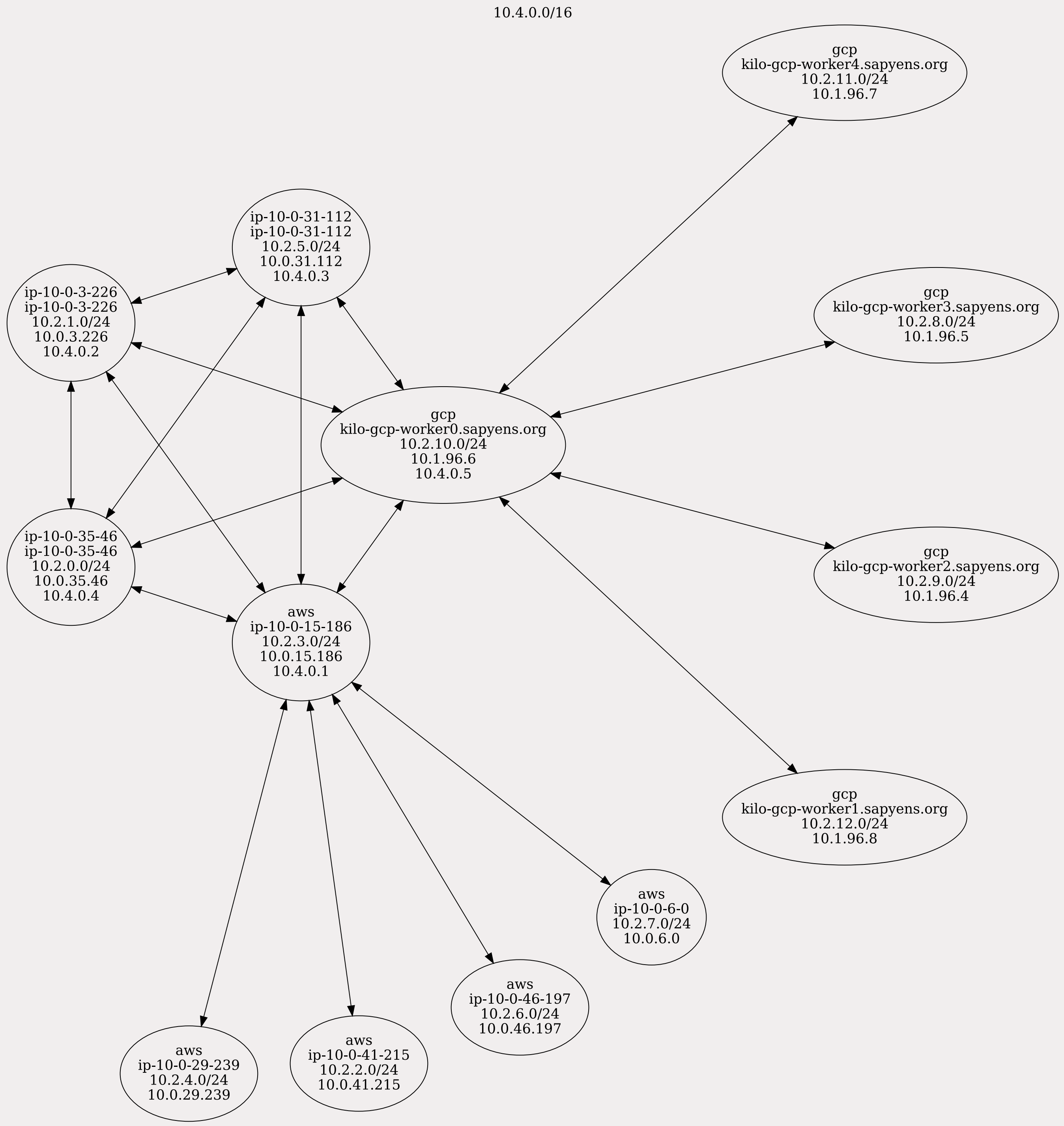
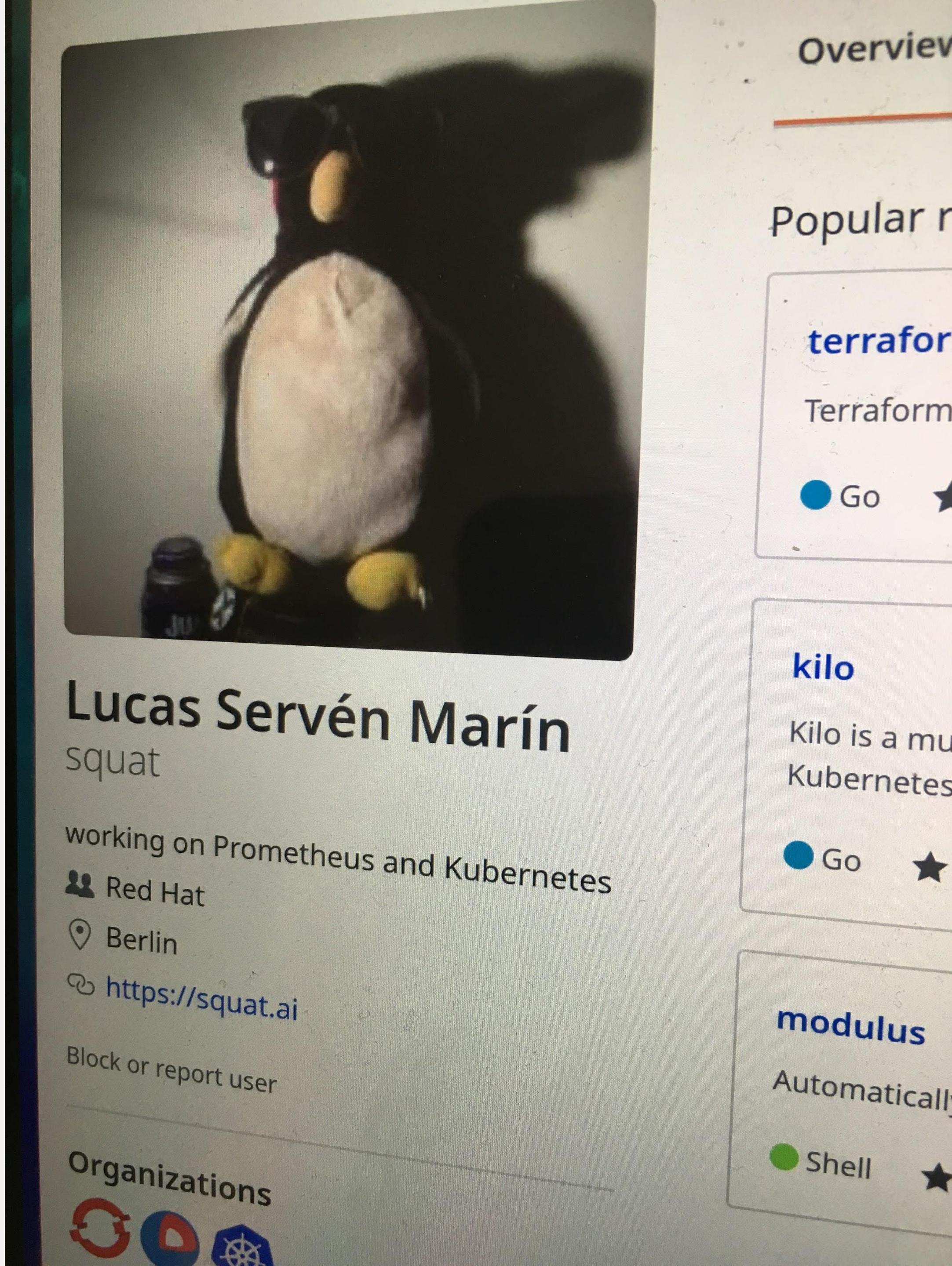
BUILDING MULTI-CLOUD CLUSTERS WITH WIREGUARD

Lucas Servén Marín



10.4.0.0/16
gcp kilo-gcp-worker4.sapyens.org 10.2.11.0/24 10.1.96.7
ip-10-0-31-112 ip-10-0-31-112 10.2.5.0/24 10.0.31.112 10.4.0.3
ip-10-0-3-226 ip-10-0-3-226 10.2.1.0/24 10.0.3.226 10.4.0.2
gcp kilo-gcp-worker3.sapyens.org 10.2.8.0/24 10.1.96.5
gcp kilo-gcp-worker0.sapyens.org 10.2.10.0/24 10.1.96.6 10.4.0.5
ip-10-0-35-46 ip-10-0-35-46 10.2.0.0/24 10.0.35.46 10.4.0.4
gcp kilo-gcp-worker2.sapyens.org 10.2.9.0/24 10.1.96.4
aws ip-10-0-15-186 10.2.3.0/24 10.0.15.186 10.4.0.1
gcp kilo-gcp-worker1.sapyens.org 10.2.12.0/24 10.1.96.8
aws ip-10-0-6-0 10.2.7.0/24 10.0.6.0
aws ip-10-0-46-197 10.2.6.0/24 10.0.46.197
aws ip-10-0-29-239 10.2.4.0/24 10.0.29.239
aws ip-10-0-41-215 10.2.2.0/24 10.0.41.215

LUCAS SERVÉN MARÍN


Lucas Servén Marín
squat

working on Prometheus and Kubernetes

Red Hat

Berlin

https://squat.ai

Block or report user

Organizations

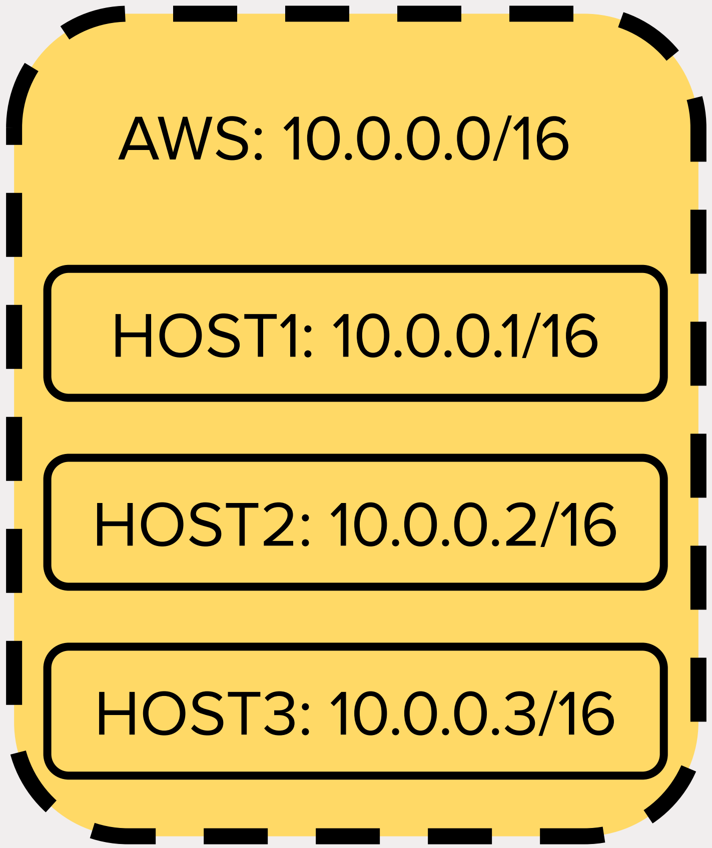
?

