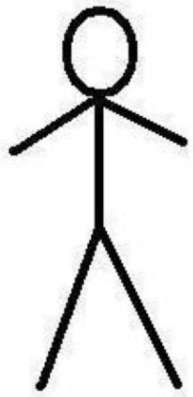
KubeCon | CloudNativeCon

Europe 2018

Cloud native container networking in AWS using CNI plugins

Anirudh Aithal – Sr SDE, AWS

$ whoami



Anirudh Aithal

aaithal

I'm a developer with Amazon Elastic Container Services (ECS)

@aws ECS

Seattle, WA

Overview | Repositories 15 | Stars 42 | Foll

Popular repositories

amazon-ecs-agent
Forked from aws/amazon-ecs-agent

Amazon EC2 Container Service Agent

Go ★ 9 1

cni
Forked from containernetworking/cni

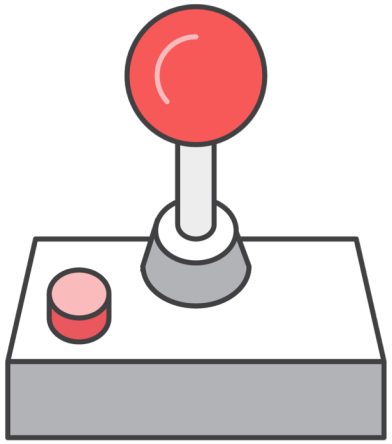
Container Network Interface - networking for Linux containers

Go ★ 1

Topics covered in today's talk

- Networking requirements for containerized applications
- Container networking primitives
- Developing (Amazon VPC) CNI plugins

What do applications need?

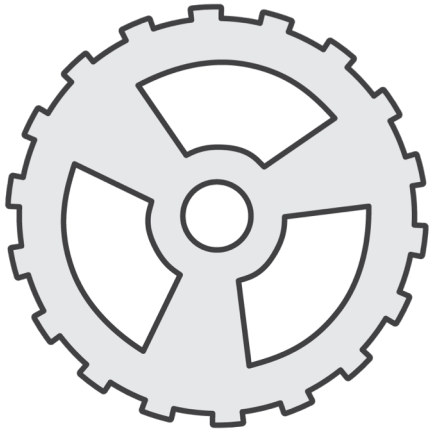


Usability

- Simple abstractions
- Discovery

Security

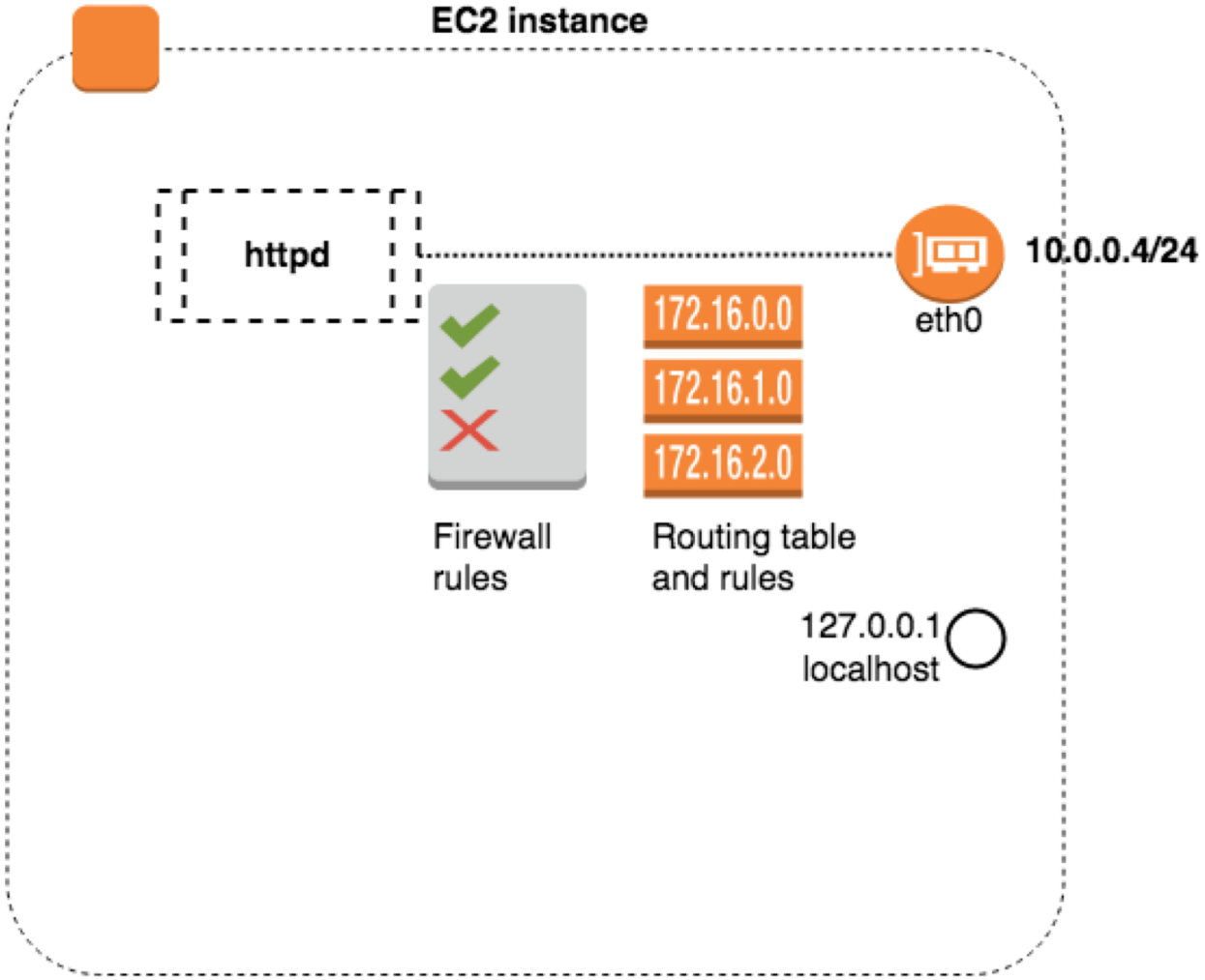
- Network isolation
- Access control
- Auditability



