Containerizing the ThousandEyes Enterprise Agent for Platform Integration

Tyler Duzan, Product Manager for Agent Strategy

Cisco Webex App

Questions?
Use Cisco Webex App to chat
with the speaker after the session

How

① Find this session in the Cisco Live Mobile App

② Click "Join the Discussion"

③ Install the Webex App or go directly to the Webex space

④ Enter messages/questions in the Webex space

Webex spaces will be moderated until February 24, 2023.

cisco *Live!*
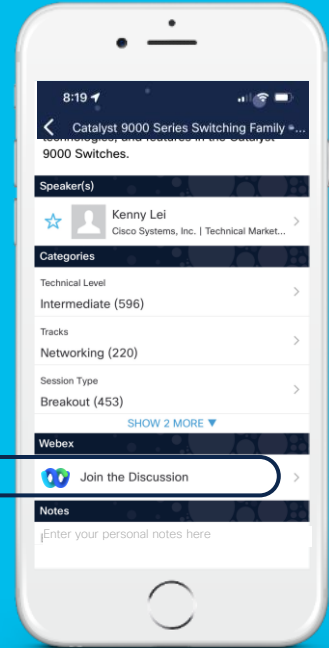
Agenda

- ThousandEyes Overview

- Cisco Application Framework Overview

- How We Containerize Now

- What's Coming: Building on Alpine

- What's Coming: ARM, Nexus, IoT

We're in a new world

Your Apps
Your People
Your Infrastructure

| All distributed

Visibility and control is shrinking

External dependencies are exponentially growing

Data Center → Cloud
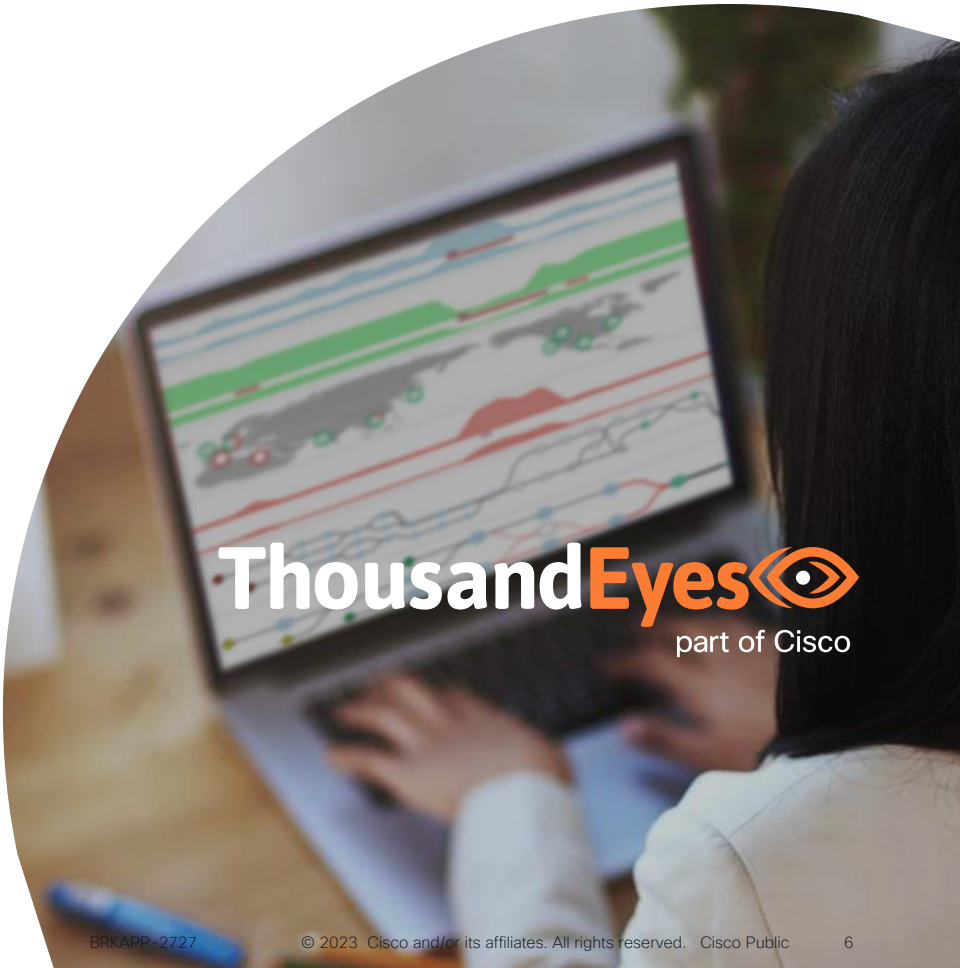
App Stack → SaaS + API

Office → Home

Network → Internet

See, understand, and improve digital experiences everywhere

What is an Enterprise Agent?