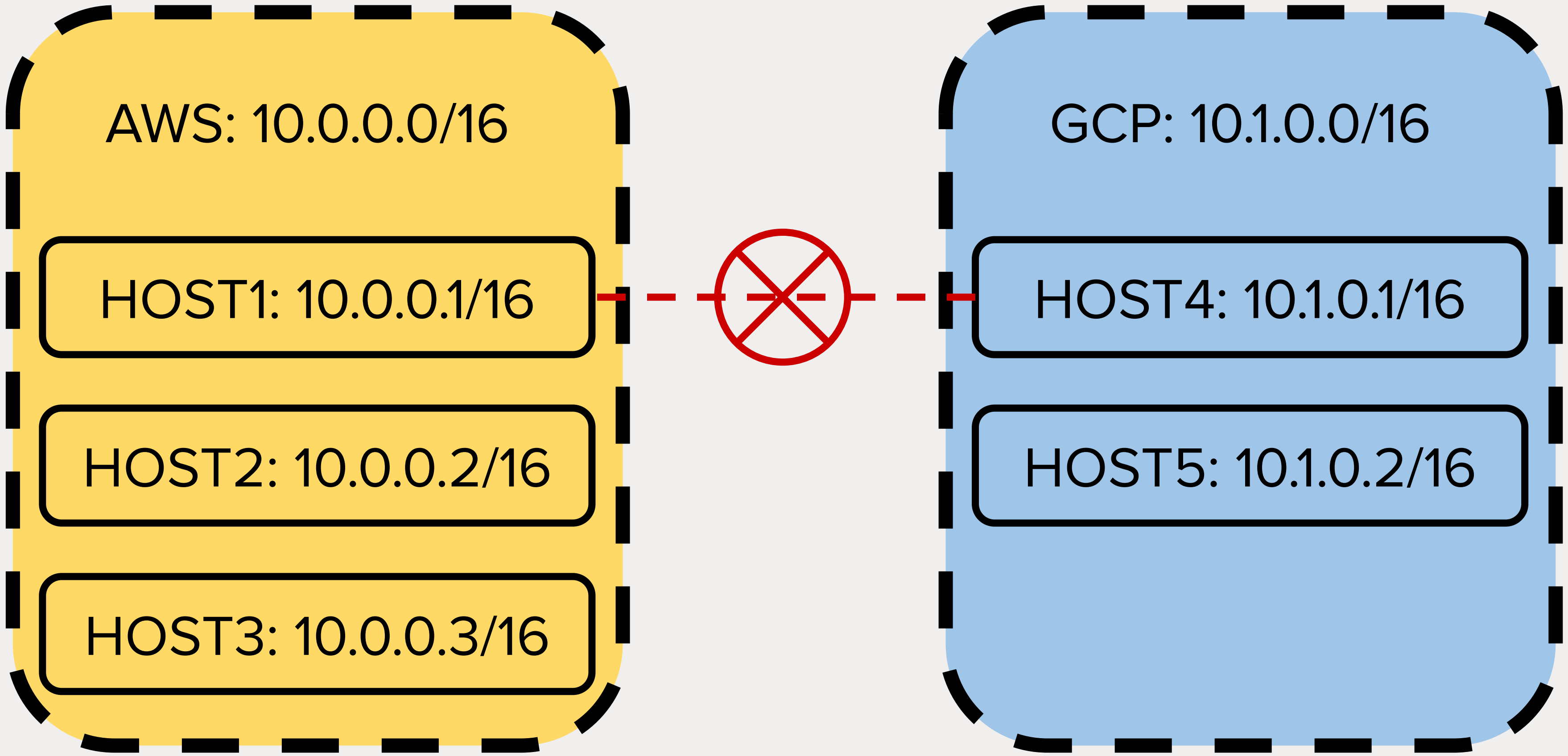
WireGuard®

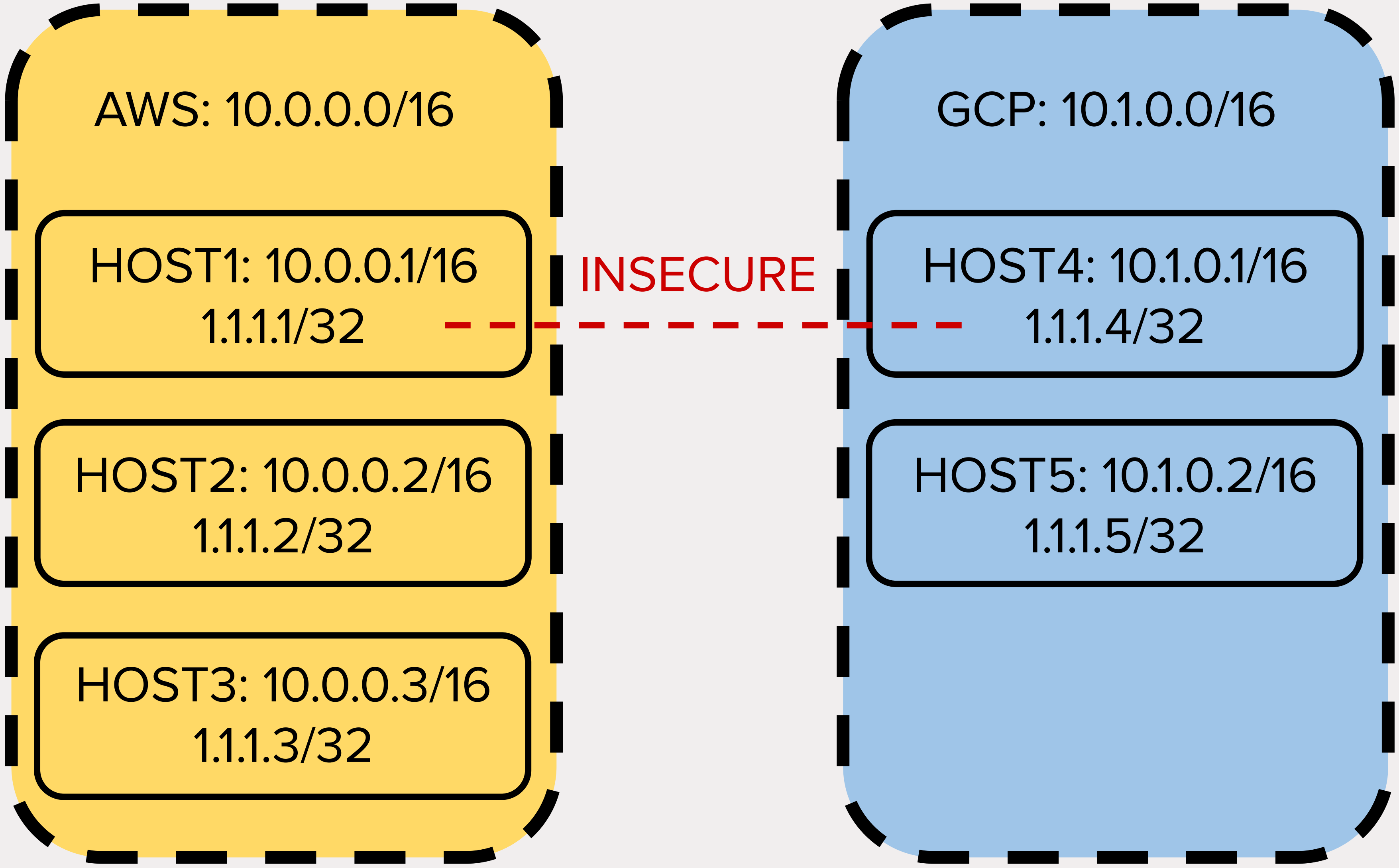
```
$ wg showconf wg0
[Interface]
ListenPort = 51820
PrivateKey = <PRIVATE-KEY>

[Peer]
PublicKey = ABC...
AllowedIPs = 10.4.0.1/32
Endpoint = 1.1.1.1:51820

[Peer]
PublicKey = XYZ...
AllowedIPs = 10.4.0.2/32
Endpoint = 2.2.2.2:51820
```

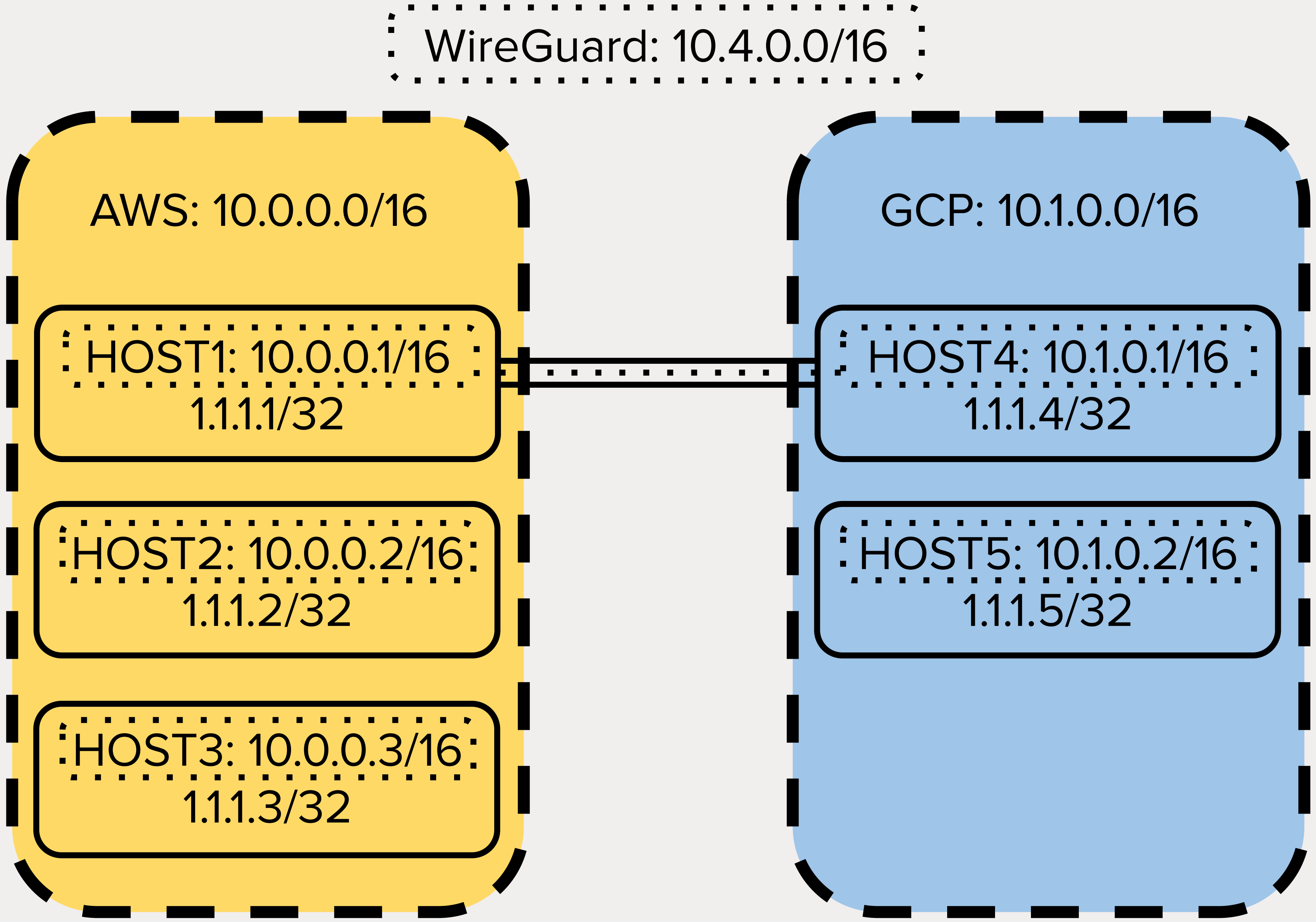
https://www.wireguard.com/papers/wireguard.pdf github.com/squat/kilo