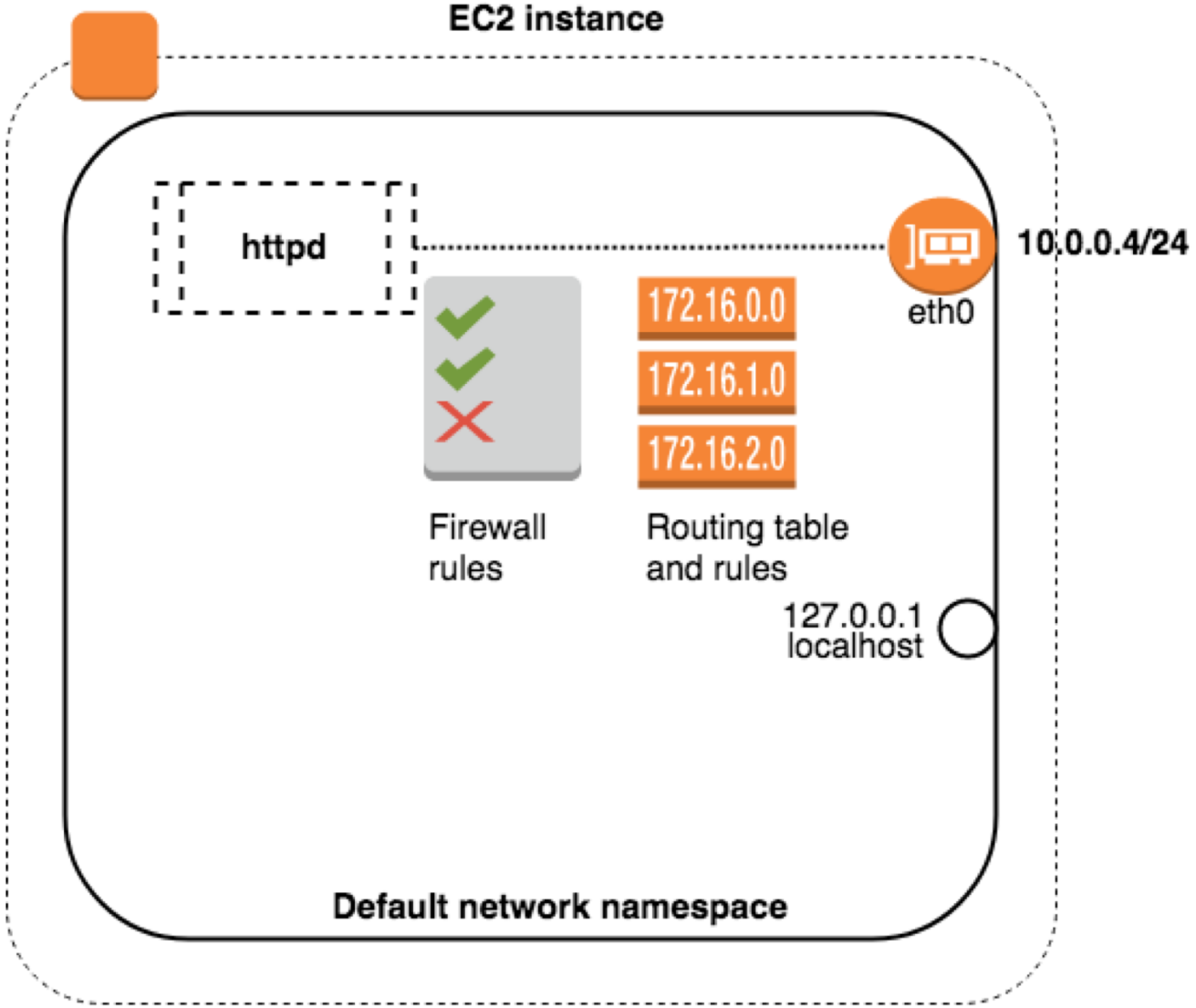
Maintainability

- Scalability
- Performance
- Monitoring

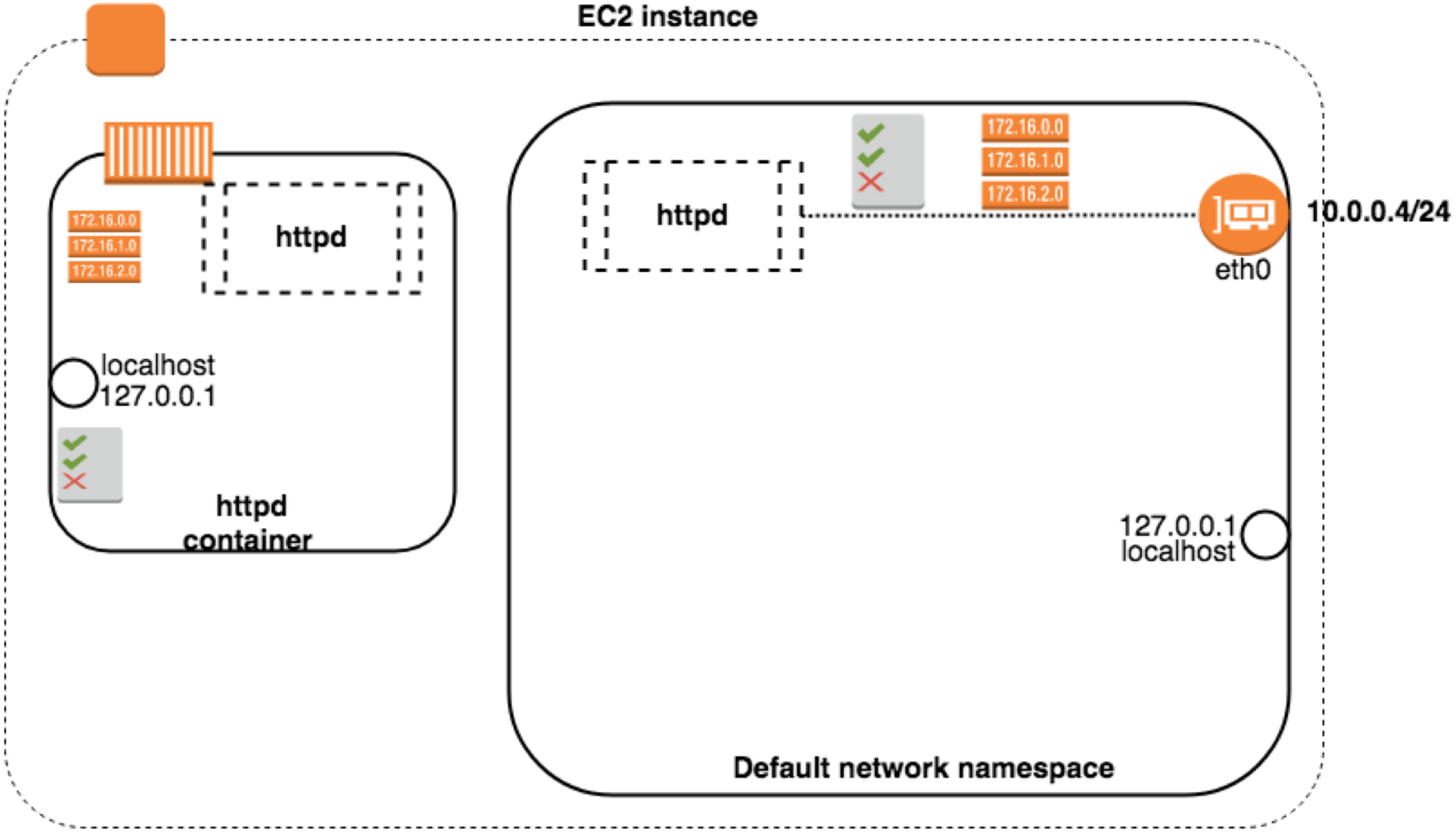
One listener : one host



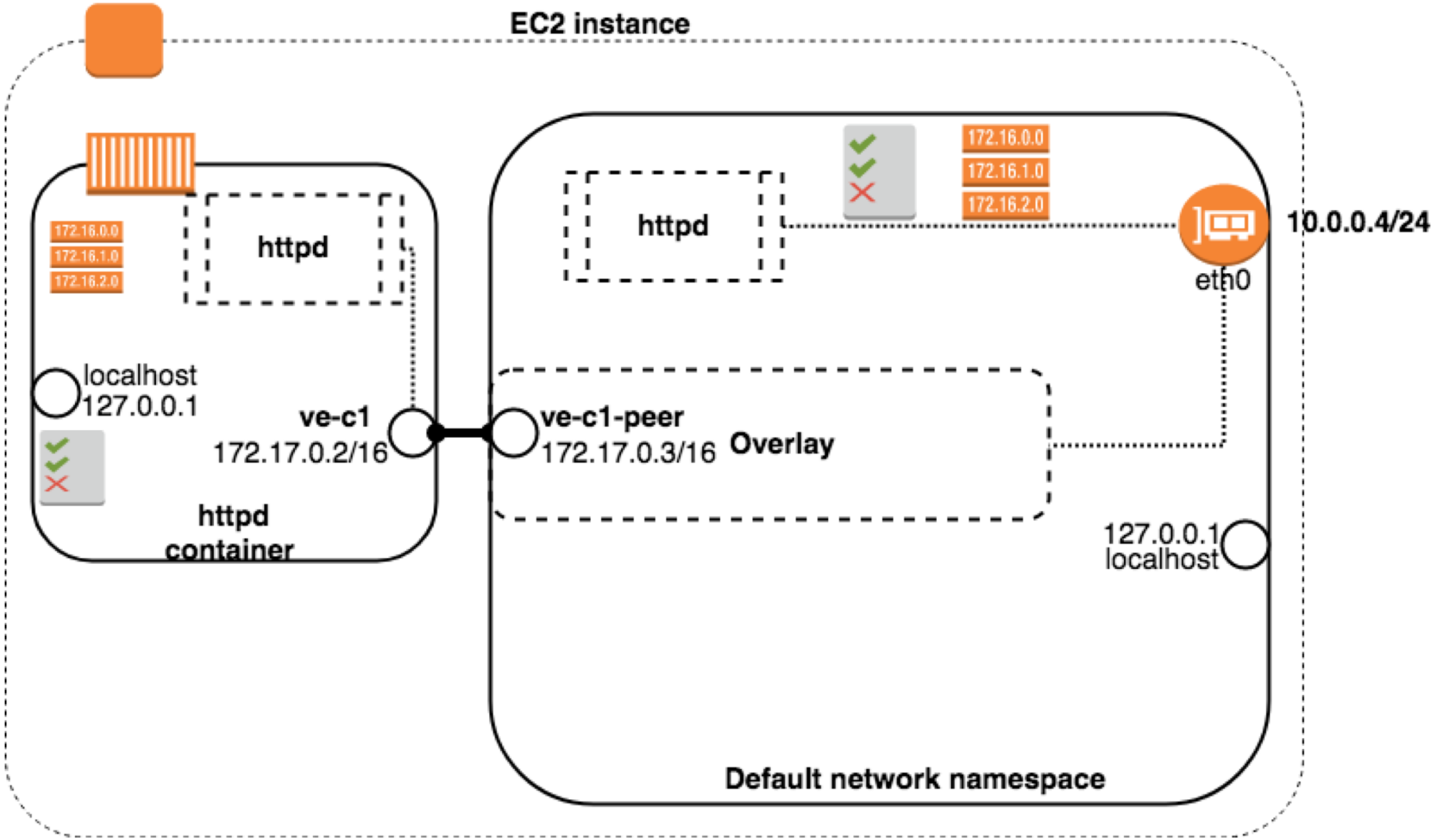
One listener : one host



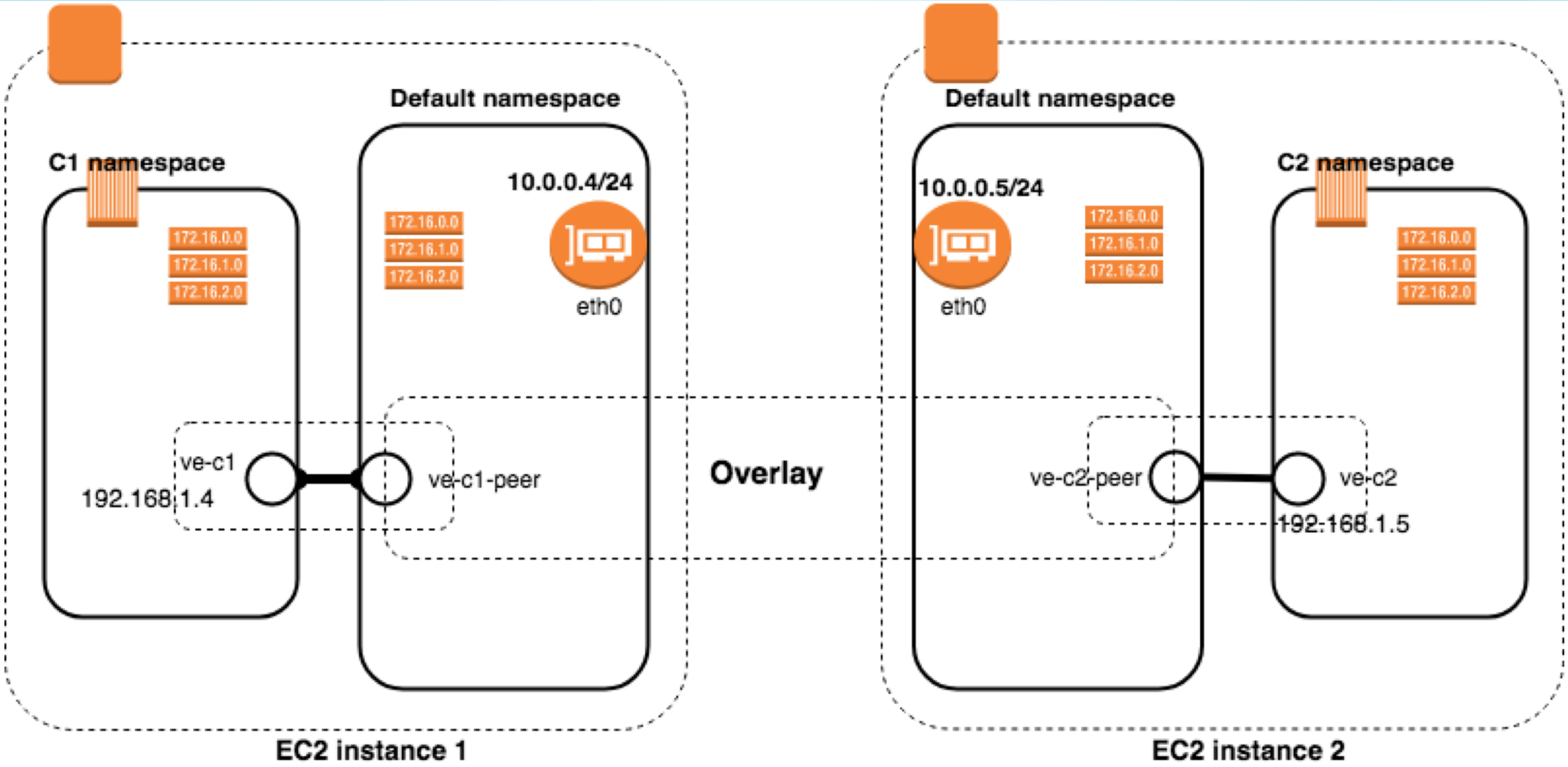
Two listeners : one host !