- Software agents actively monitoring the network

- Designed to run on many different platforms with minimal requirements

- Deployed within your enterprise network

Hosting / SaaS Provider

Internet

ERP, CRM or Email

Data Center

Enterprise Agent

Enterprise Agent

Branch
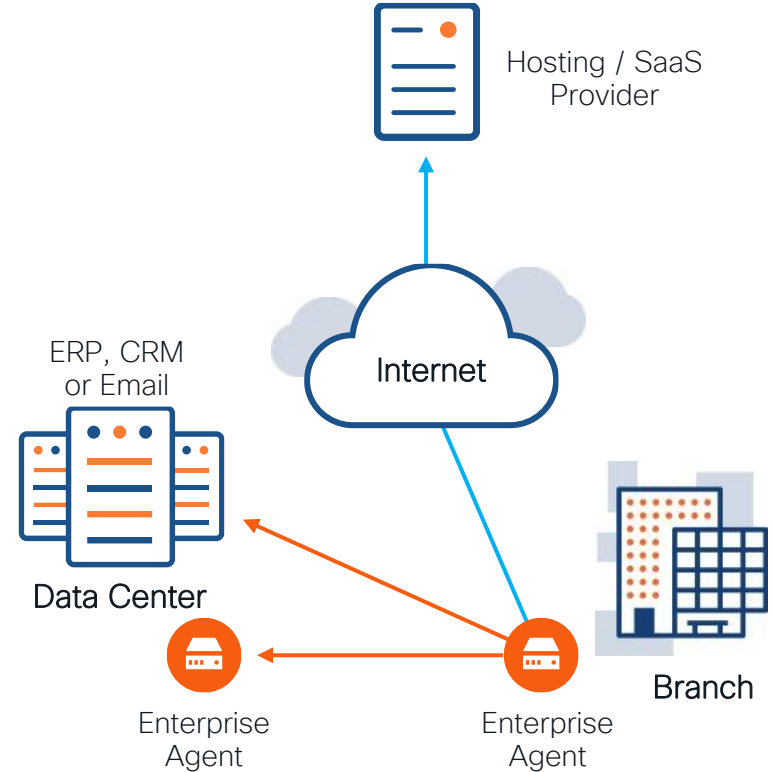
ThousandEyes Virtual Appliance (TEVA)

- VM that's the most often deployed for the Enterprise Agent

- Intentionally limited control as part of its managed experience

- On-prem integration on top of VMware, Hyper-V, and KVM

- Downsides include heavy weight and requires on-prem compute capability

What Customers Really Needed

We learned that customers wanted to see into their retail and branch locations. Especially with the push to SD-WAN.

To address this, we took two steps:
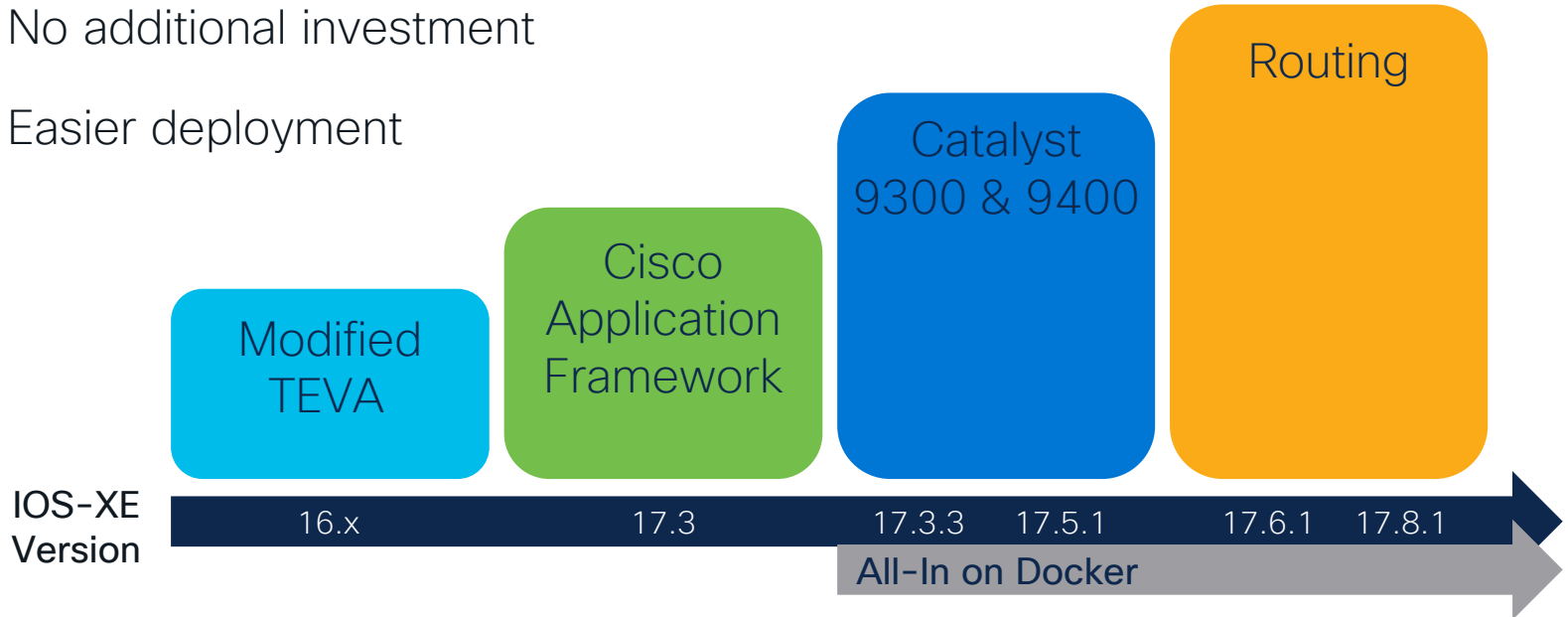
ThousandEyes Physical Appliances