HOW

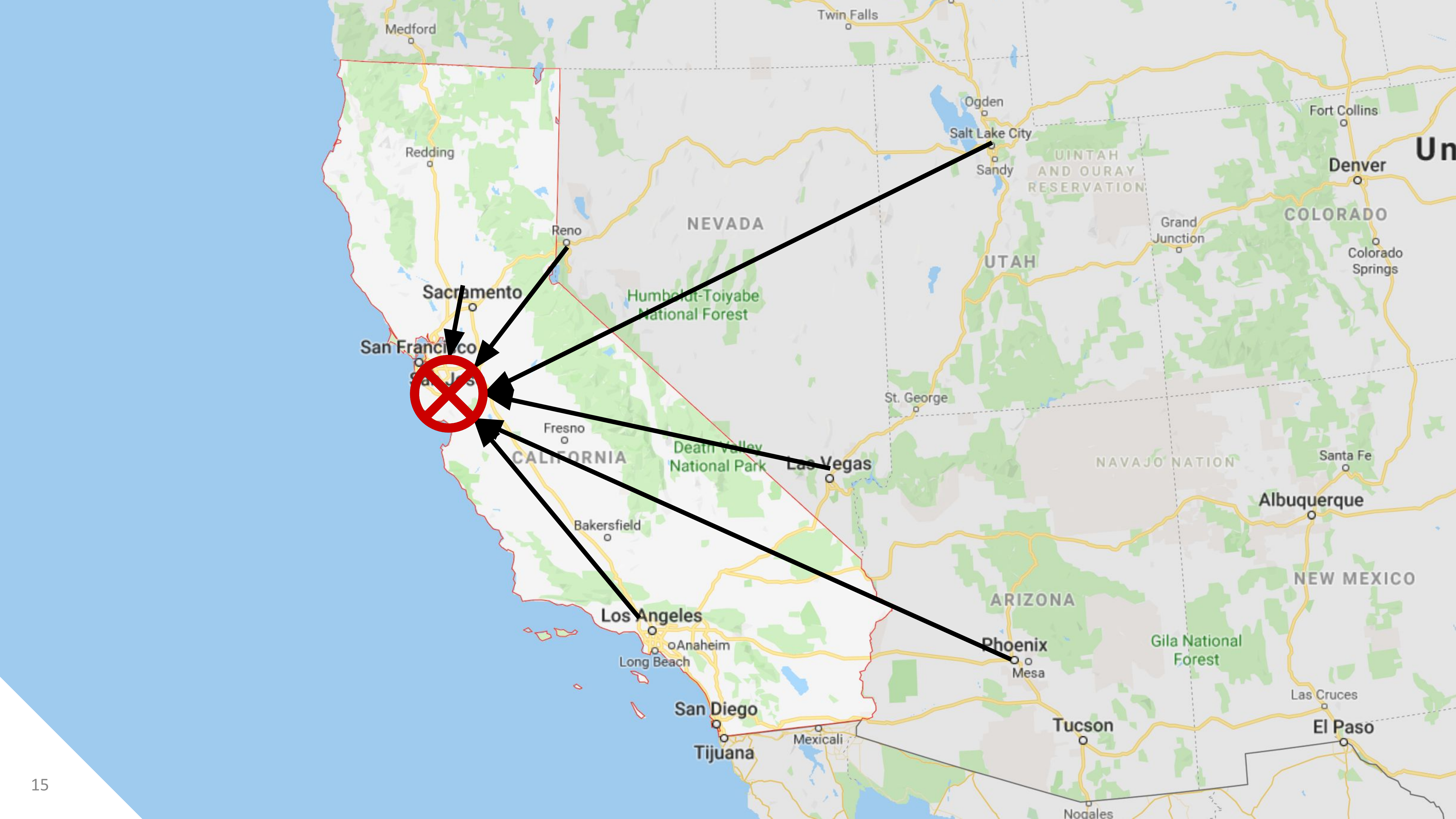

AWS: 10.0.0.0/16
HOST1: 10.0.0.1/16
HOST2: 10.0.0.2/16
HOST3: 10.0.0.3/16

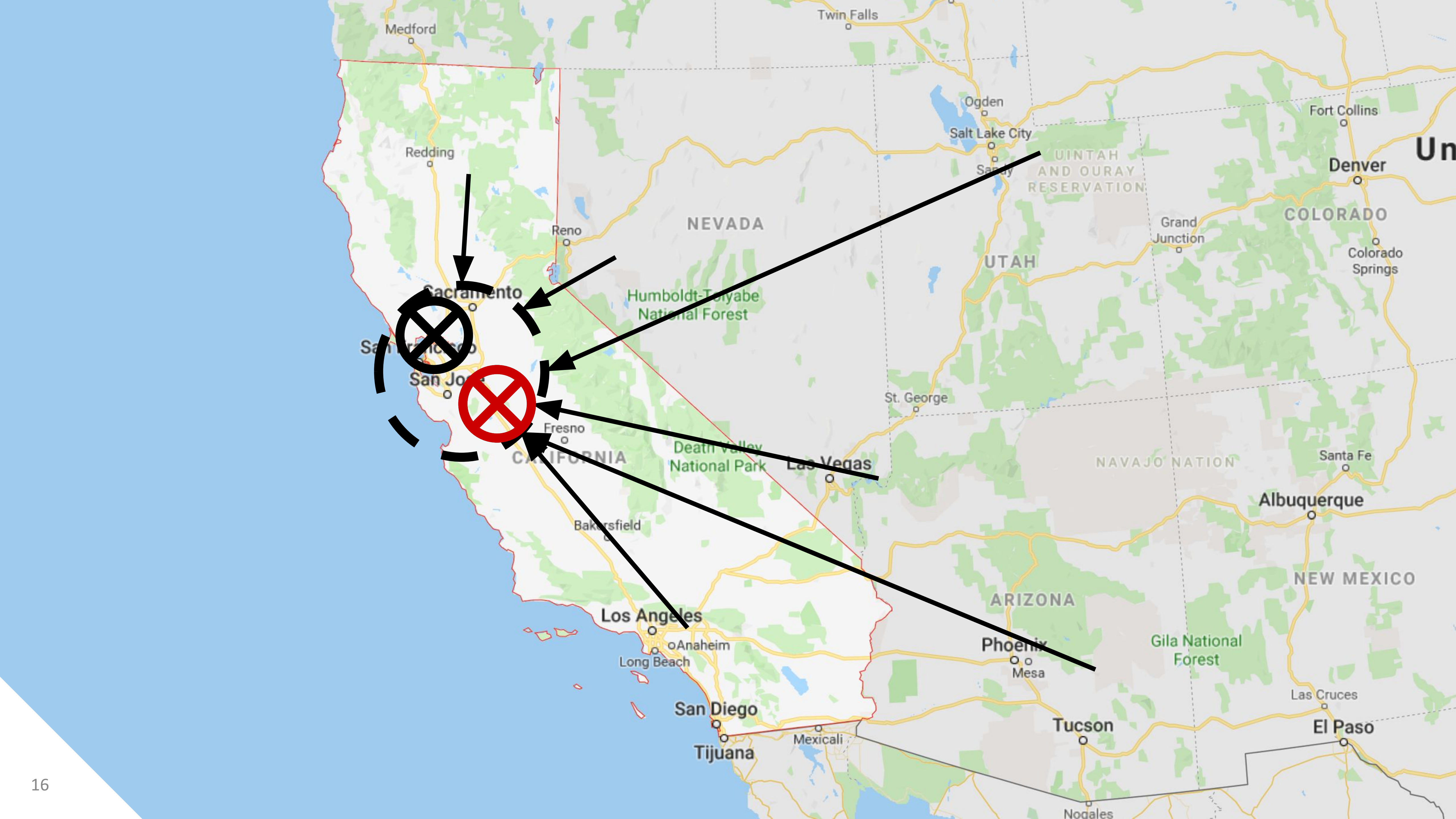

AWS: 10.0.0.0/16
HOST1: 10.0.0.1/16
HOST2: 10.0.0.2/16
HOST3: 10.0.0.3/16
GCP: 10.1.0.0/16
HOST4: 10.1.0.1/16
HOST5: 10.1.0.2/16


AWS: 10.0.0.0/16
HOST1: 10.0.0.1/16
1.1.1.1/32
HOST2: 10.0.0.2/16
1.1.1.2/32
HOST3: 10.0.0.3/16
1.1.1.3/32
INSECURE
GCP: 10.1.0.0/16
HOST4: 10.1.0.1/16
1.1.1.4/32
HOST5: 10.1.0.2/16
1.1.1.5/32

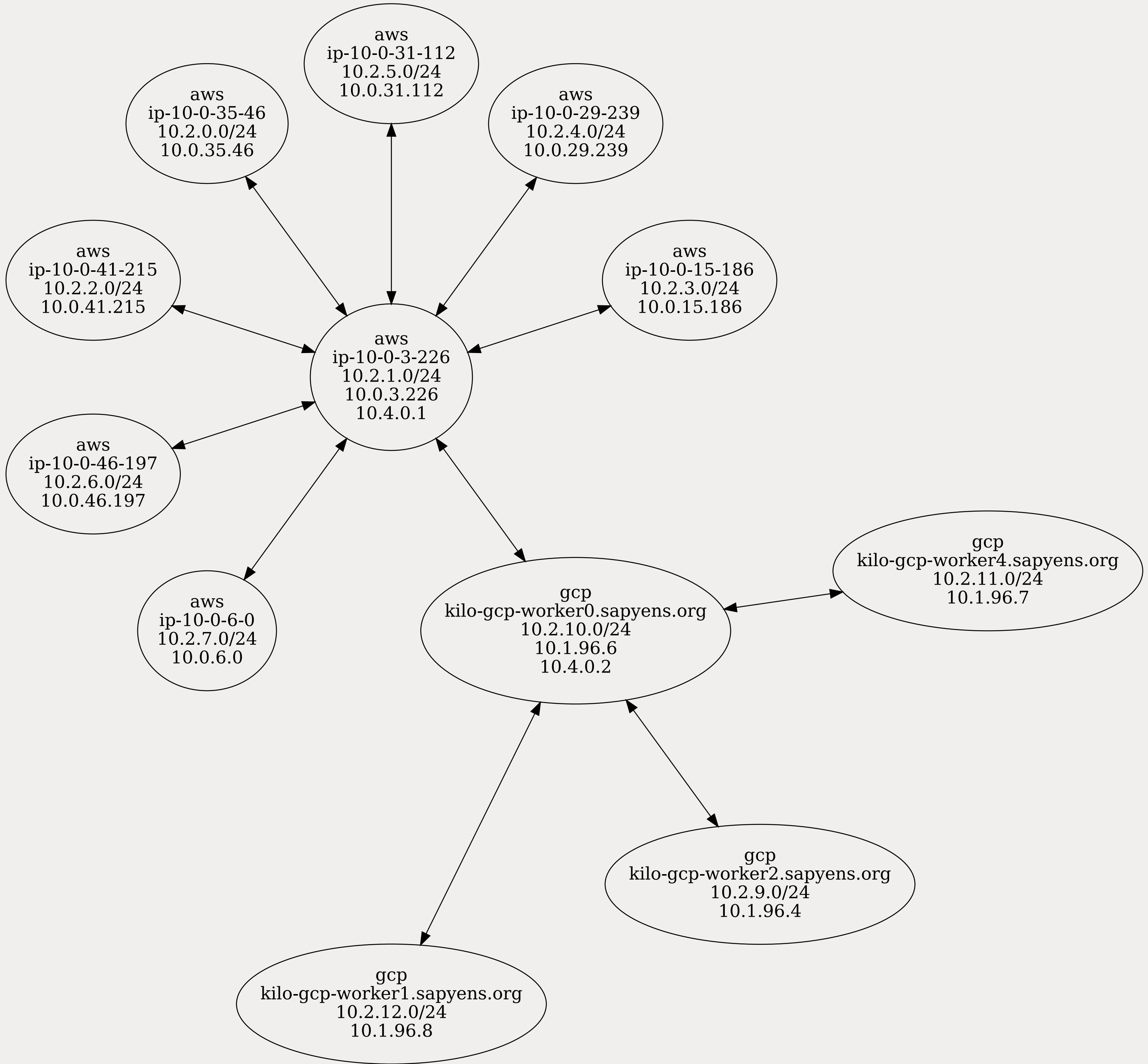

WireGuard: 10.4.0.0/16
AWS: 10.0.0.0/16
HOST1: 10.0.0.1/16
1.1.1.1/32
HOST2: 10.0.0.2/16
1.1.1.2/32
HOST3: 10.0.0.3/16
1.1.1.3/32
GCP: 10.1.0.0/16
HOST4: 10.1.0.1/16
1.1.1.4/32
HOST5: 10.1.0.2/16
1.1.1.5/32

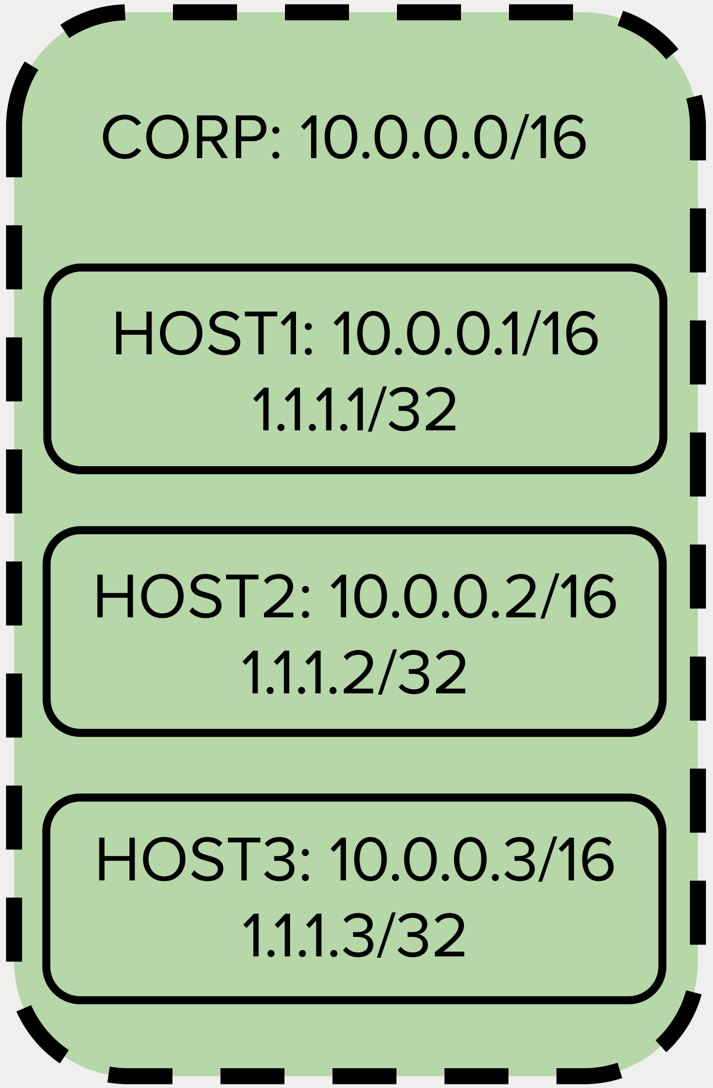
WHY

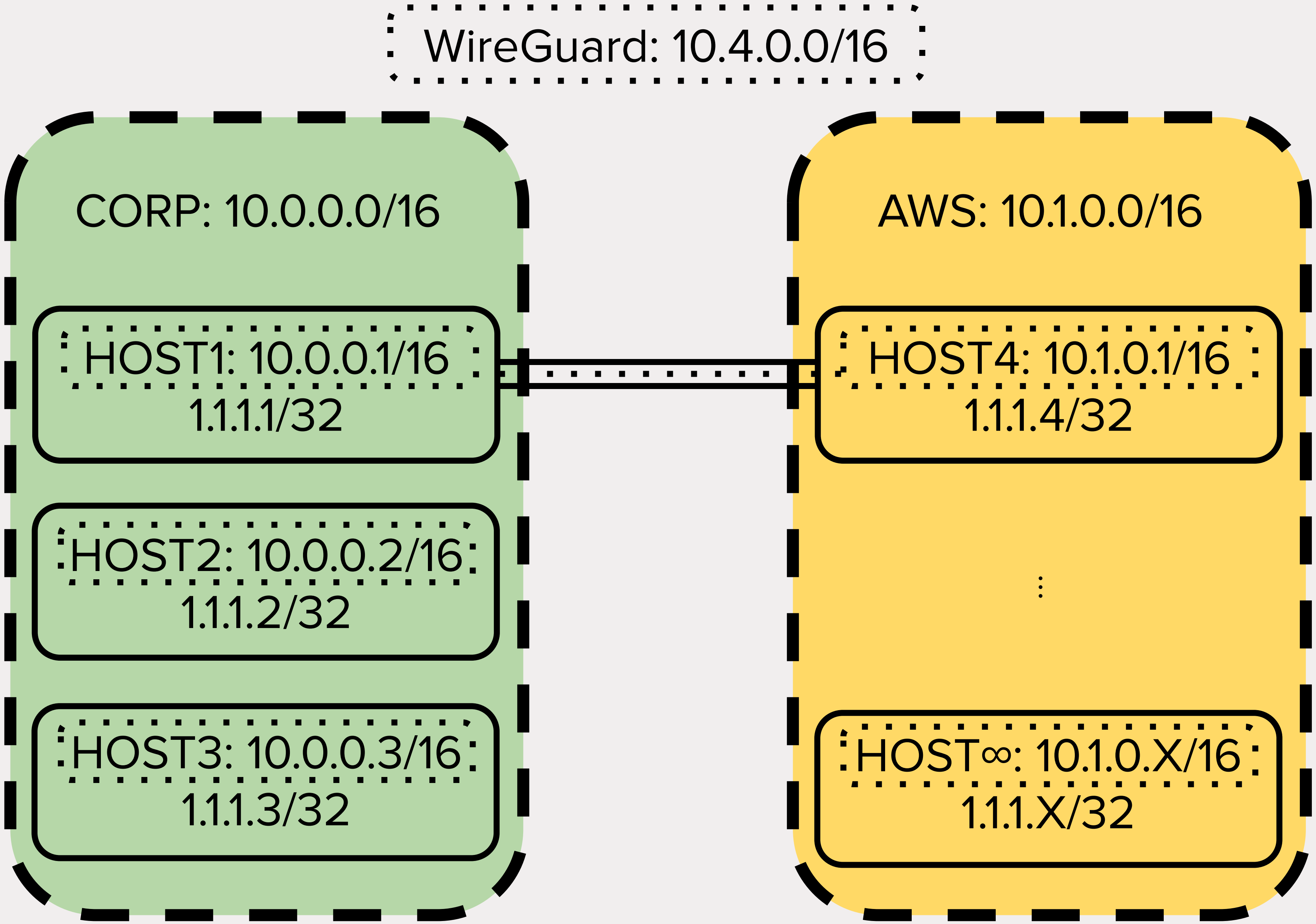

Medford
Twin Falls
Redding
Reno
NEVADA
Ogden
Salt Lake City
Sandy
UINTAH AND OURAY RESERVATION
Fort Collins
Denver
COLORADO
Grand Junction
Colorado Springs
UTAH
Sacramento
Humboldt-Toiyabe National Forest
San Francisco
San Jose
St. George
Fresno
CALIFORNIA
Death Valley National Park
Las Vegas
NAVAJO NATION
Santa Fe
Albuquerque
Bakersfield
NEW MEXICO
ARIZONA
Los Angeles
Anaheim
Long Beach
Phoenix
Mesa
Gila National Forest
San Diego
Tijuana
Mexicali
Tucson
Las Cruces
El Paso
Nogales