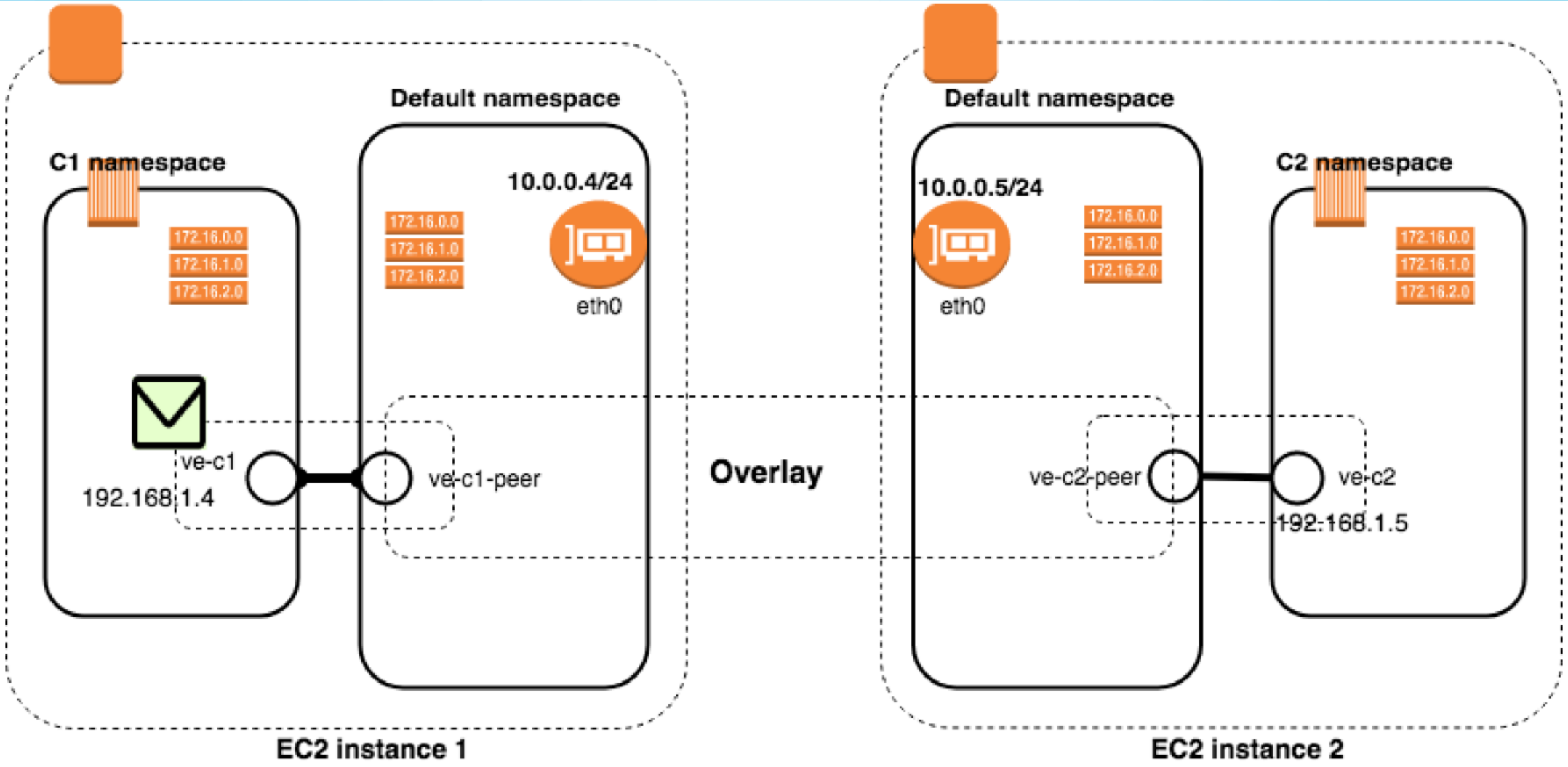
Two listeners : one host !