Intel NUC | Raspberry Pi

Docker-Based Agent Containers

Taking it One Step Further

How can we integrate with hardware already on site?

✓ No additional investment

✓ Easier deployment

Routing

Catalyst 9300 & 9400

Cisco Application Framework

Modified TEVA

IOS-XE Version

16.x   17.3   17.3.3   17.5.1   17.6.1   17.8.1

All-In on Docker

Cisco Application Framework

What is Cisco Application Framework?

An abstraction layer on top of Cisco hardware that includes a management capability

Originally created to hosting lightweight applications for IoT

- Expanded onto enterprise Catalyst Switching and Routing

Flexible enough to allow platform-specific implementation details

- Robust enough to act as a development target for containerized applications
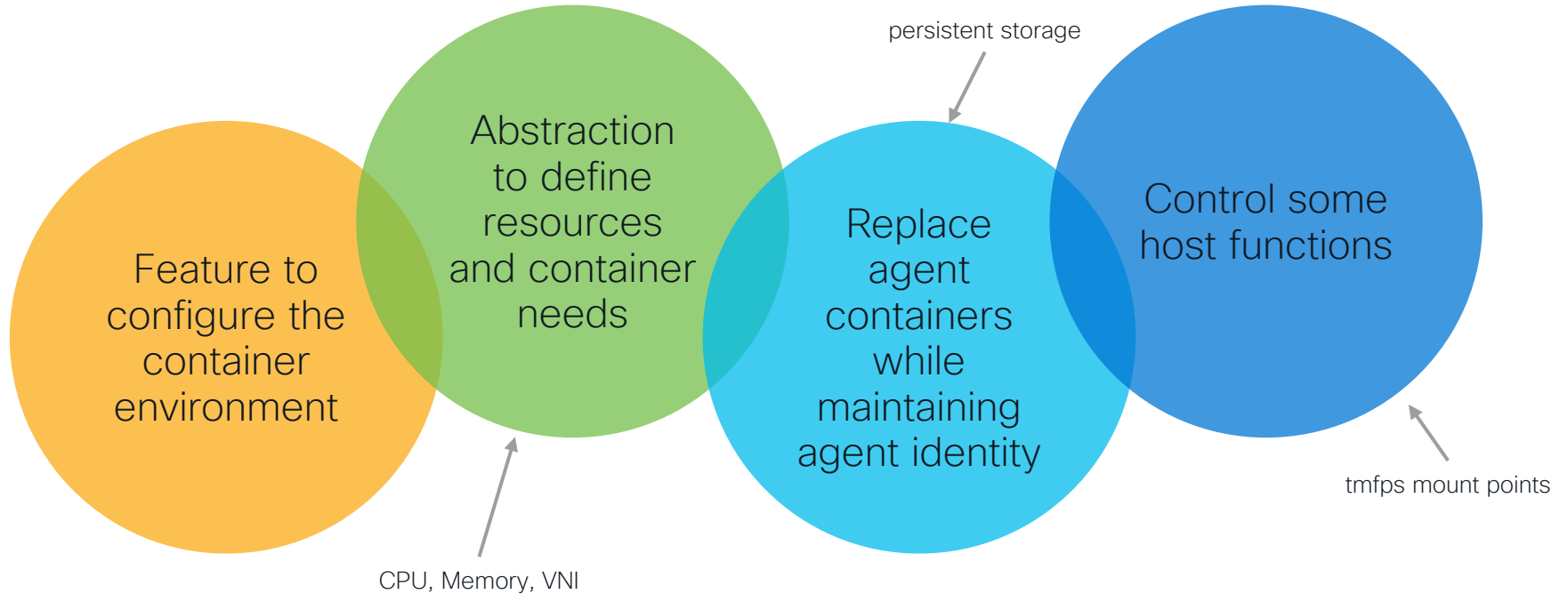
Why Use Cisco Application Framework?