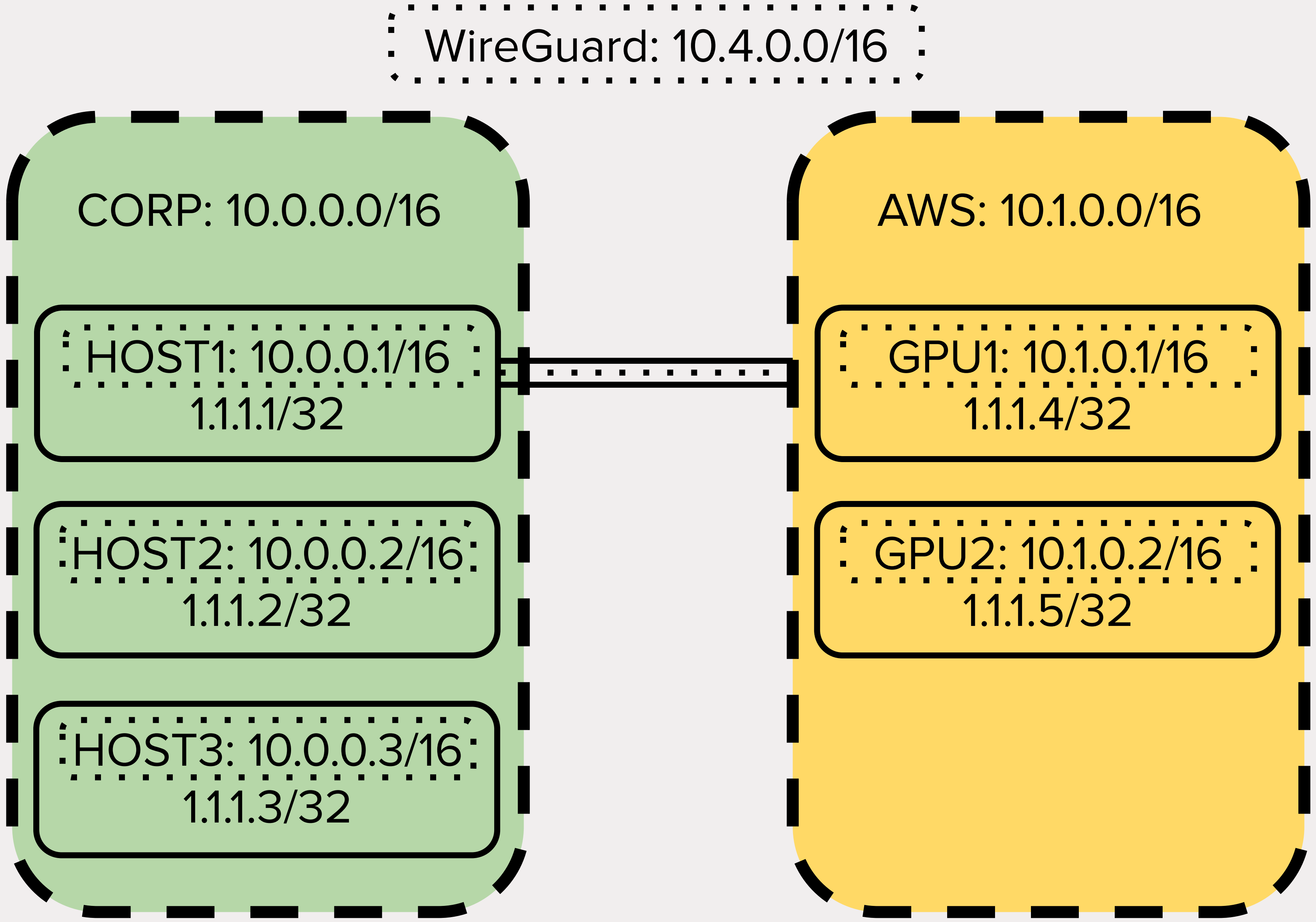

Medford
Twin Falls
Ogden
Salt Lake City
Sandy
Fort Collins
Denver
Redding
Reno
NEVADA
UINTAH AND OURAY RESERVATION
COLORADO
Grand Junction
Colorado Springs
UTAH
Sacramento
Humboldt-Toiyabe National Forest
San Francisco
San Jose
St. George
Fresno
CALIFORNIA
Death Valley National Park
Las Vegas
NAVAJO NATION
Santa Fe
Albuquerque
Bakersfield
NEW MEXICO
ARIZONA
Los Angeles
Anaheim
Long Beach
Phoenix
Mesa
Gila National Forest
San Diego
Tijuana
Mexicali
Tucson
Las Cruces
El Paso
Nogales

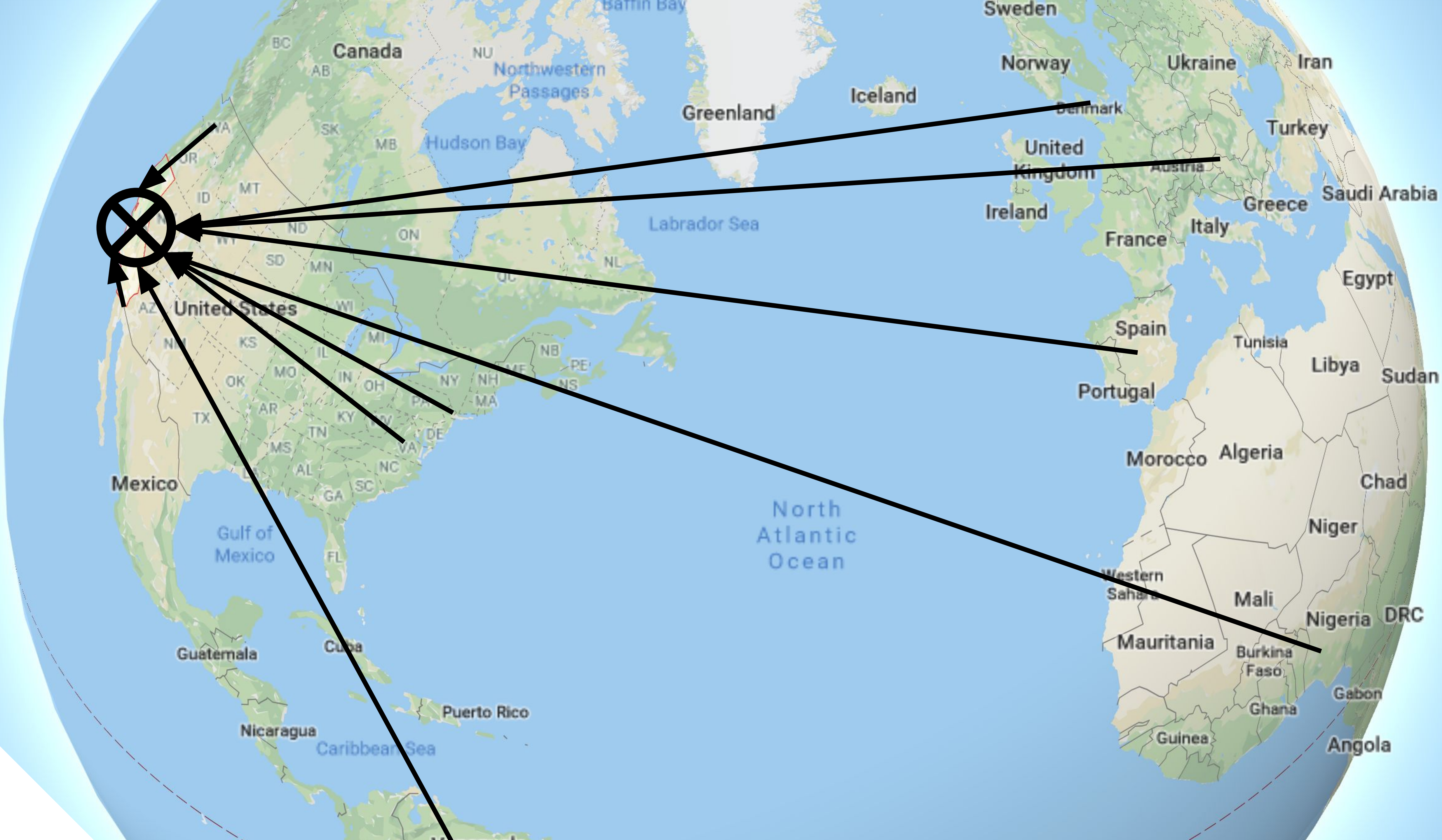
10.4.0.0/16

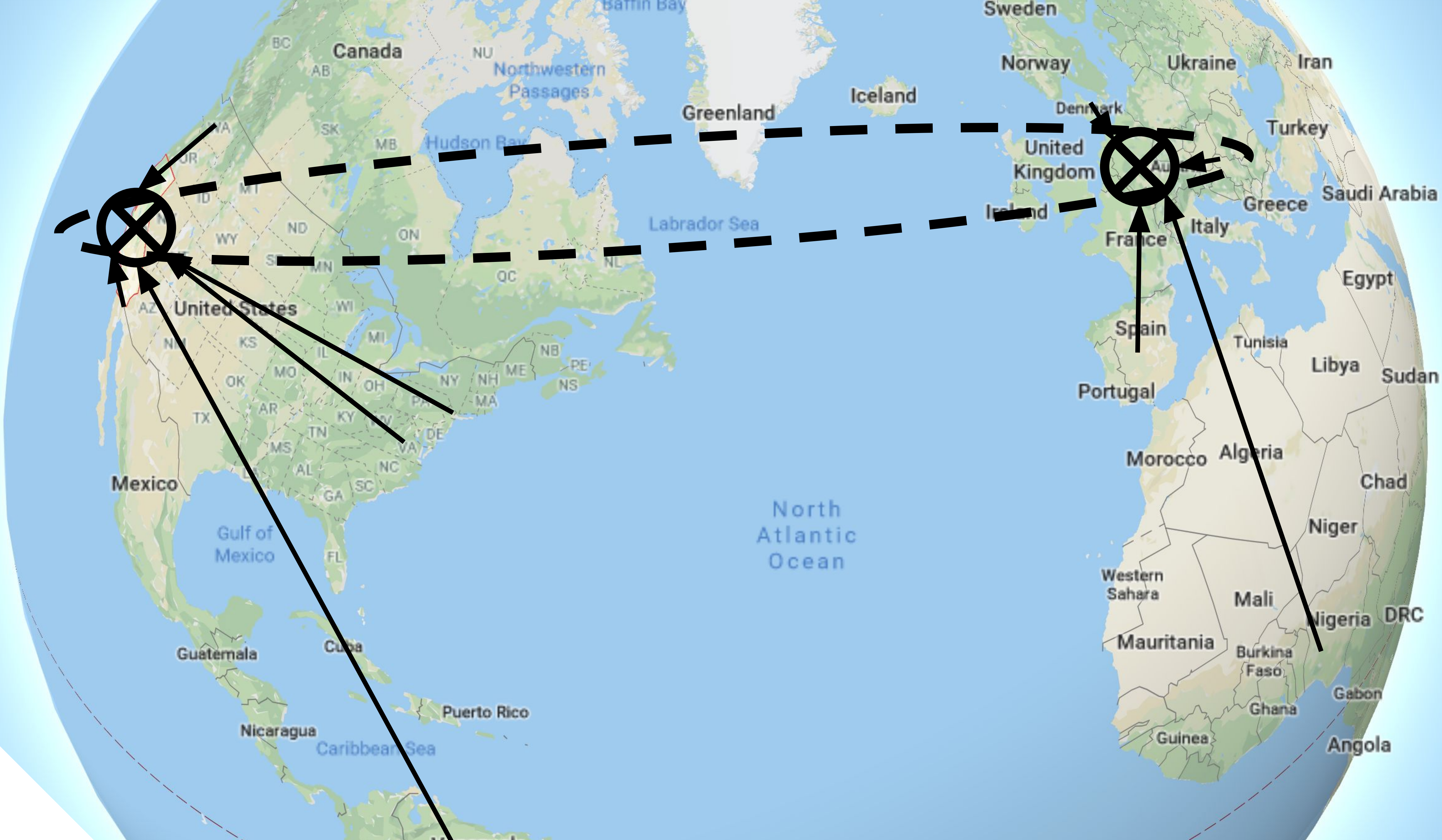

aws
ip-10-0-31-112
10.2.5.0/24
10.0.31.112
aws
ip-10-0-35-46
10.2.0.0/24
10.0.35.46
aws
ip-10-0-29-239
10.2.4.0/24
10.0.29.239
aws
ip-10-0-41-215
10.2.2.0/24
10.0.41.215
aws
ip-10-0-15-186
10.2.3.0/24
10.0.15.186
aws
ip-10-0-3-226
10.2.1.0/24
10.0.3.226
10.4.0.1
aws
ip-10-0-46-197
10.2.6.0/24
10.0.46.197
aws
ip-10-0-6-0
10.2.7.0/24
10.0.6.0
gcp
kilo-gcp-worker0.sapyens.org
10.2.10.0/24
10.1.96.6
10.4.0.2
gcp
kilo-gcp-worker4.sapyens.org
10.2.11.0/24
10.1.96.7
gcp
kilo-gcp-worker2.sapyens.org
10.2.9.0/24
10.1.96.4
gcp
kilo-gcp-worker1.sapyens.org
10.2.12.0/24
10.1.96.8