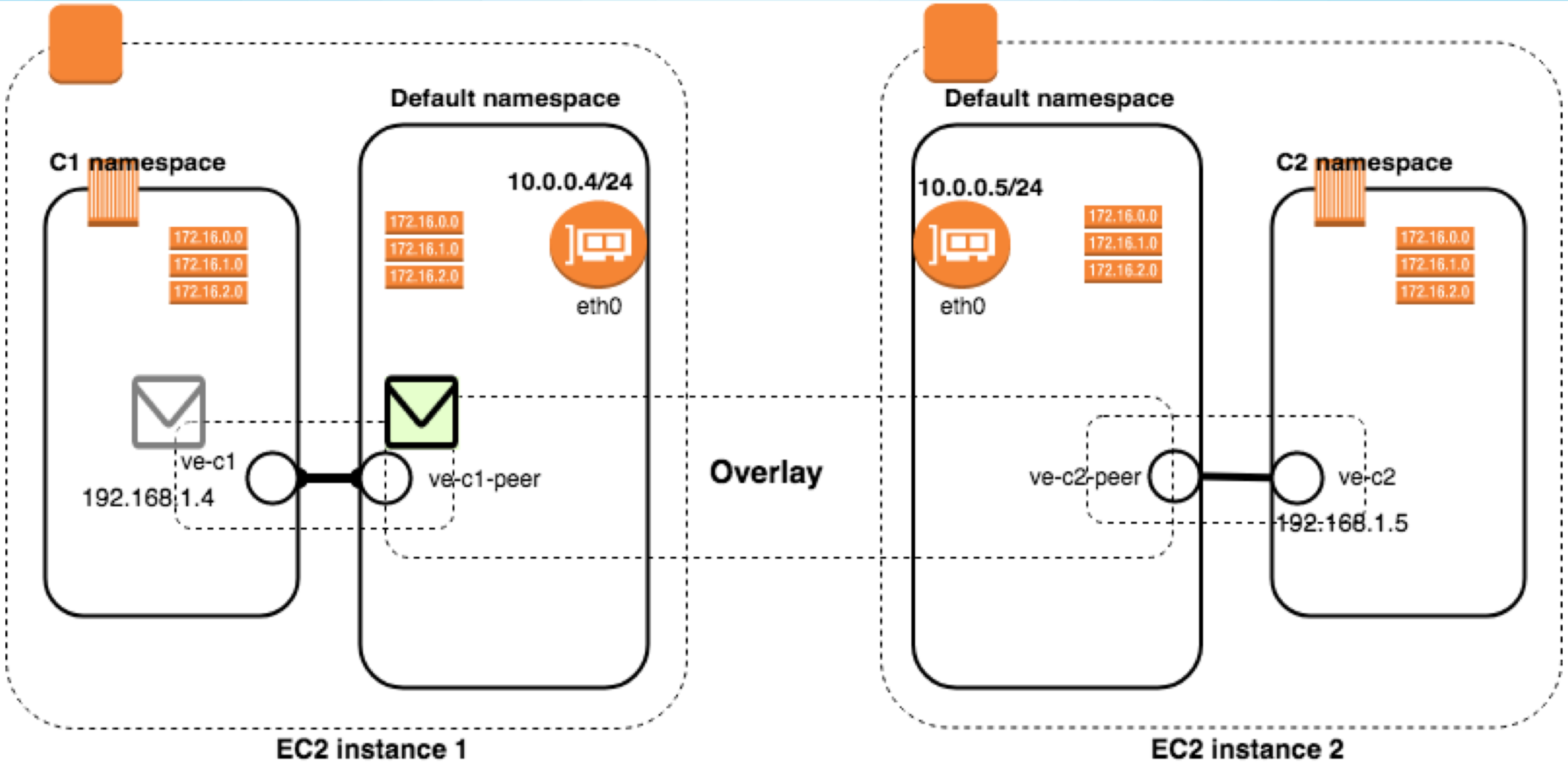
Connecting containers across hosts



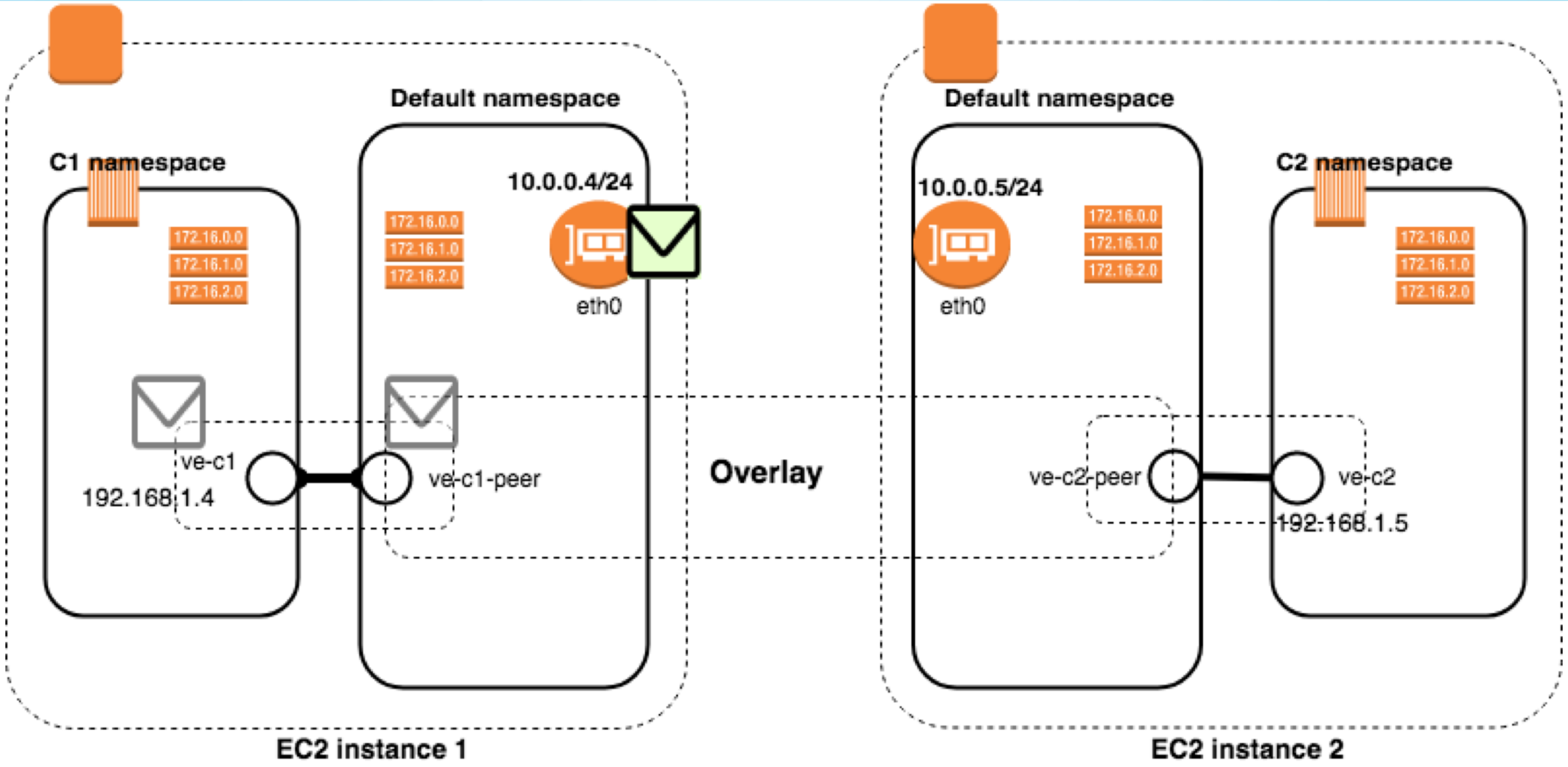
Container communication – 0



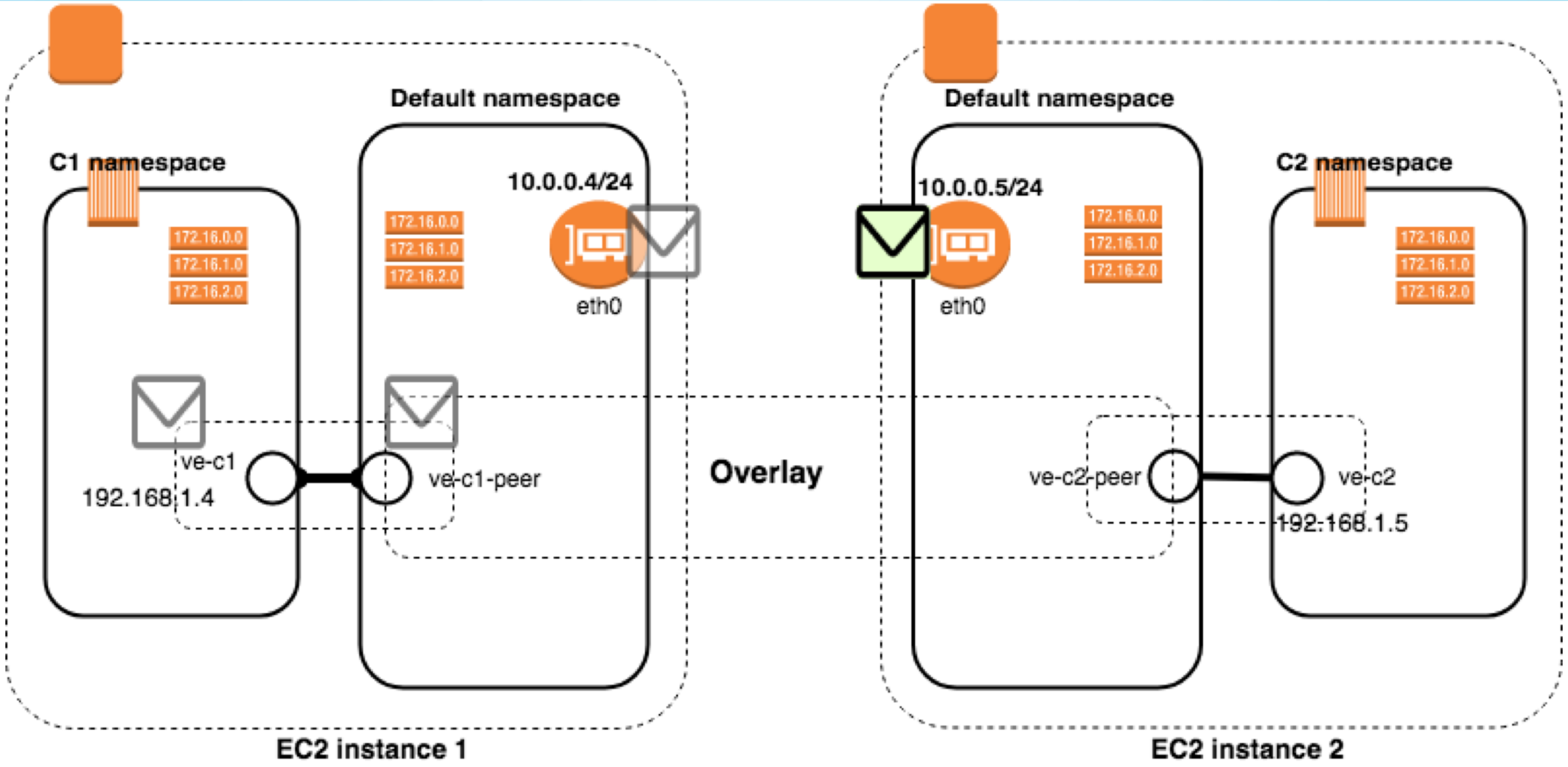
Container communication – 1



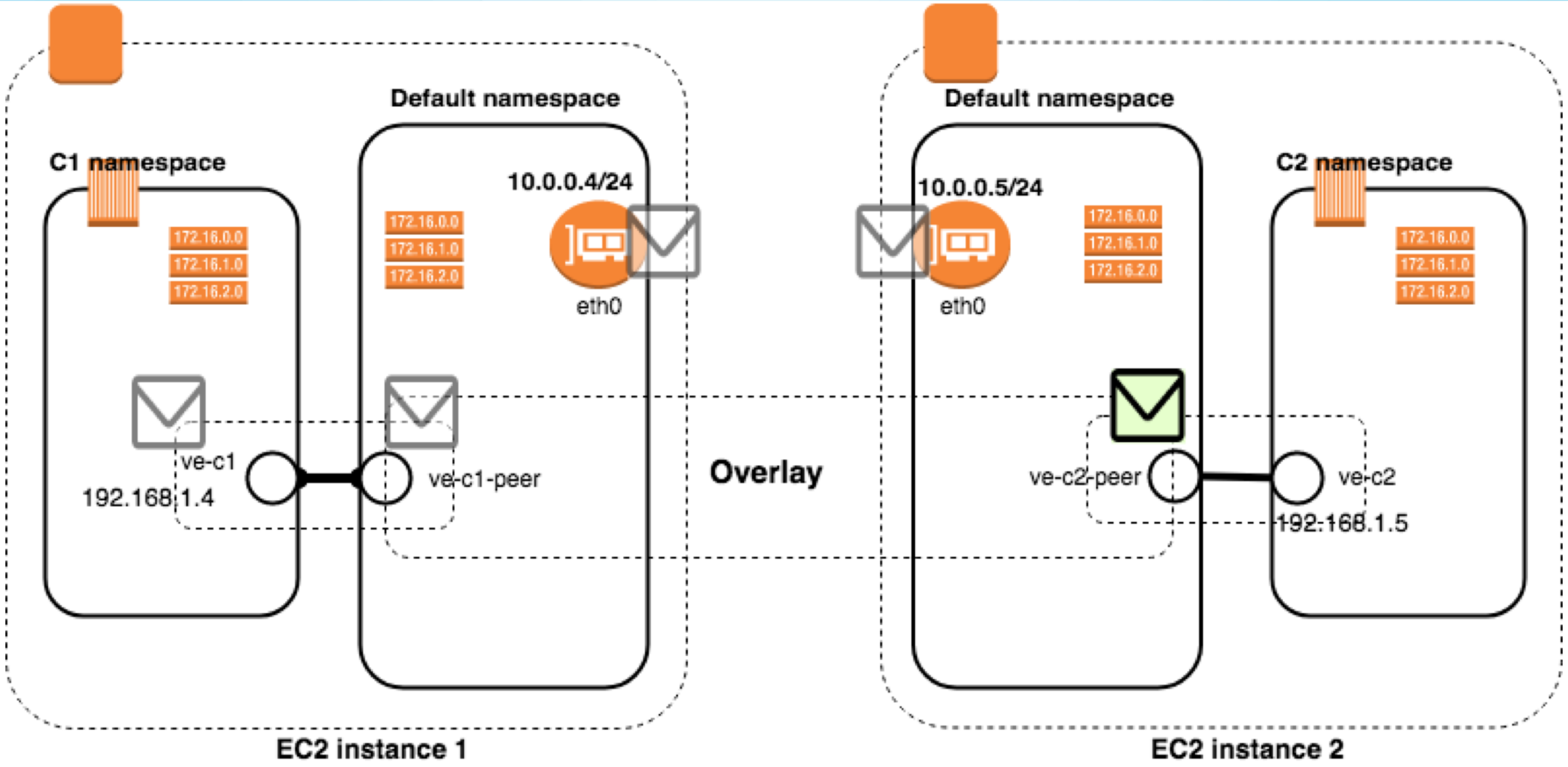
Container communication – 2



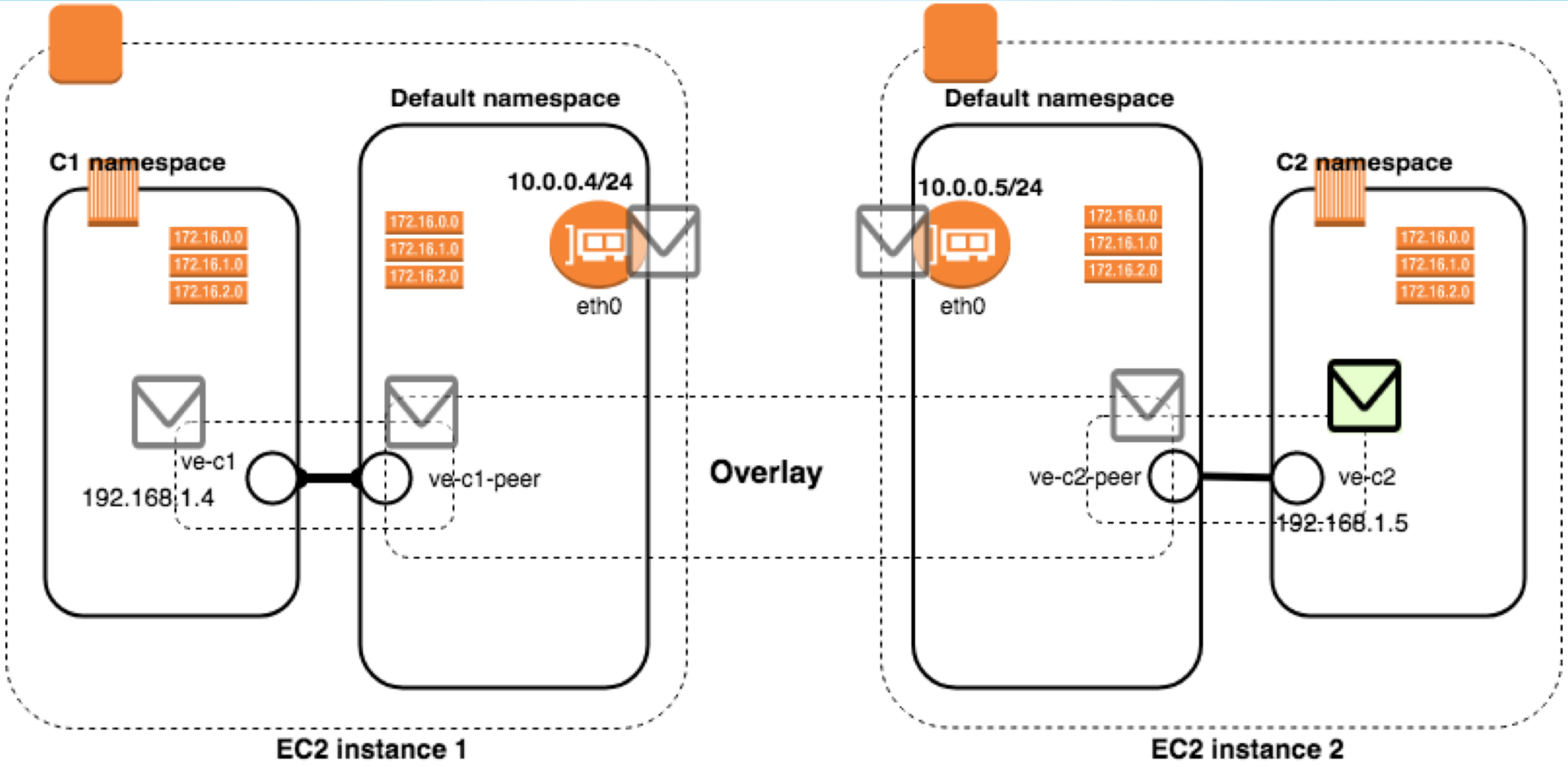
