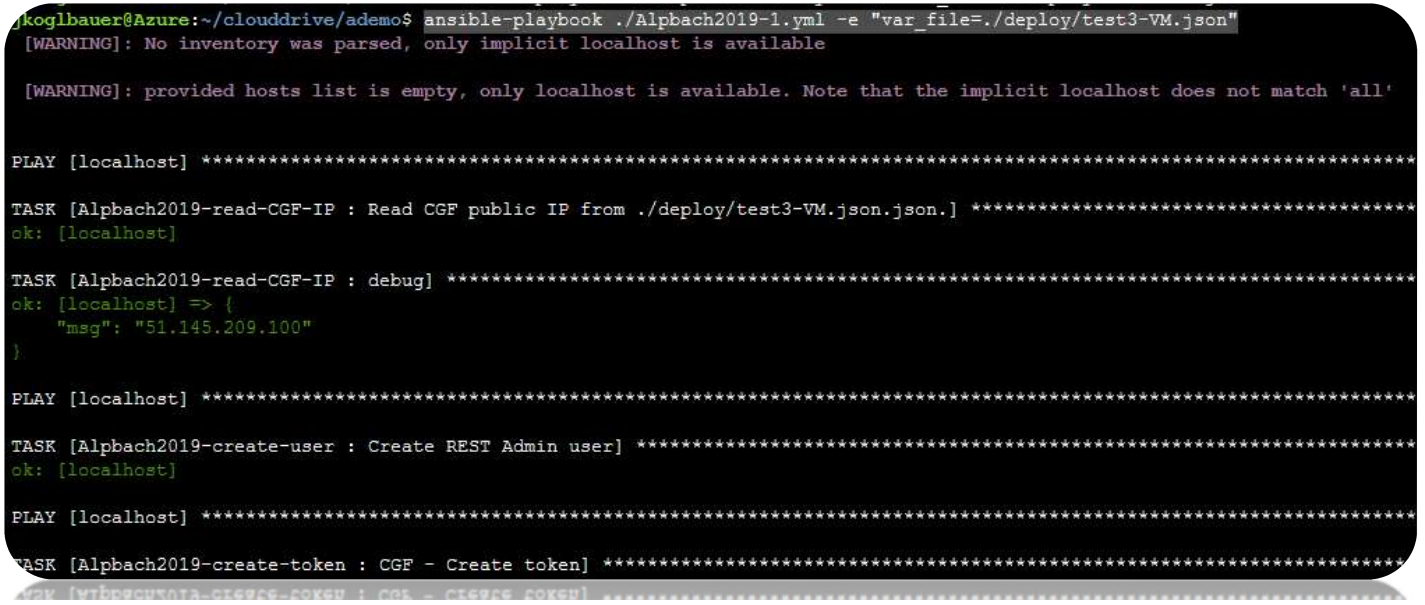


# CGF VPN-Hub Azure Automation Example

- Open Cloud Shell and start deploy.sh
  - `> cd /cloudshell/…  [Folder where deploy1.sh is located]`
  - `> ./deploy1.sh`
- Check the created files
  - `> cat /cloudshell/…/MyPrefix-VM.json`
  - `> cat /cloudshell/…/MyPrefix-VNET.json`
  - `> cat /cloudshell/…/Alpbach2019-create-network-object/files/ create_network_object.json`