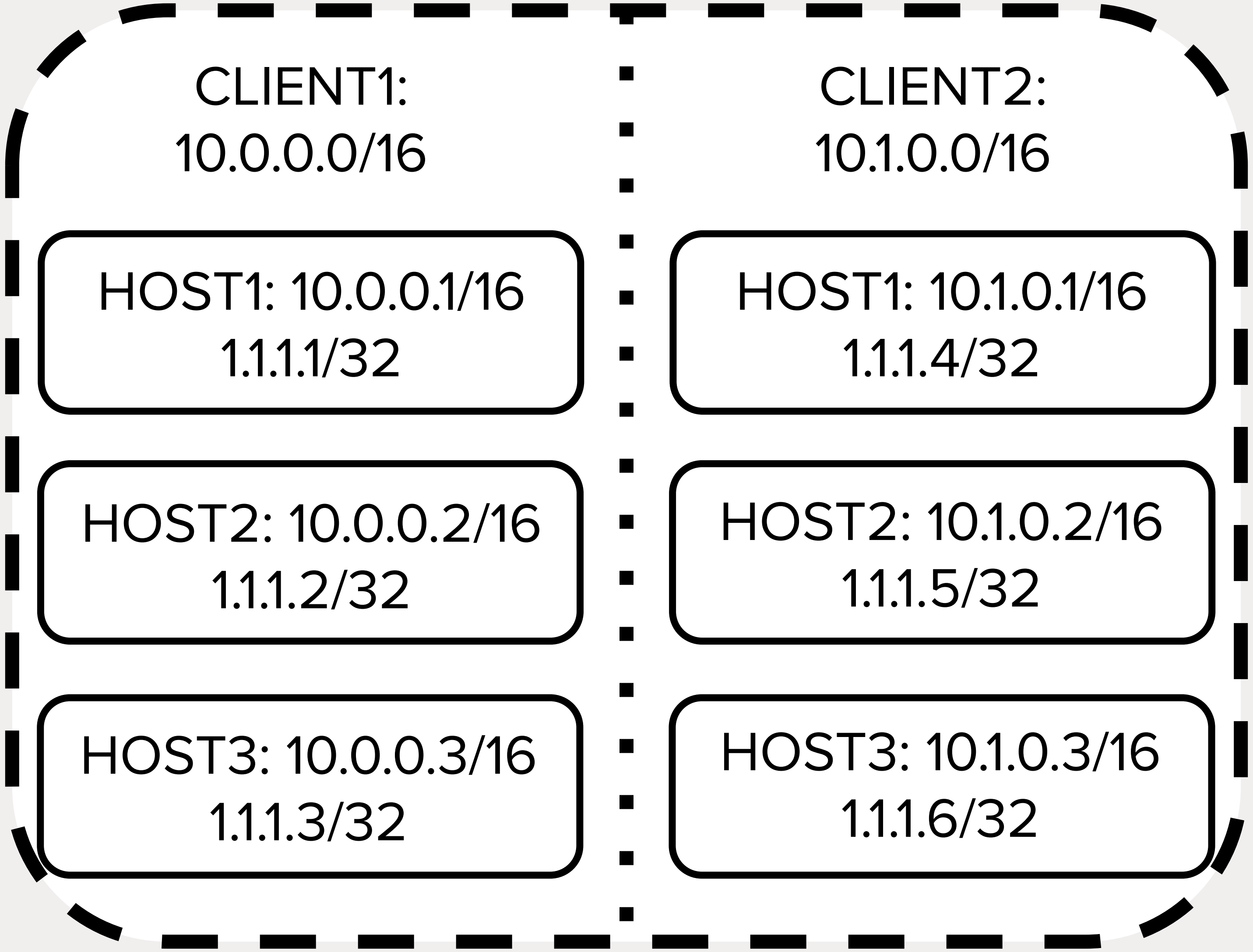

CORP: 10.0.0.0/16
HOST1: 10.0.0.1/16
1.1.1.1/32
HOST2: 10.0.0.2/16
1.1.1.2/32
HOST3: 10.0.0.3/16
1.1.1.3/32

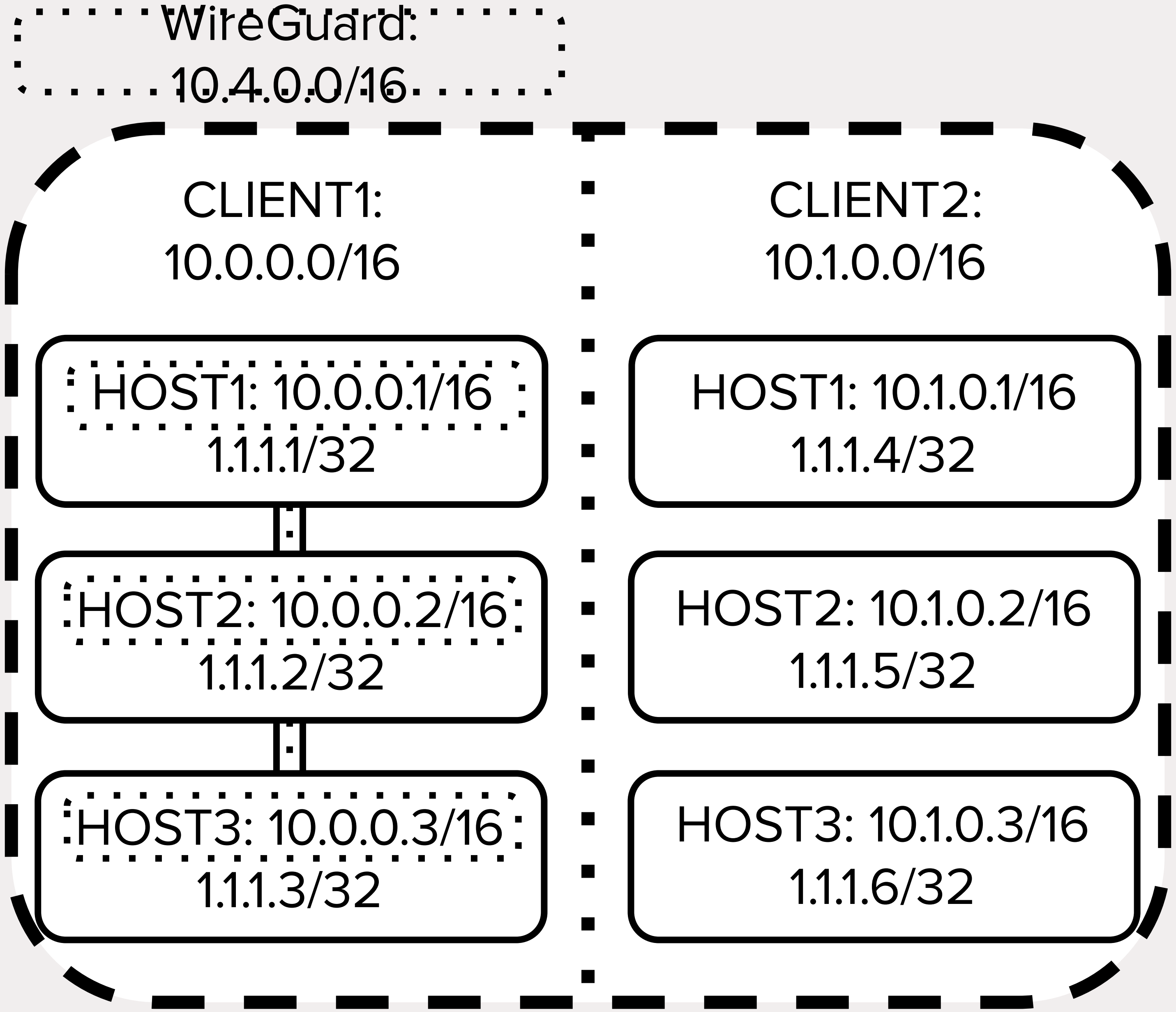

WireGuard: 10.4.0.0/16
CORP: 10.0.0.0/16
HOST1: 10.0.0.1/16
1.1.1.1/32
HOST2: 10.0.0.2/16
1.1.1.2/32
HOST3: 10.0.0.3/16
1.1.1.3/32
AWS: 10.1.0.0/16
HOST4: 10.1.0.1/16
1.1.1.4/32
⋮
HOST∞: 10.1.0.X/16
1.1.1.X/32

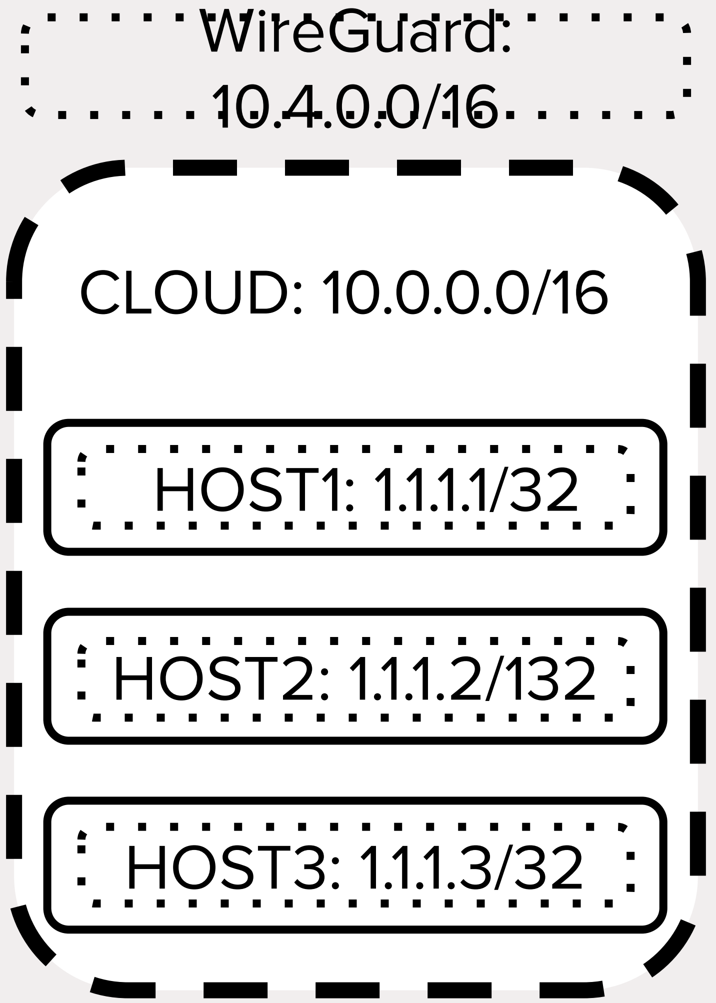

WireGuard: 10.4.0.0/16
CORP: 10.0.0.0/16
HOST1: 10.0.0.1/16
1.1.1.1/32
HOST2: 10.0.0.2/16
1.1.1.2/32
HOST3: 10.0.0.3/16
1.1.1.3/32
AWS: 10.1.0.0/16
GPU1: 10.1.0.1/16
1.1.1.4/32
GPU2: 10.1.0.2/16
1.1.1.5/32


Canada
Greenland
Iceland
Sweden
Norway
Ukraine
Iran
Turkey
United Kingdom
Ireland
France
Italy
Greece
Saudi Arabia
Spain
Portugal
Egypt
Tunisia
Libya
Sudan
Morocco
Algeria
Chad
Niger
Mali
Nigeria
DRC
Mauritania
Burkina Faso
Ghana
Guinea
Gabon
Angola
Western Sahara
Denmark
Austria
Hudson Bay
Labrador Sea
Northwestern Passages
North Atlantic Ocean
United States
Mexico
Gulf of Mexico
Guatemala
Cuba
Puerto Rico
Nicaragua
Caribbean Sea

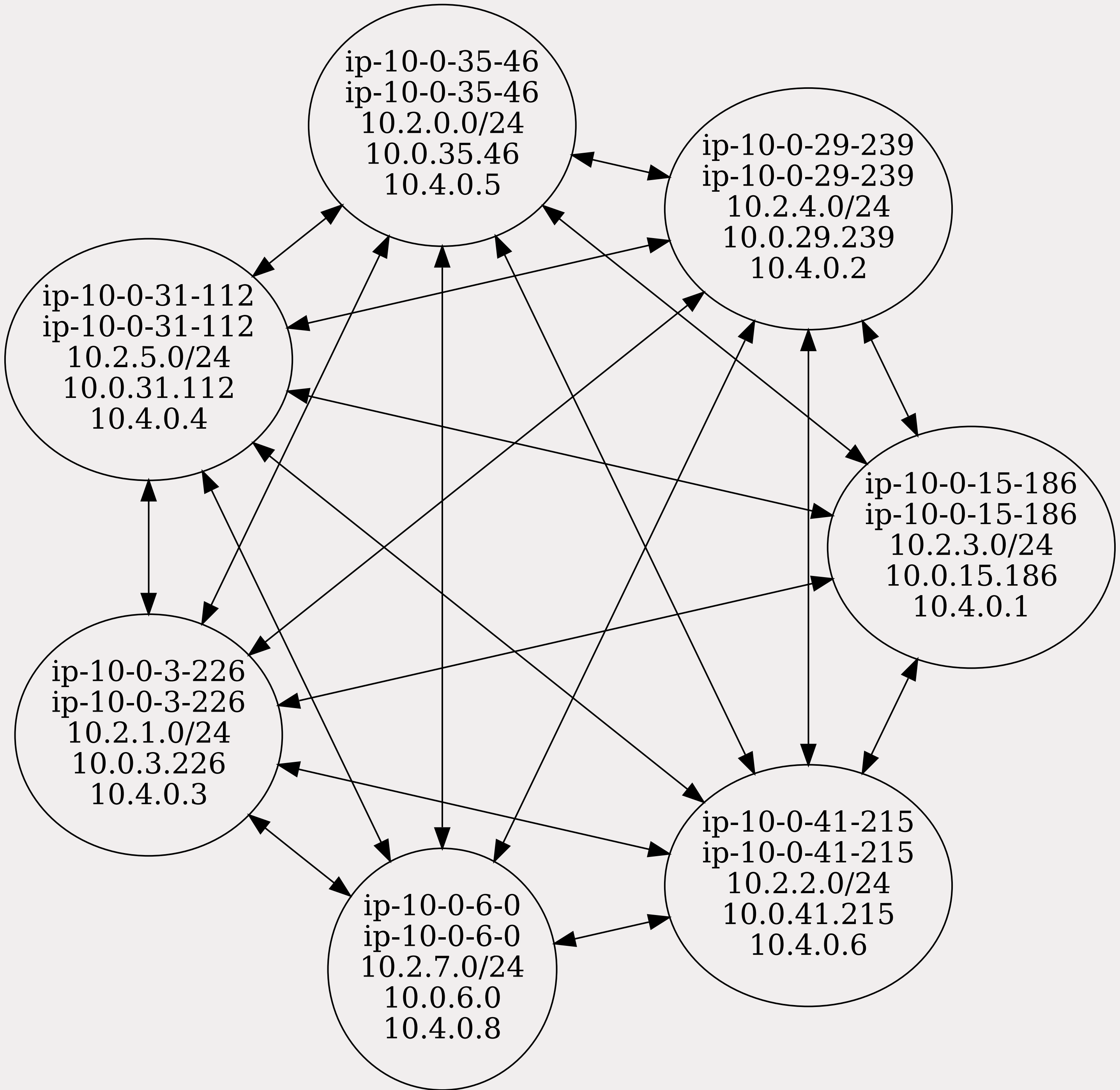

Canada
Greenland
Iceland
Sweden
Norway
Ukraine
Iran
Turkey
United Kingdom
Ireland
France
Italy
Greece
Saudi Arabia
Spain
Portugal
Egypt
Tunisia
Libya
Sudan
Morocco
Algeria
Chad
Niger
Western Sahara
Mali
Mauritania
Nigeria
DRC
Burkina Faso
Gabon
Ghana
Guinea
Angola
Labrador Sea
North Atlantic Ocean
Hudson Bay
Northwestern Passages
United States
Mexico
Gulf of Mexico
Cuba
Guatemala
Puerto Rico
Nicaragua
Caribbean Sea