Container communication – 3



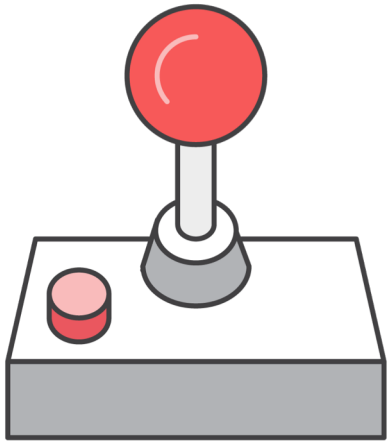
Container communication – 4



Container communication – 5



How did we do?

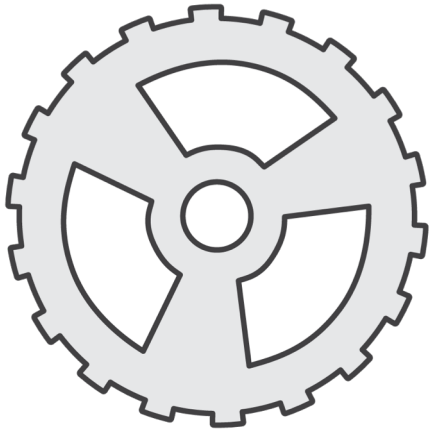


Usability

- Simple abstractions
- Discovery

Security

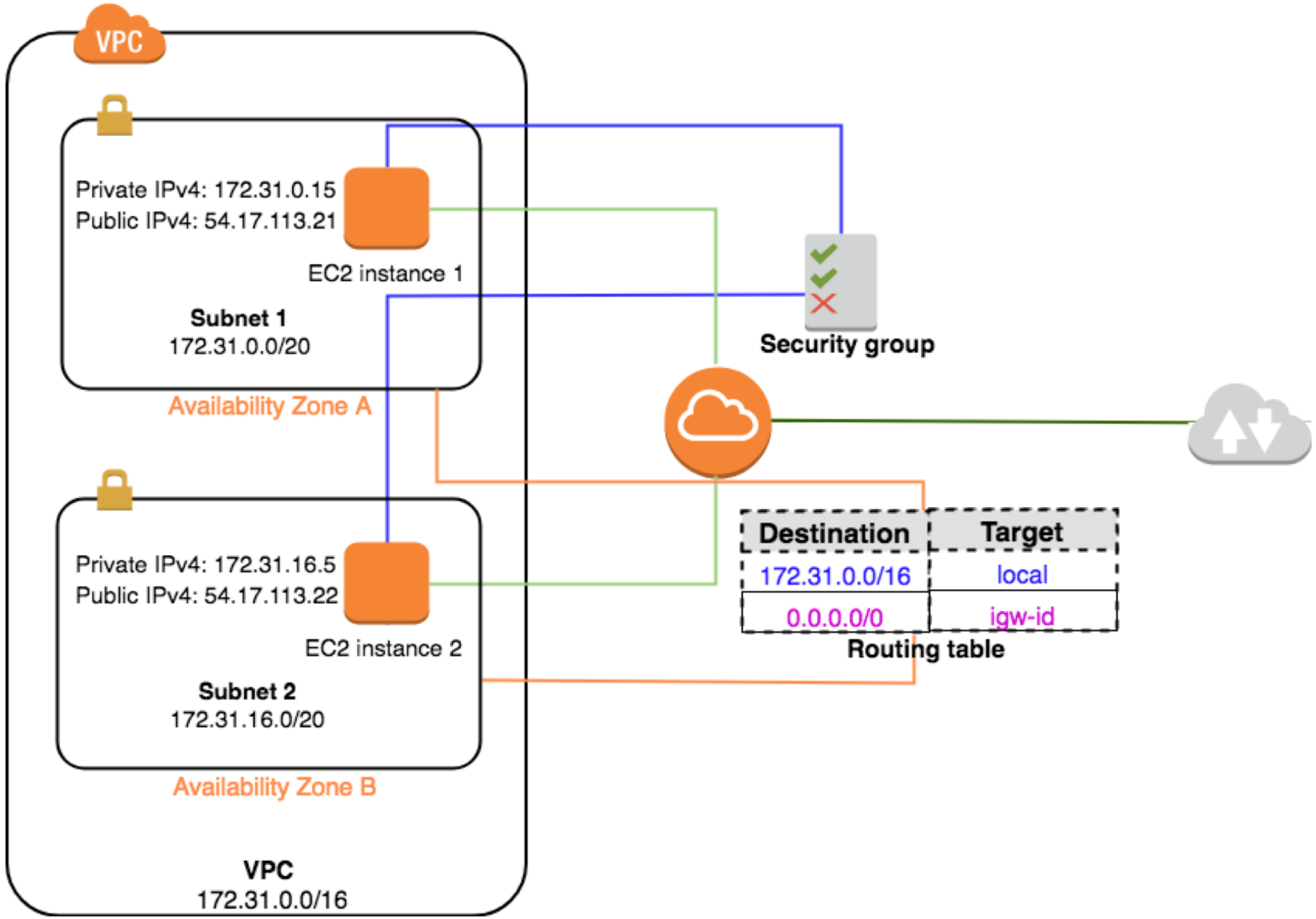
- ~~Network isolation~~
- ~~Access control~~
- ~~Auditability~~



