Why container networking?
What I’ve got

What I want
Goals

• For Users
  – Direct L3 addressability (no NAT) between connected peers
  – Single network interface per container
  – Policy based connectivity between applications

• For Operators
  – Choice of network technology
  – Policy enforcement
  – Application identity on the packet
How do we get there?

• Networking technology & Extensibility
• Address management
• Policy expression for Cloud Foundry
• Policy enforcement
• Discovery
Garden Runtime

Garden-RunC CNI adapter

Container Network Interface (CNI)

veth  macvlan  ipvlan  vxlan  anything…
Garden Runtime

Garden-RunC CNI adapter

Container Network Interface (CNI)

netman-release

flannel
Address Management

• Goal: Direct addressability to every container
  – Every container is assigned its own IP
  – Single network interface in each container
Policy: Roles & UX

• Developer
  – pushes apps
  – wants connectivity, ease of use

• Operator
  – administers CF & network
  – may have compliance requirements
Policy UX

- Implicit
  - Intra-org
  - Intra-space
  - Intra-app
- Explicit
  - Org to org
  - Space to org
  - Space to space
  - App-to-app
  - Process to process
  - +Port +Protocol

Coarse grained

Fine grained
Policy UX: What should CF support?

- Coarse grained:
  - Org to org
  - Space to org
  - Space to space

- Fine grained:
  - App-to-app
  - Process to process
  + Port + Protocol

- Implicit:
  - Intra-org
  - Intra-space
  - Intra-app

- Explicit:
Policy Expression

• How is it declared?
  – Cloud Controller API?
  – Separate Networking API?
  – Service bindings?
Explicit rules (option #1)

```
cf net allow src_org:src_space:src_app \
dst_org:dst_space:dst_app:protocol:port

cf net allow web_ui inventory:tcp:8080

cf net allow web_ui checkout:tcp:443

cf net allow checkout pci-space:credit-card:tcp:443

• “net-allow” requires appropriate roles / scopes
  – Space Developer on both apps?
  – Network admin scope in oauth token?
Explicit rules (option #2)

```bash
cf push inventory --expose inventory-ssl:tcp:443

cf push web_ui --bind inventory-ssl
```

- Internal endpoints exposed per app
- Intent to use endpoint is explicitly declared as “bind”
- Bindings could appear in environment, like for service bindings
- “bind" requires appropriate roles / scopes
  - Space Developer on both apps?
  - Network admin scope in oauth token?
CC API v3: process types?

• Explicit, directional access grants?
  
  • \texttt{process\_type1} $\rightarrow$ \texttt{process\_type2:protocol:port}

• Should processes in the same app have implicit access?
Implicit vs Explicit?

• With default deny
  • you write lots of rules
  • bi-directional flows are especially clumsy

```bash
cf push iftt --expose iftt:tcp:8080
cf push frontend --bind iftt
  --expose callback:tcp:8080
  --no-start
cf net allow iftt callback
cf start frontend
```

• With default allow, do you have compliance problems?
• How should we balance usability vs security?
Policy Enforcement

• Goal: Batteries included, but replaceable
• Open-source batteries: VXLAN overlay network
  – Encapsulate application traffic
  – Add “tag” to represent app (or v3 process) identity
  – Decouples addressing from identity
Policy Enforcement via tagged packets

- Each CF app is assigned a unique N-bit tag
- Traffic from a container is encapsulated & tagged
- Traffic to a container is filtered based on source tag
Batteries Included: VXLAN

- Tag where?
  - 24-bit VXLAN VNI (~16M max)
  - 16-bit VXLAN Group Policy ID (~65K max)
  - Both, concatenated? (~1T max)
  - Both, VNI is “tenant”, GPID is app ID?
Policy Enforcement

Policy "Brain"

Remote Policy API

Diego Cell

Garden

Netman

Local Policy API

CNI API

Policy plugin

CNI plugin
Extensibility

- Remote Policy API
  - Diego Cell
    - Garden
    - Netman
  - Local Policy API
    - Policy plugin
  - CNI API
    - CNI Plugin

- Policy “Brain”
- 3rd-party policy management
Discovery

• Now that you can reach other applications, how do you find them?

• Proposals
  – Bring your own mechanism (e.g. Consul, Eureka)
  – CF-provided DNS
    • application guid as the host name
    • “internal routes”: `cf map-internal-route app my-name`
    • name associated with the policy “binding”

• We’ve explored CF-provided DNS, not on current roadmap
Get involved!

- Ask questions
- Share your requirements!
- What are we missing?

#container-networking @ cloudfoundry.slack.com
More container networking at CF Summit…

Wednesday, May 25 • 2:10pm - 2:40pm
Project Office Hours: Container to Container Networking

Wednesday, May 25 • 2:50pm - 3:20pm
Discussion on the Container Network Interface (CNI) and the OCI projects
Craig Mcluckie, Google, Chris Aniszczyk, Cloud Native Computing Foundation, Richard Kaufmann, Samsung, and Ken Owens, Cisco

cfsummit2016.sched.org
Questions?