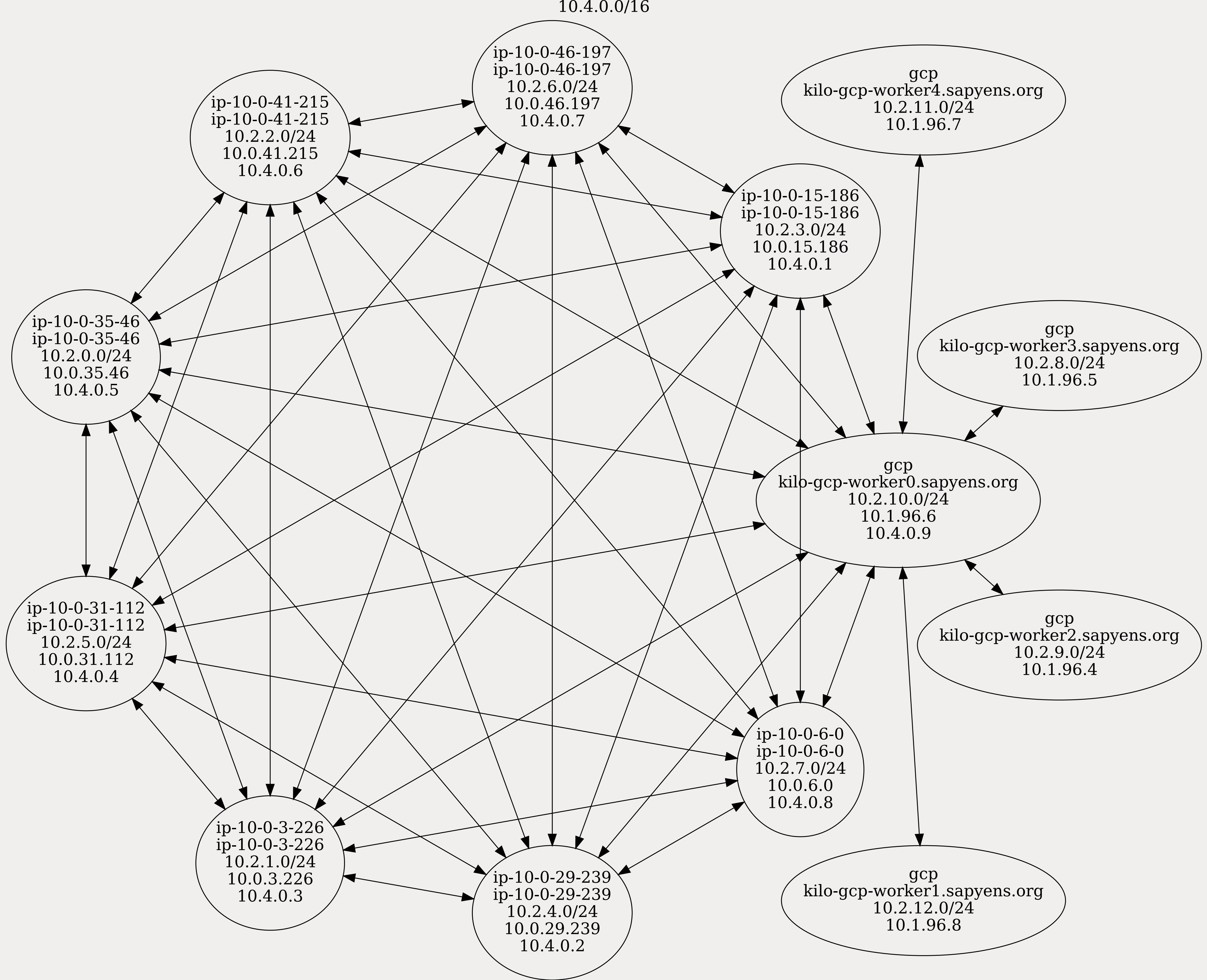

CLIENT1:
10.0.0.0/16
HOST1: 10.0.0.1/16
1.1.1.1/32
HOST2: 10.0.0.2/16
1.1.1.2/32
HOST3: 10.0.0.3/16
1.1.1.3/32
CLIENT2:
10.1.0.0/16
HOST1: 10.1.0.1/16
1.1.1.4/32
HOST2: 10.1.0.2/16
1.1.1.5/32
HOST3: 10.1.0.3/16
1.1.1.6/32


WireGuard:
10.4.0.0/16
CLIENT1:
10.0.0.0/16
HOST1: 10.0.0.1/16
1.1.1.1/32
HOST2: 10.0.0.2/16
1.1.1.2/32
HOST3: 10.0.0.3/16
1.1.1.3/32
CLIENT2:
10.1.0.0/16
HOST1: 10.1.0.1/16
1.1.1.4/32
HOST2: 10.1.0.2/16
1.1.1.5/32
HOST3: 10.1.0.3/16
1.1.1.6/32

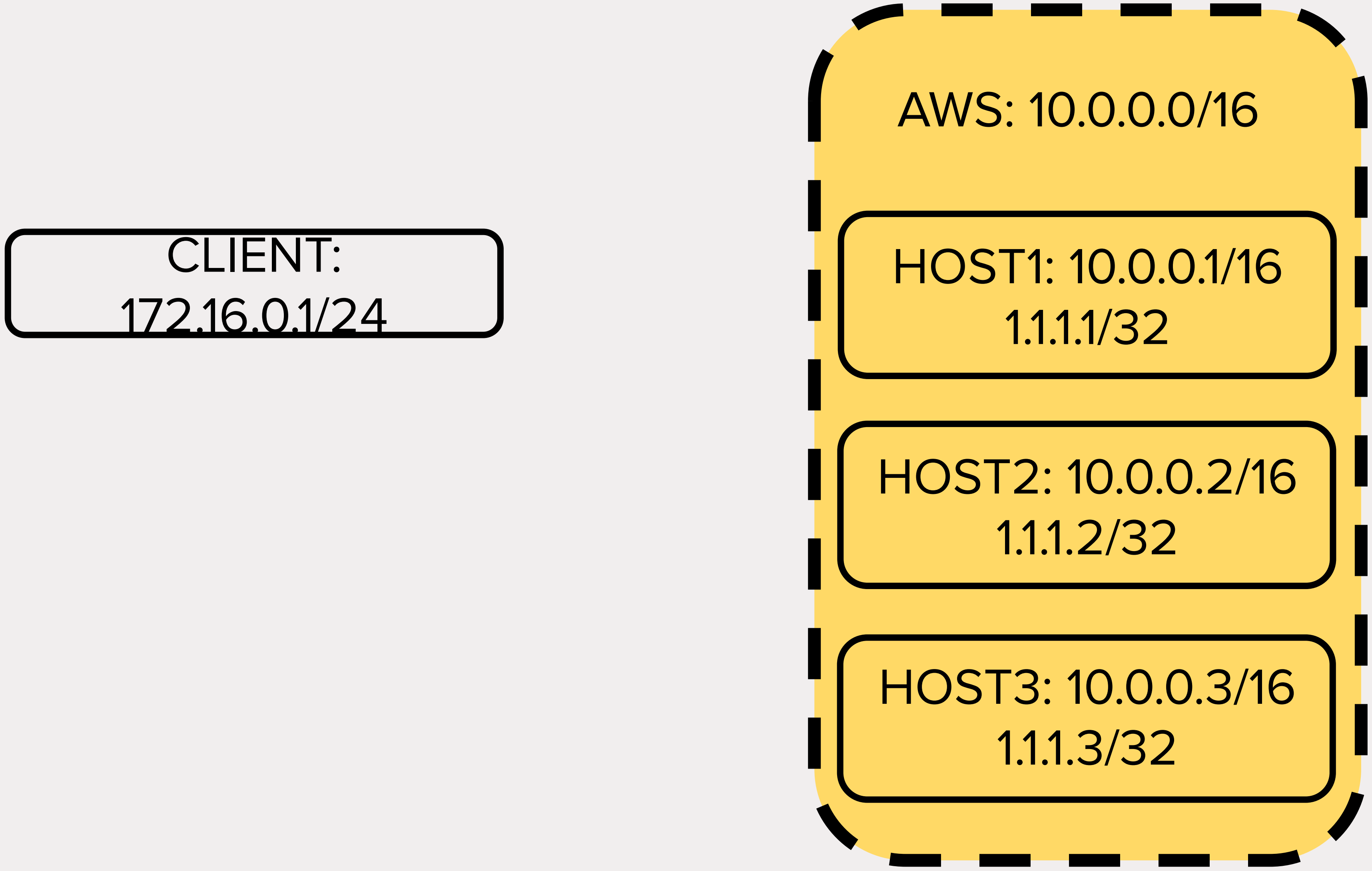

WireGuard:
10.4.0.0/16
CLOUD: 10.0.0.0/16
HOST1: 1.1.1.1/32
HOST2: 1.1.1.2/132
HOST3: 1.1.1.3/32

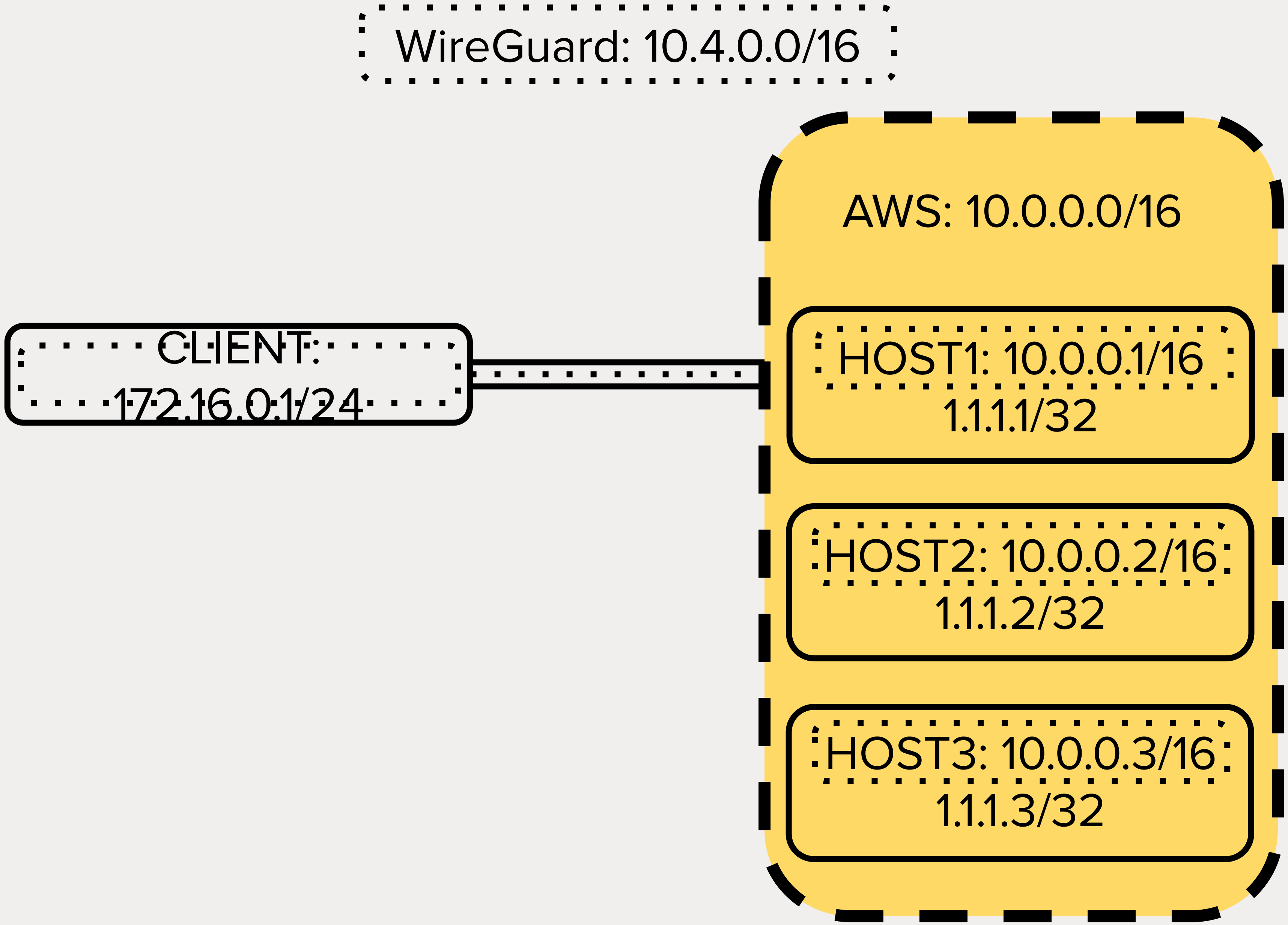
10.4.0.0/16


ip-10-0-35-46
ip-10-0-35-46
10.2.0.0/24
10.0.35.46
10.4.0.5
ip-10-0-29-239
ip-10-0-29-239
10.2.4.0/24
10.0.29.239
10.4.0.2
ip-10-0-31-112
ip-10-0-31-112
10.2.5.0/24
10.0.31.112
10.4.0.4
ip-10-0-15-186
ip-10-0-15-186
10.2.3.0/24
10.0.15.186
10.4.0.1
ip-10-0-3-226
ip-10-0-3-226
10.2.1.0/24
10.0.3.226
10.4.0.3
ip-10-0-41-215
ip-10-0-41-215
10.2.2.0/24
10.0.41.215
10.4.0.6
ip-10-0-6-0
ip-10-0-6-0
10.2.7.0/24
10.0.6.0
10.4.0.8