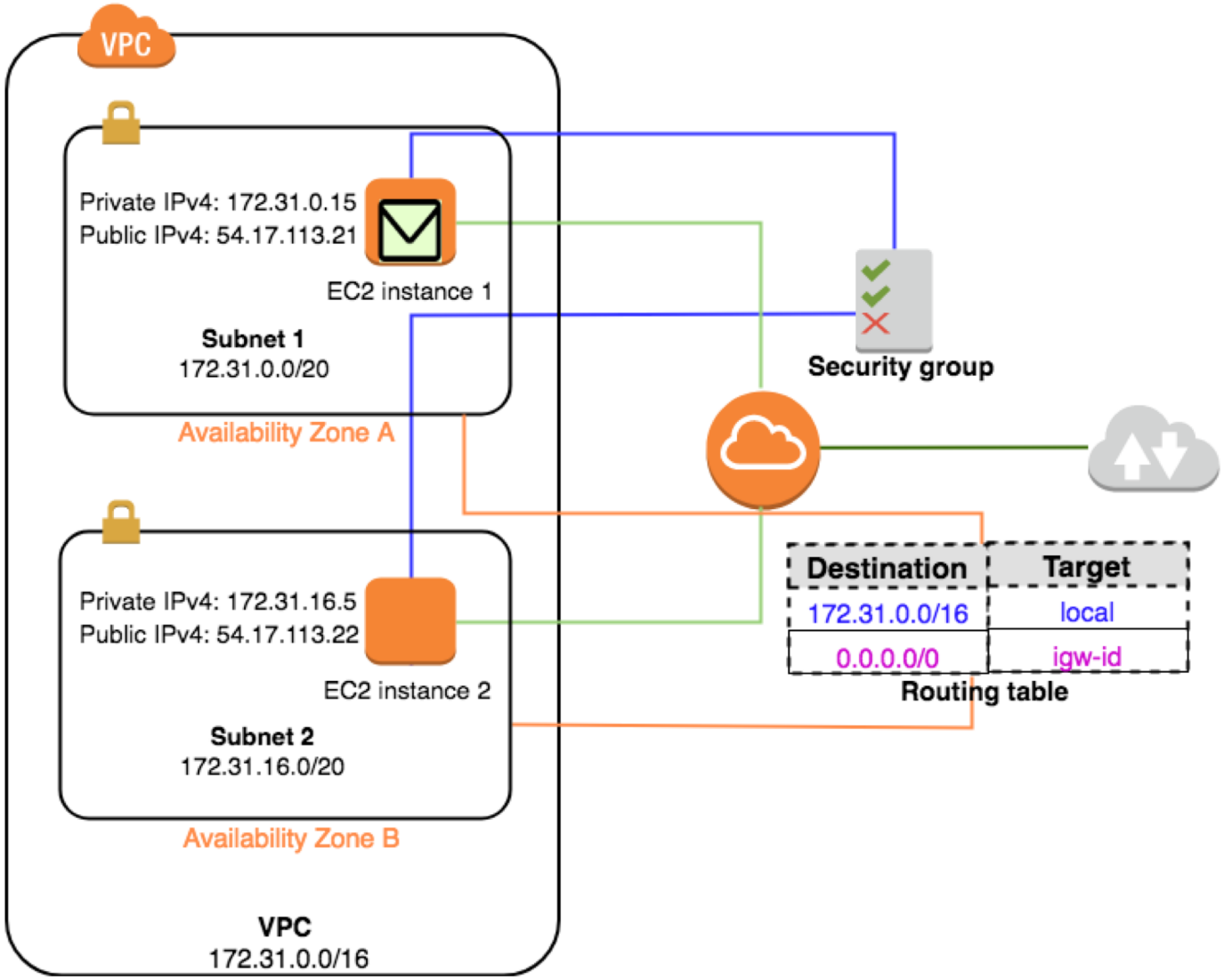
Maintainability

- ~~Scalability~~
- ~~Performance~~
- Monitoring

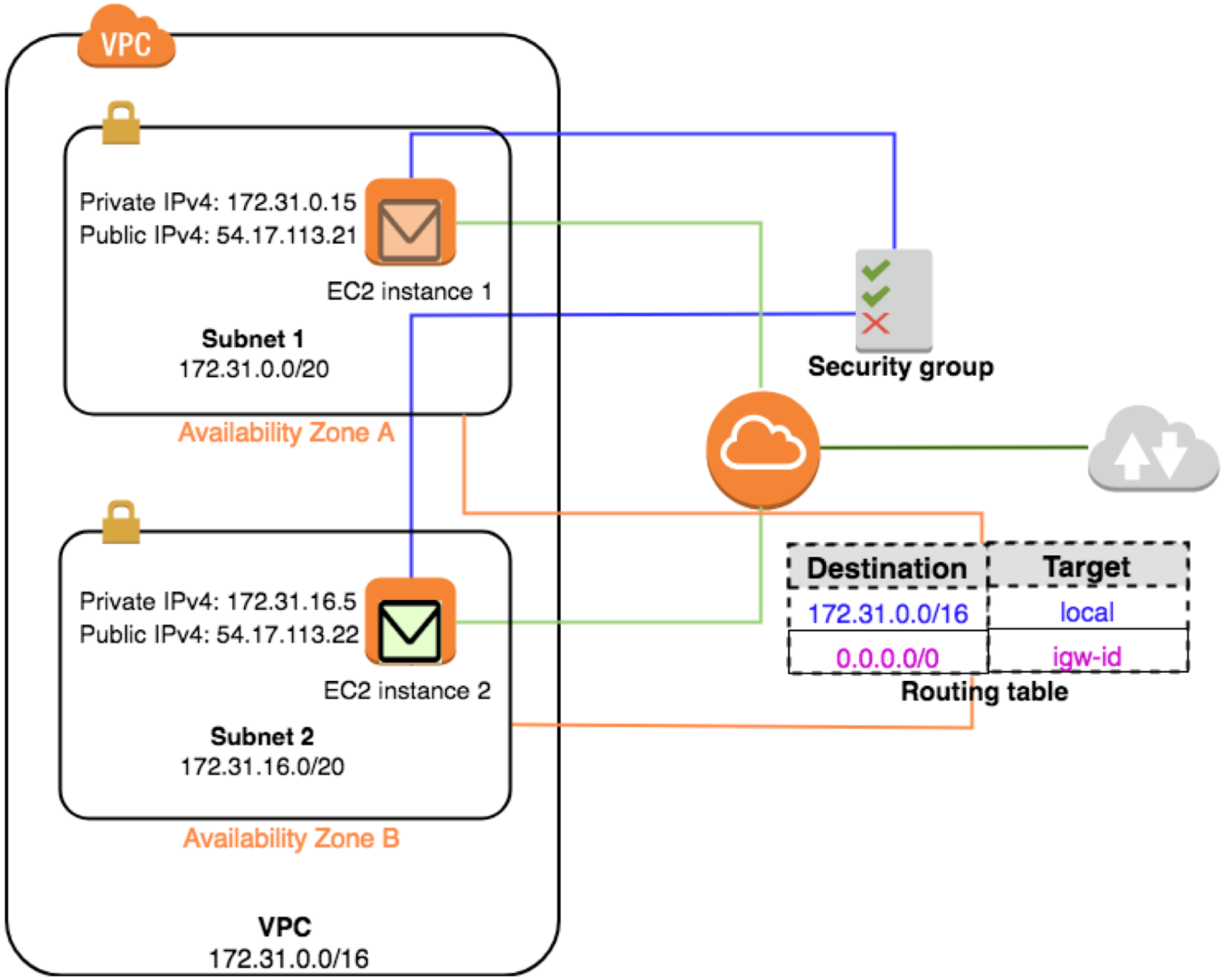
VPC networking – EC2 instances



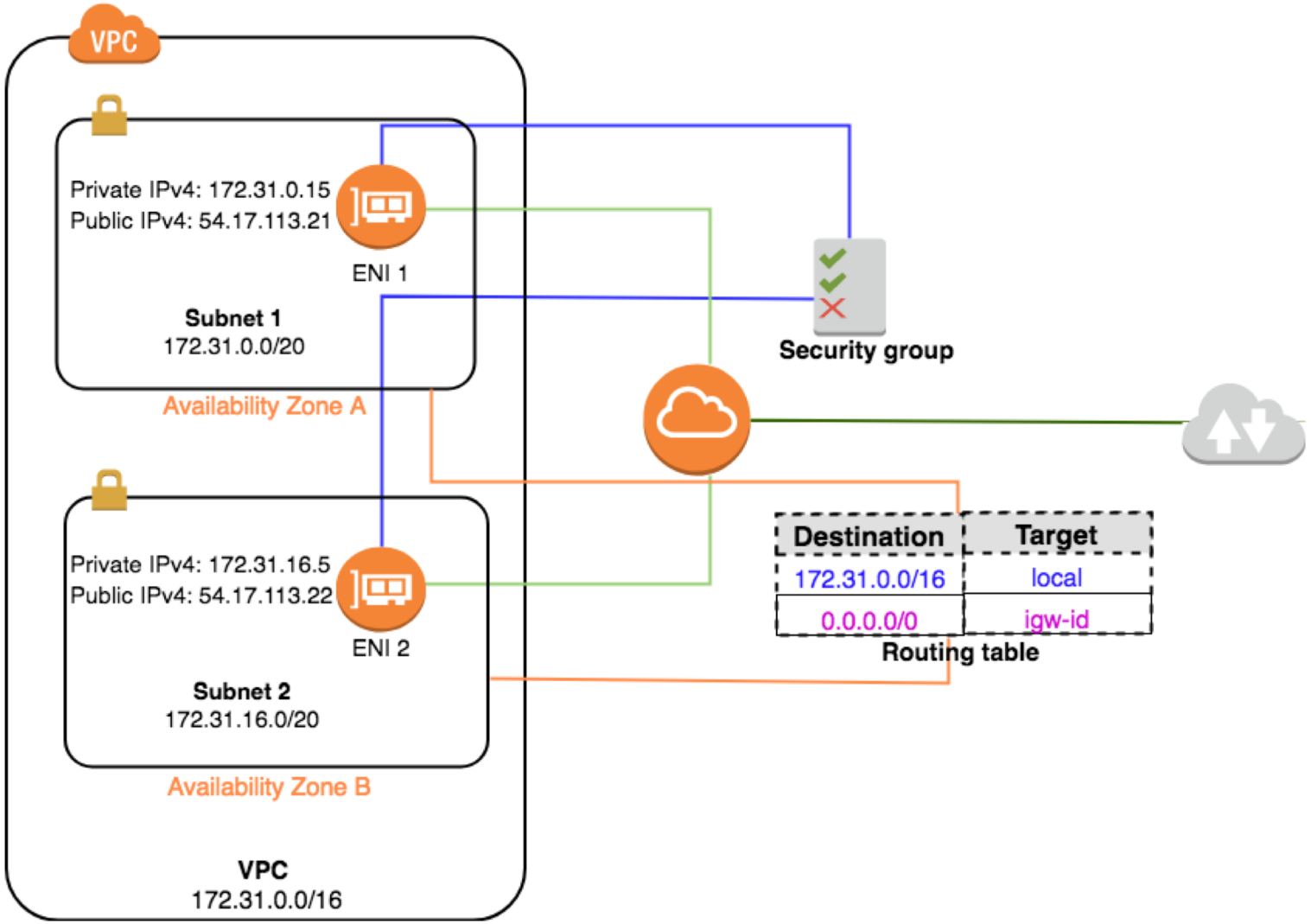
Packet flow – EC2 instances



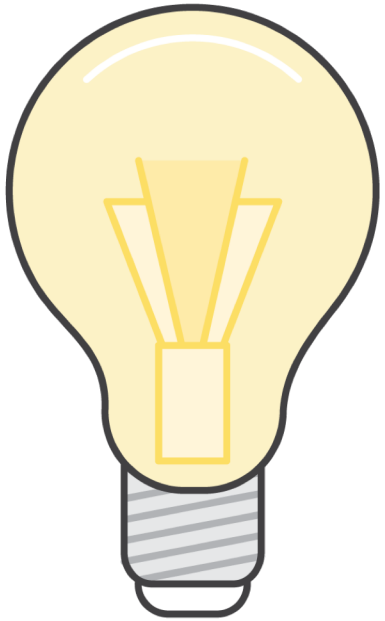
Packet flow – EC2 instances



VPC networking - ENIs



ENIs for tasks & pods



Amazon ECS Introduces Task Networking for Containers

Posted On: Nov 14, 2017

Tasks running on Amazon EC2 Container Service (Amazon ECS) can now take advantage of *awsvpc* mode for container networking. This new mode allocates an elastic networking interface to each running task, providing a dynamic private IP address and internal DNS name. This simplifies container networking operations, allowing tasks to run with full networking features on AWS, just like EC2 instances.