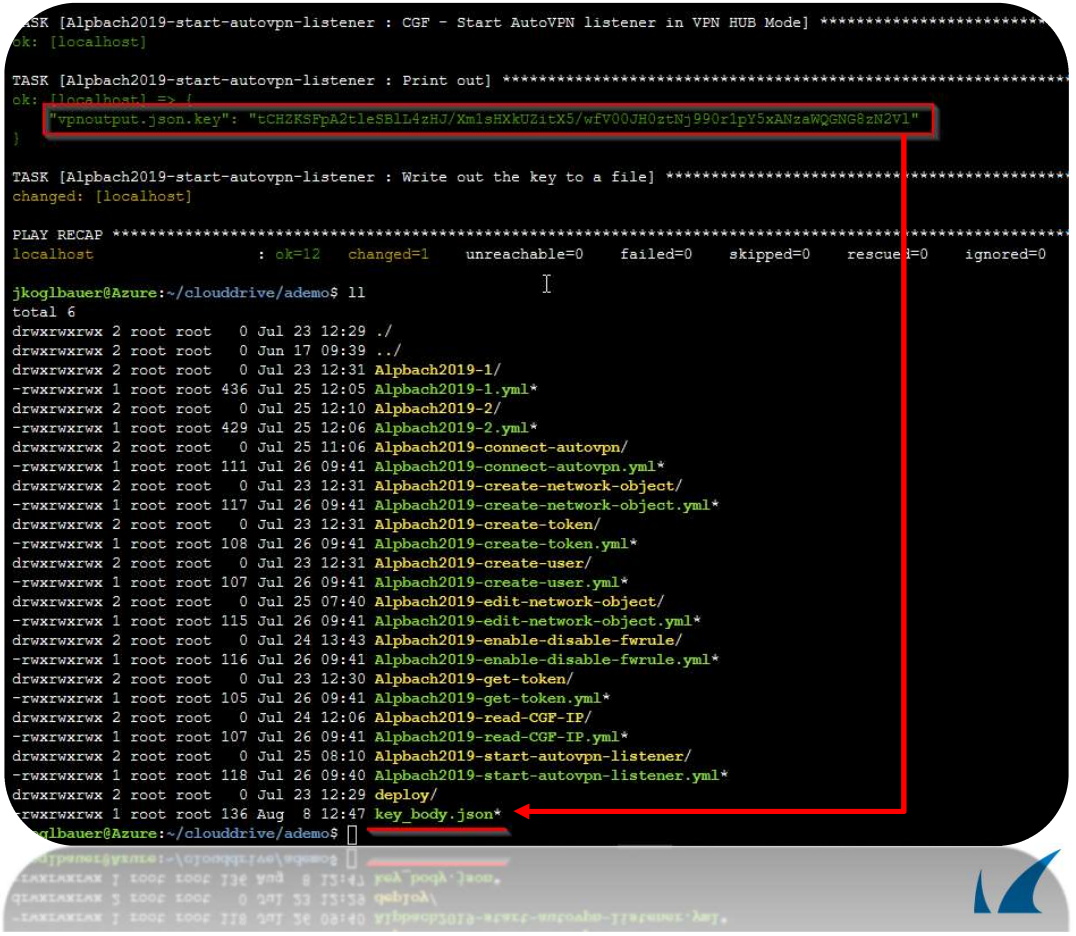
# CGF Azure Automation Example

- Start Ansible
  - > cd ..  [Folder where Alpbach2019-1.yml is located]
  - >ansible-playbook ./Alpbach2019-1.yml -e "var_file=./deploy/test3-VM.json"



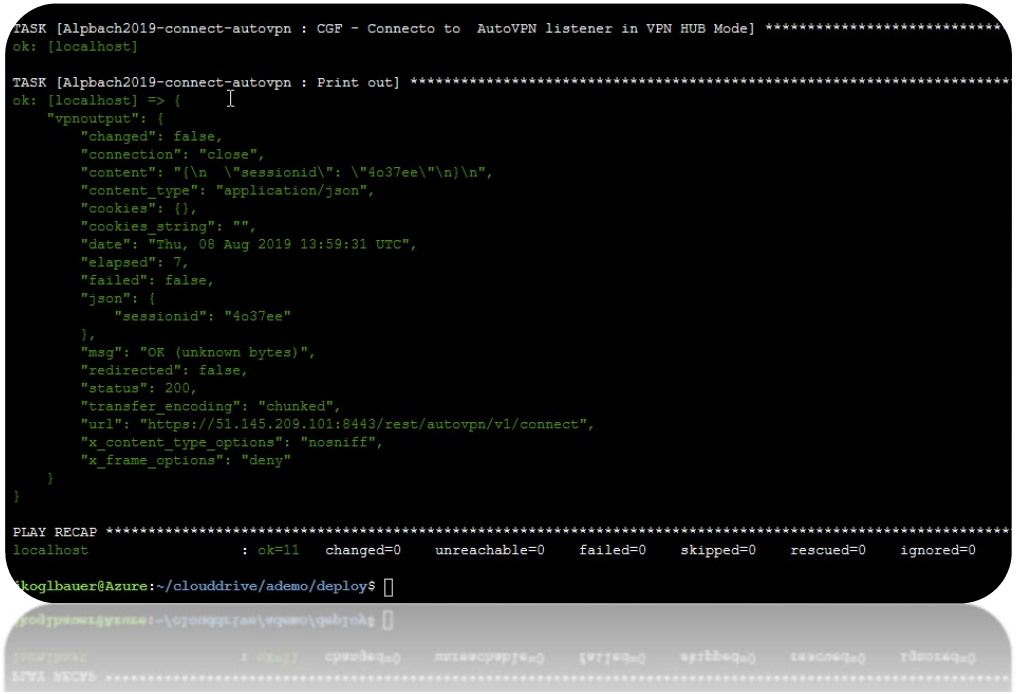
# CGF Azure Automation Example

- Check AutoVPN token



# CGF AutoVPN Client Automation Example

- Open Cloud Shell and start deploy.sh
  - `> cd /cloudshell/…  [Folder where deploy2.sh is located]`
  - `> ./deploy2.sh`




Thank you
Barracuda
TECHSUMMIT19
BARRACUDA TECHNICAL SUMMIT