10.4.0.0/16
ip-10-0-46-197
ip-10-0-46-197
10.2.6.0/24
10.0.46.197
10.4.0.7
ip-10-0-41-215
ip-10-0-41-215
10.2.2.0/24
10.0.41.215
10.4.0.6
gcp
kilo-gcp-worker4.sapyens.org
10.2.11.0/24
10.1.96.7
ip-10-0-15-186
ip-10-0-15-186
10.2.3.0/24
10.0.15.186
10.4.0.1
ip-10-0-35-46
ip-10-0-35-46
10.2.0.0/24
10.0.35.46
10.4.0.5
gcp
kilo-gcp-worker3.sapyens.org
10.2.8.0/24
10.1.96.5
gcp
kilo-gcp-worker0.sapyens.org
10.2.10.0/24
10.1.96.6
10.4.0.9
ip-10-0-31-112
ip-10-0-31-112
10.2.5.0/24
10.0.31.112
10.4.0.4
gcp
kilo-gcp-worker2.sapyens.org
10.2.9.0/24
10.1.96.4
ip-10-0-6-0
ip-10-0-6-0
10.2.7.0/24
10.0.6.0
10.4.0.8
ip-10-0-3-226
ip-10-0-3-226
10.2.1.0/24
10.0.3.226
10.4.0.3
ip-10-0-29-239
ip-10-0-29-239
10.2.4.0/24
10.0.29.239
10.4.0.2
gcp
kilo-gcp-worker1.sapyens.org
10.2.12.0/24
10.1.96.8

ACT 2: Peers


CLIENT:
172.16.0.1/24
AWS: 10.0.0.0/16
HOST1: 10.0.0.1/16
1.1.1.1/32
HOST2: 10.0.0.2/16
1.1.1.2/32
HOST3: 10.0.0.3/16
1.1.1.3/32

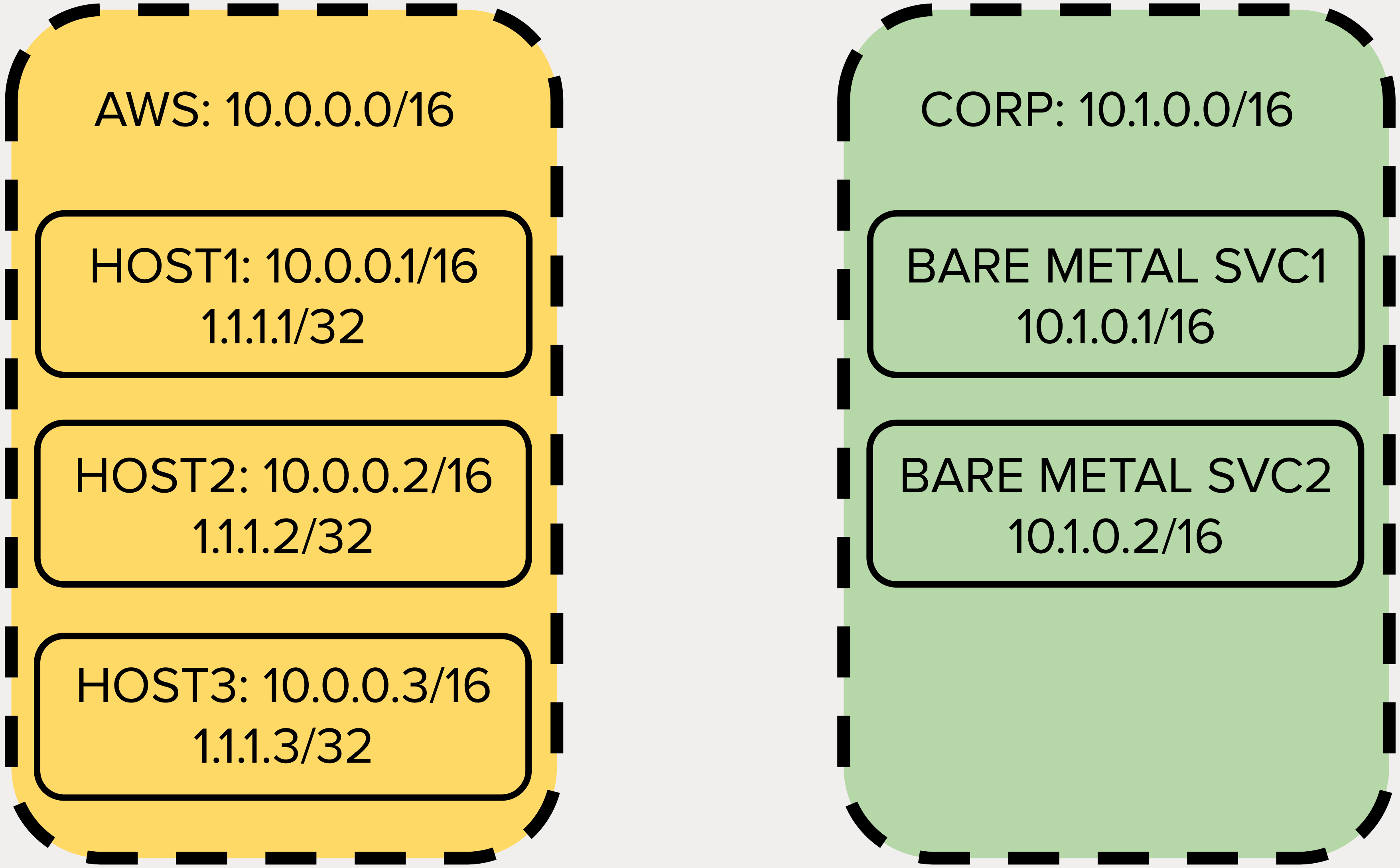

WireGuard: 10.4.0.0/16
AWS: 10.0.0.0/16
CLIENT:
172.16.0.1/24
HOST1: 10.0.0.1/16
1.1.1.1/32
HOST2: 10.0.0.2/16
1.1.1.2/32
HOST3: 10.0.0.3/16
1.1.1.3/32

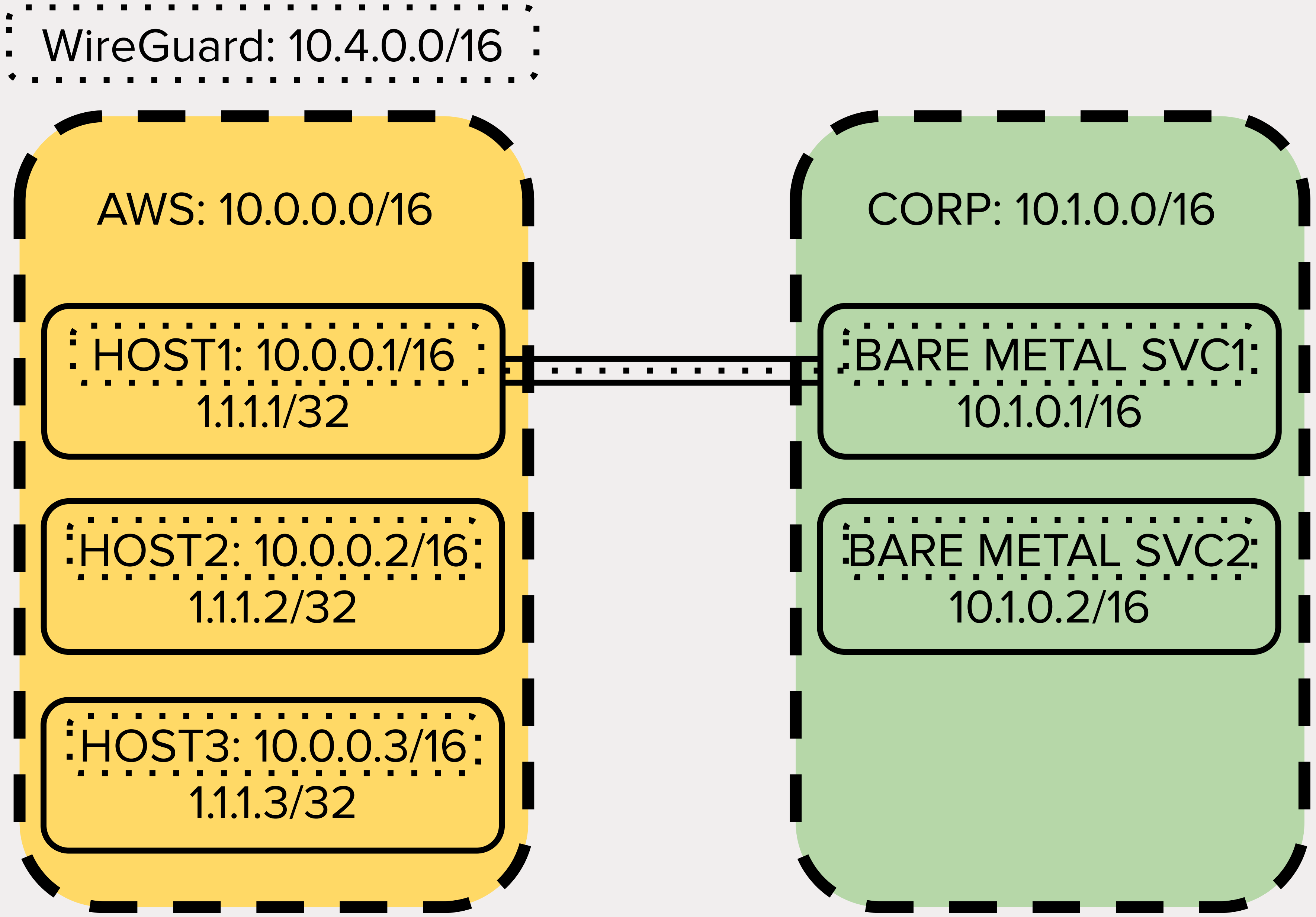
```
$ cat <<'EOF' | kubectl apply -f -
apiVersion: kilo.squat.ai/v1alpha1
kind: Peer
metadata:
  name: client
spec:
  allowedIPs:
  - 10.5.0.1/32
  publicKey: ABC...
EOF
```

```
$ kgctl showconf peer client
```

```
[Peer]
PublicKey = ABC...
AllowedIPs = 10.4.0.2/32,
10.2.0.0/16, 10.0.0.1/32, 10.0.0.2/32,
10.0.0.3/32
Endpoint = 1.1.1.1:51820
```

```
$ curl http://10.2.0.1
```

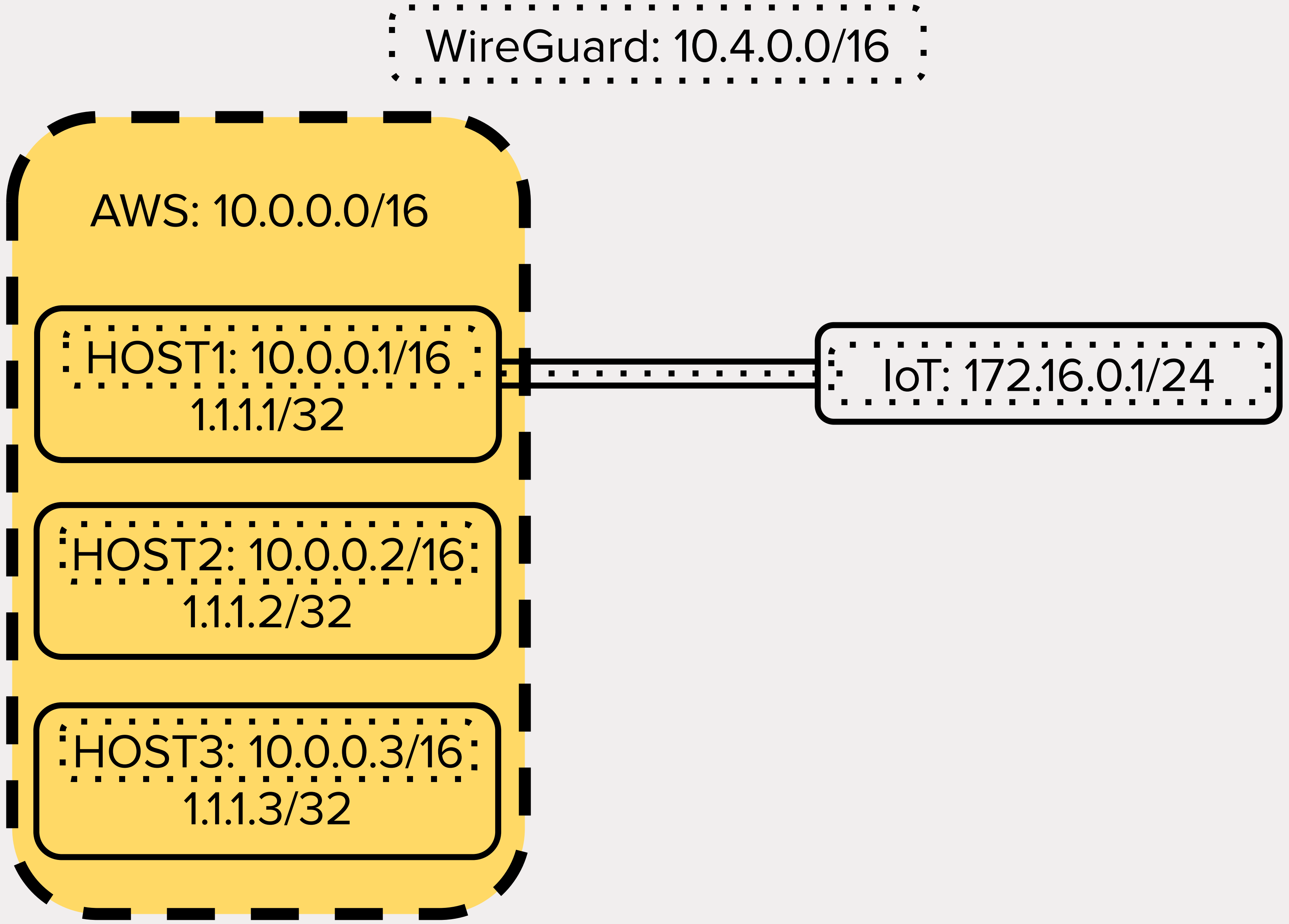

AWS: 10.0.0.0/16
HOST1: 10.0.0.1/16
1.1.1.1/32
HOST2: 10.0.0.2/16
1.1.1.2/32
HOST3: 10.0.0.3/16
1.1.1.3/32
CORP: 10.1.0.0/16
BARE METAL SVC1
10.1.0.1/16
BARE METAL SVC2
10.1.0.2/16