aws / **amazon-ecs-cni-plugins**

Code | Issues 9 | Pull requests 2 | Projects 0

Networking Plugins repository for ECS Task Networking

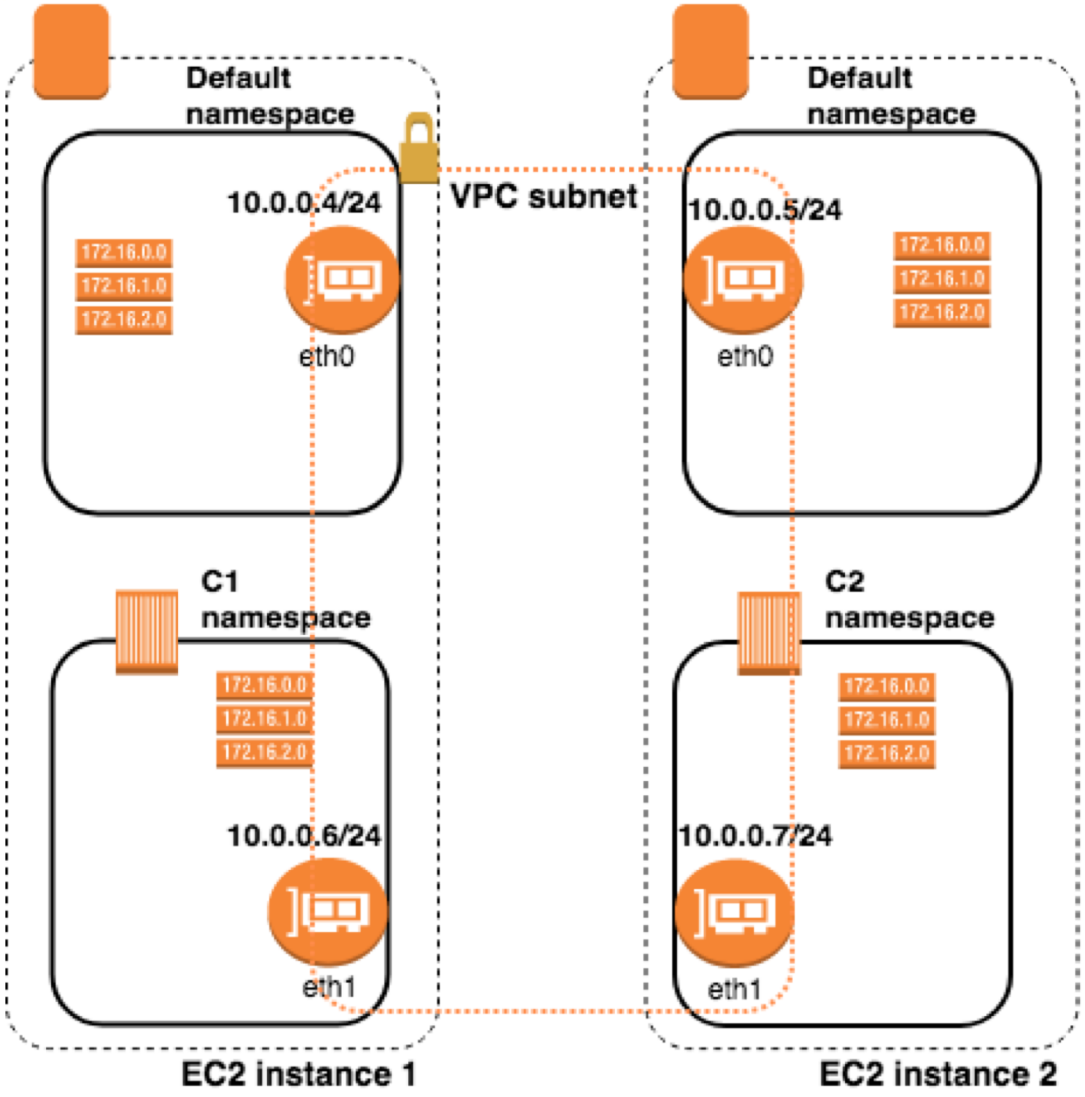
cni-plugin Manage topics

aws / **amazon-vpc-cni-k8s**

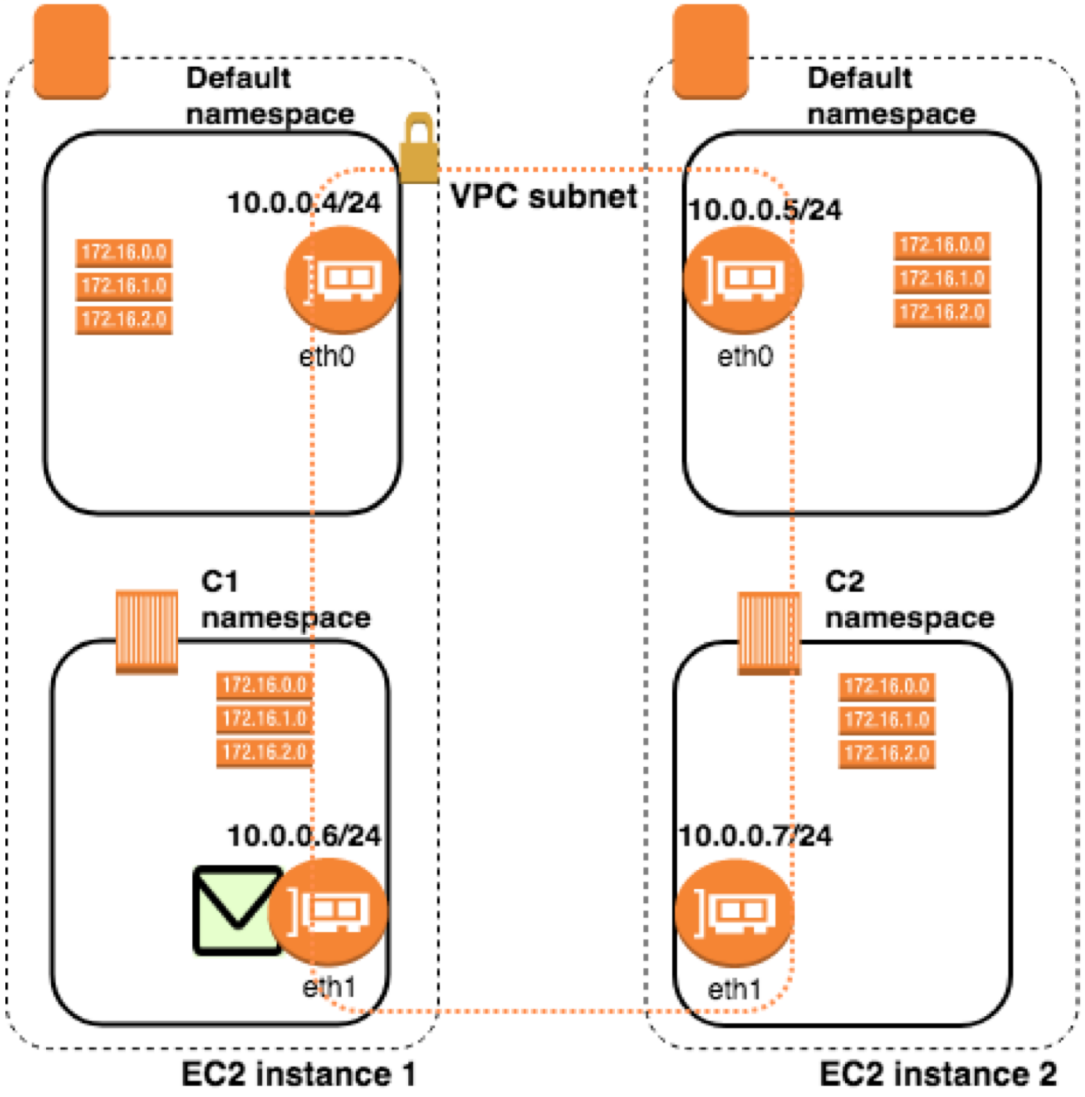
Code | Issues 25 | Pull requests 8

Networking plugin repository for pod networking i

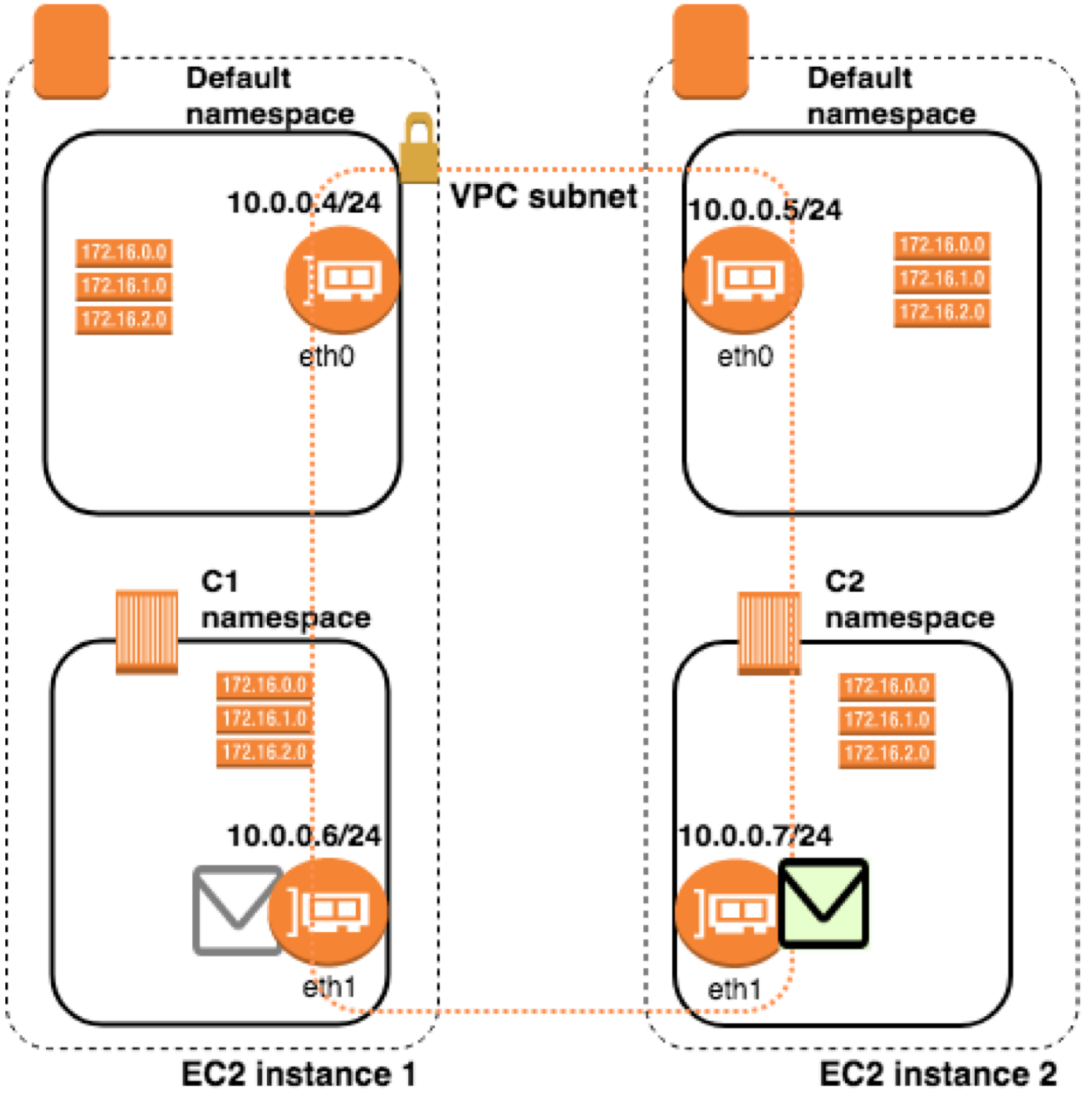
VPC networking – containers



Packet flow

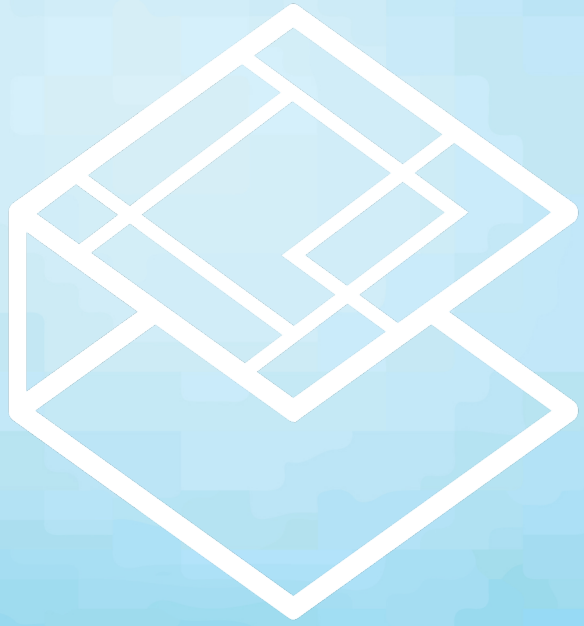

Default namespace
10.0.0.4/24
172.16.0.0
172.16.1.0
172.16.2.0
eth0
VPC subnet
Default namespace
10.0.0.5/24
172.16.0.0
172.16.1.0
172.16.2.0
eth0
C1 namespace
172.16.0.0
172.16.1.0
172.16.2.0
10.0.0.6/24
eth1
C2 namespace
172.16.0.0
172.16.1.0
172.16.2.0
10.0.0.7/24
eth1
EC2 instance 1
EC2 instance 2