WireGuard: 10.4.0.0/16
AWS: 10.0.0.0/16
HOST1: 10.0.0.1/16
1.1.1.1/32
HOST2: 10.0.0.2/16
1.1.1.2/32
HOST3: 10.0.0.3/16
1.1.1.3/32
CORP: 10.1.0.0/16
BARE METAL SVC1:
10.1.0.1/16
BARE METAL SVC2:
10.1.0.2/16

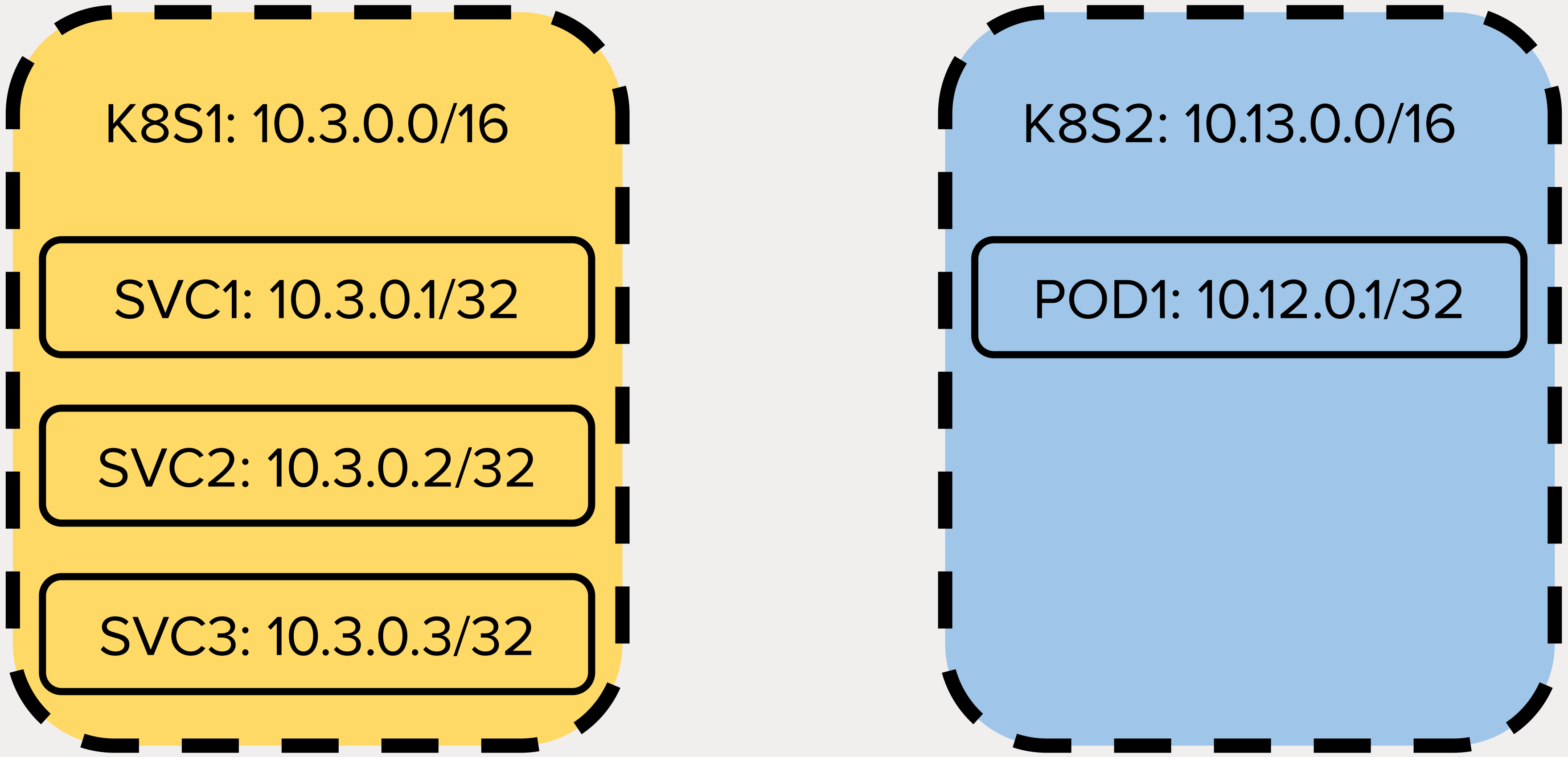
```
$ cat <<'EOF' | kubectl apply -f -
apiVersion: v1
kind: Endpoints
metadata:
    name: bare-metal-1
subsets:
  - addresses:
      - ip: 10.1.0.1
    ports:
      - port: 80
EOF
```

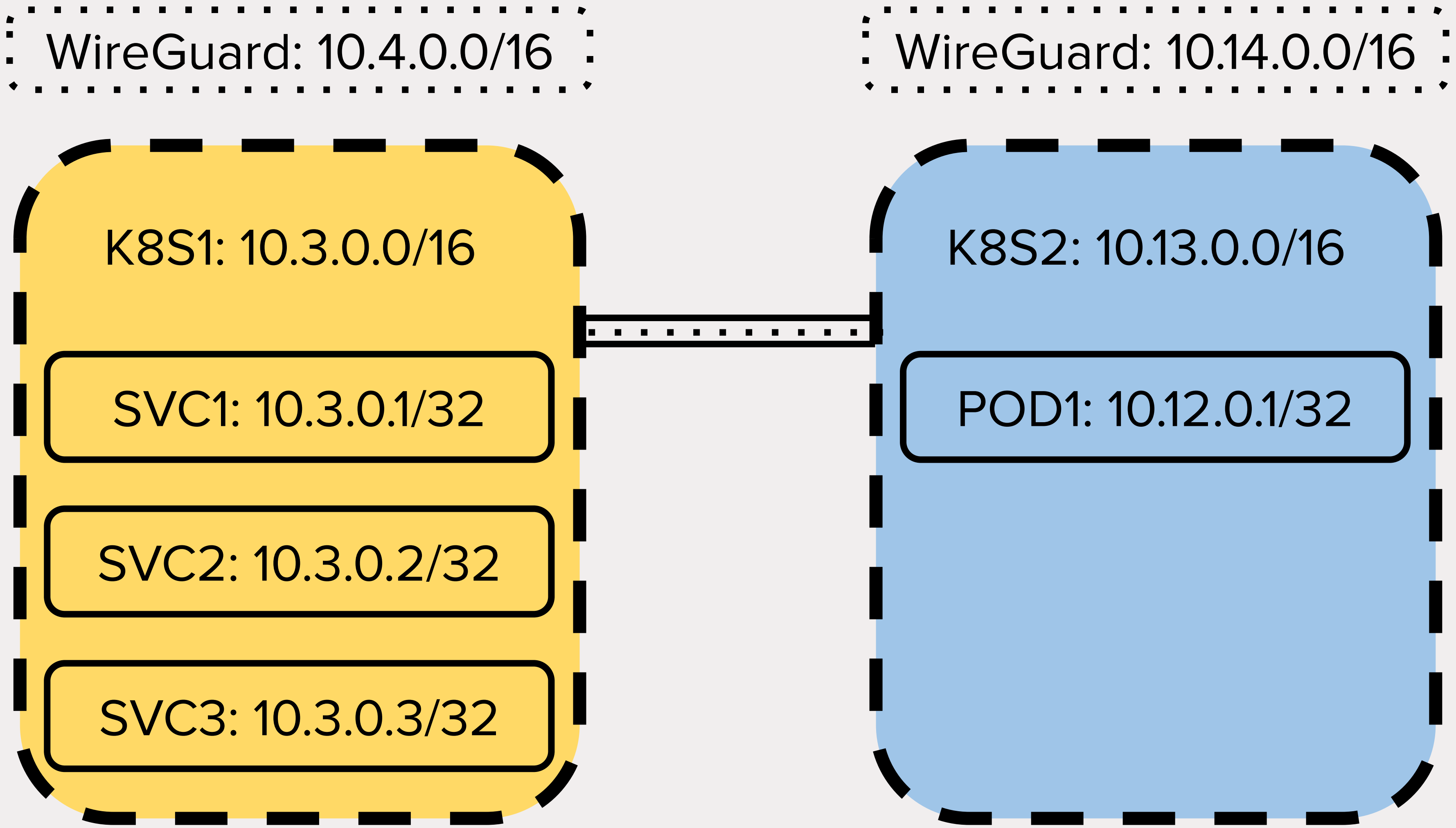
```
$ cat <<'EOF' | kubectl apply -f -
apiVersion: v1
kind: Service
metadata:
  name: bare-metal-1
spec:
  ports:
    - port: 80
EOF
```

```
$ curl http://bare-metal-1.default.svc.cluster.local
```


WireGuard: 10.4.0.0/16
AWS: 10.0.0.0/16
HOST1: 10.0.0.1/16
1.1.1.1/32
HOST2: 10.0.0.2/16
1.1.1.2/32
HOST3: 10.0.0.3/16
1.1.1.3/32
IoT: 172.16.0.1/24

ACT 3:
Multi-Cluster


K8S1: 10.3.0.0/16
SVC1: 10.3.0.1/32
SVC2: 10.3.0.2/32
SVC3: 10.3.0.3/32
K8S2: 10.13.0.0/16
POD1: 10.12.0.1/32


WireGuard: 10.4.0.0/16
WireGuard: 10.14.0.0/16
K8S1: 10.3.0.0/16
SVC1: 10.3.0.1/32
SVC2: 10.3.0.2/32
SVC3: 10.3.0.3/32
K8S2: 10.13.0.0/16
POD1: 10.12.0.1/32

DEMO

github.com/squat/kubeconeu2019

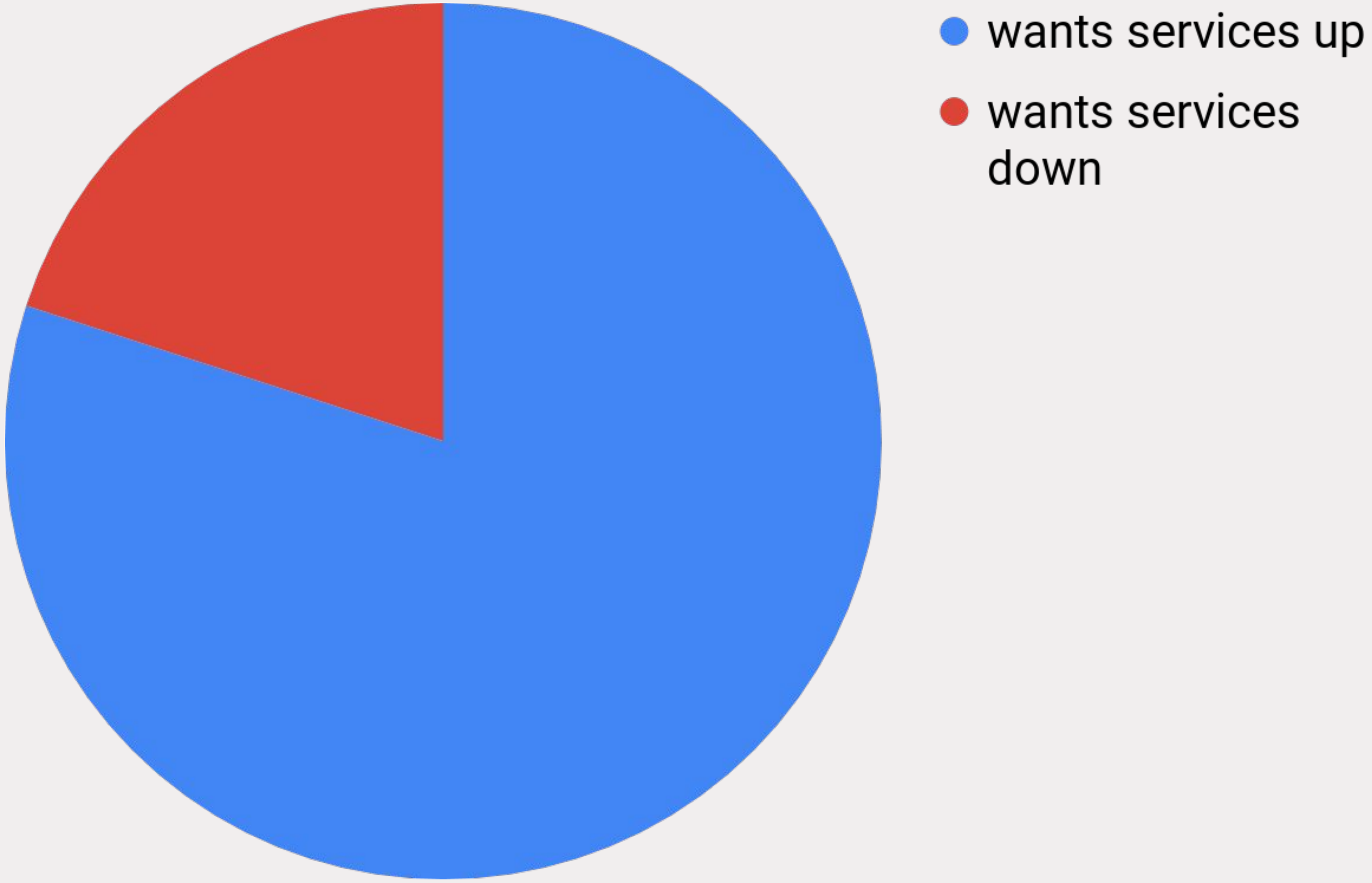
FAQ

1. What are latencies like?
2. Automatic service replication?
3. How does this compare to X multi-cluster?
4. Can peers be authorized?
5. How can I get started?
6. How can I ensure kube-proxy doesn’t load balance to pods in another region?

Attribution:

- icons: https://fontawesome.com/license
- WireGuard logo: https://www.wireguard.com/#license

```
$ for n in $(kubectl --kubeconfig $KUBECONFIG1 get no -o name
| cut -d'/' -f2); do
    kgctl --kubeconfig $KUBECONFIG1 showconf node $n --as-peer
-o yaml --allowed-ips $SERVICECIDR1 | kubectl --kubeconfig
KUBECONFIG2 apply -f -
done

$ for n in $(kubectl --kubeconfig $KUBECONFIG2 get no -o name
| cut -d'/' -f2); do
    kgctl --kubeconfig $KUBECONFIG2 showconf node $n --as-peer
-o yaml --allowed-ips $SERVICECIDR2 | kubectl --kubeconfig
KUBECONFIG1 apply -f -
done
```


● wants services up
● wants services down