Packet flow


Default namespace
10.0.0.4/24
172.16.0.0
172.16.1.0
172.16.2.0
eth0
VPC subnet
Default namespace
10.0.0.5/24
172.16.0.0
172.16.1.0
172.16.2.0
eth0
C1 namespace
172.16.0.0
172.16.1.0
172.16.2.0
10.0.0.6/24
eth1
C2 namespace
172.16.0.0
172.16.1.0
172.16.2.0
10.0.0.7/24
eth1
EC2 instance 1
EC2 instance 2

KubeCon | CloudNativeCon

Europe 2018

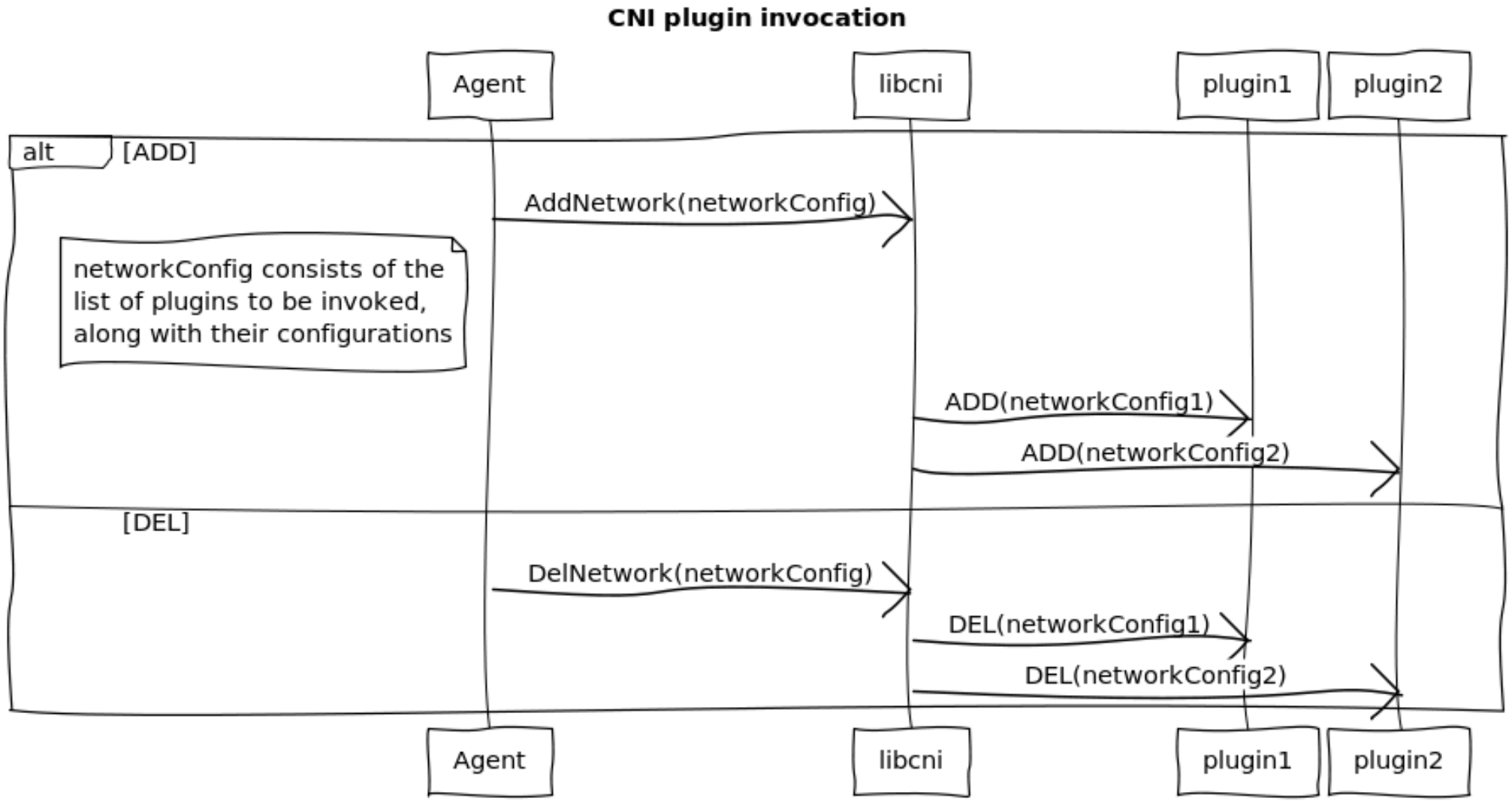


CNI

CNI or CNM ?

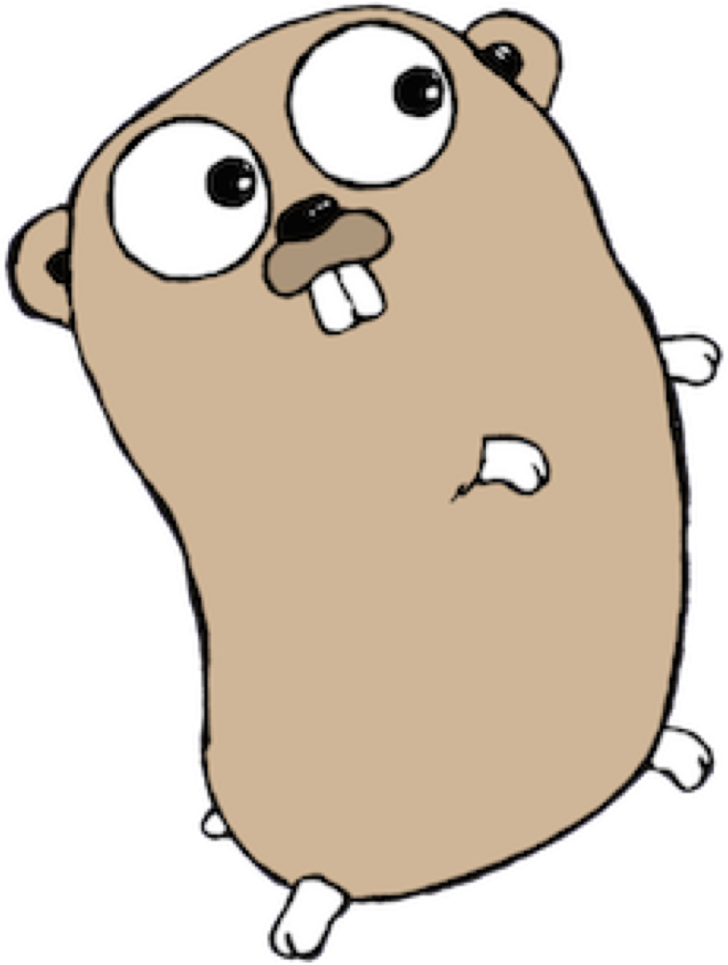
- ❒ Maintainability
  - ❒ Minimal intrusion to container life-cycle
  - ❒ Life-cycle management
  - ❒ Rolling out updates
- ❒ Simplicity
  - ❒ Consistent & reliable interface
  - ❒ Testability
- ❒ Extensibility
- ❒ Ecosystem support

CNI plugins 101



golang or …?

- Static binary
- Ecosystem support



Packaging & distribution

KubeCon | CloudNativeCon

Europe 2018

Branch: master ▾ amazon-ecs-agent / scripts / dockerfiles / **Dockerfile.release**

vsiddharth CNI Plugins Packaging: Address review comments

4 contributors

32 lines (25 sloc) | 1.13 KB

```
# Copyright 2014-2017 Amazon.com, Inc. or its affiliates. All Rights Reserved.
#
# Licensed under the Apache License, Version 2.0 (the "License"). You may
# not use this file except in compliance with the License. A copy of the
# License is located at
#
#       http://aws.amazon.com/apache2.0/
#
# or in the "license" file accompanying this file. This file is distributed
# on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either
# express or implied. See the License for the specific language governing
# permissions and limitations under the License.

# Not from scratch because we also want a little directory structure,
# specifically /tmp
FROM amazon/amazon-ecs-scratch:make

COPY out/amazon-ecs-agent /agent
COPY ["LICENSE", "NOTICE", "/"]

COPY out/amazon-ecs-pause.tar /images/amazon-ecs-pause.tar

# Copy our cni plugins ecs-eni, ecs-ipam and ecs-bridge
COPY out/cni-plugins /amazon-ecs-cni-plugins

# Copy our bundled certs to the first place go will check: see
# https://golang.org/src/pkg/crypto/x509/root_unix.go
COPY misc/certs/ca-certificates.crt /etc/ssl/certs/ca-certificates.crt

EXPOSE 51678 51679
ENTRYPOINT ["/agent"]
```

Versioning


```
{
  "version":"2017.06.0",
  "dirty":false,
  "gitShortHash":"226db3"
}
```

Automating version info


```
.PHONY: plugins
plugins: $(LOCAL_ENI_PLUGIN_BINARY) $(LOCAL_IPAM_PLUGIN_BINARY) $(LOCAL_BRIDGE_PLUGIN_BINARY)

$(LOCAL_ENI_PLUGIN_BINARY): $(SOURCES)
        GOOS=linux CGO_ENABLED=0 go build -installsuffix cgo -a -ldflags "\
             -X github.com/aws/amazon-ecs-cni-plugins/pkg/version.GitShortHash=$(GIT_SHORT_HASH) \
             -X github.com/aws/amazon-ecs-cni-plugins/pkg/version.GitPorcelain=$(GIT_PORCELAIN) \
             -X github.com/aws/amazon-ecs-cni-plugins/pkg/version.Version=$(VERSION) -s" \
             -o ${ROOT}/${LOCAL_ENI_PLUGIN_BINARY} github.com/aws/amazon-ecs-cni-plugins/plugins/eni
        @echo "Built eni plugin"
```

Testing the plugin


```
.PHONY: unit-test integration-test e2e-test
unit-test: $(SOURCES)
        go test -v -cover -race -timeout 10s ./pkg/... ./plugins/...

integration-test: $(SOURCE)
        go test -v -tags integration -race -timeout 10s ./pkg/... ./plugins/...

e2e-test:  $(SOURCE) plugins
        sudo -E CNI_PATH=${ROOT}/bin/plugins ${GO_EXECUTABLE} test -v -tags e2e -race -timeout 120s ./plugins/...
```

In conclusion …

- Avoid/minimize feature envy, especially for networking
- CNI plugins ftw!
- Version everything (with git SHAs)

Related links

https://github.com/aws/amazon-ecs-cni-plugins

https://github.com/aws/amazon-vpc-cni-k8s/

https://github.com/vishvananda/netns

https://github.com/vishvananda/netlink

https://github.com/containernetworking/cni

https://github.com/containernetworking/plugins

KubeCon | CloudNativeCon

Europe 2018

Thank you!

https://github.com/aws/amazon-ecs-cni-plugins/

https://github.com/aws/amazon-vpc-cni-k8s/

@aaithal