A single container image works across multiple devices

Supporting new Cisco platforms is now a validation process, not a development one

Standardizes how agents are run across the Cisco hardware portfolio

Minimal documentation and support required

What Does CAF Provide?

Feature to configure the container environment

Abstraction to define resources and container needs

persistent storage

Replace agent containers while maintaining agent identity

Control some host functions

tmfps mount points

CPU, Memory, VNI

What Limitations Does CAF Have?

Some Incompatible Platforms

Containers Limited by OS kernel config
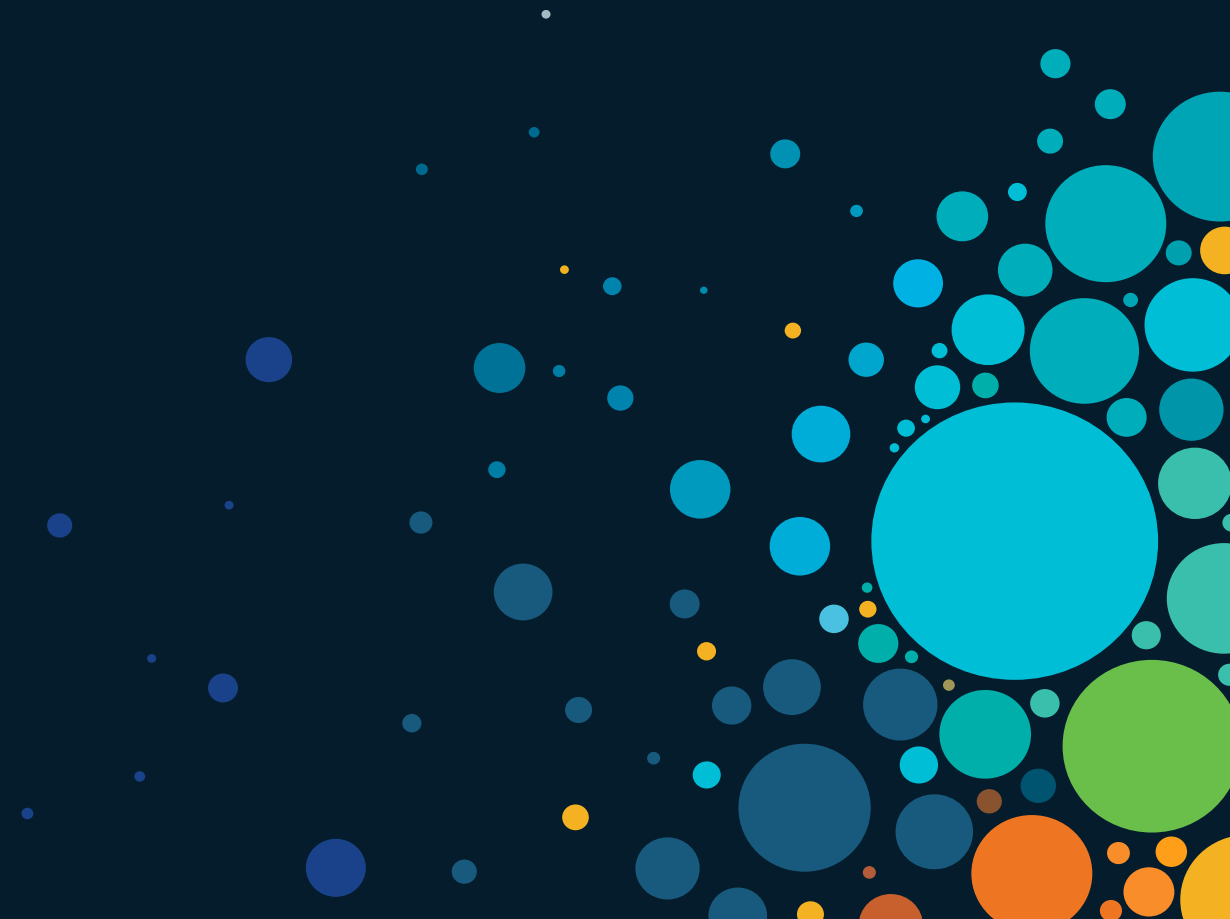
Rate-limiting of Packets Entering Container

Platforms With Differing Resource Constraints
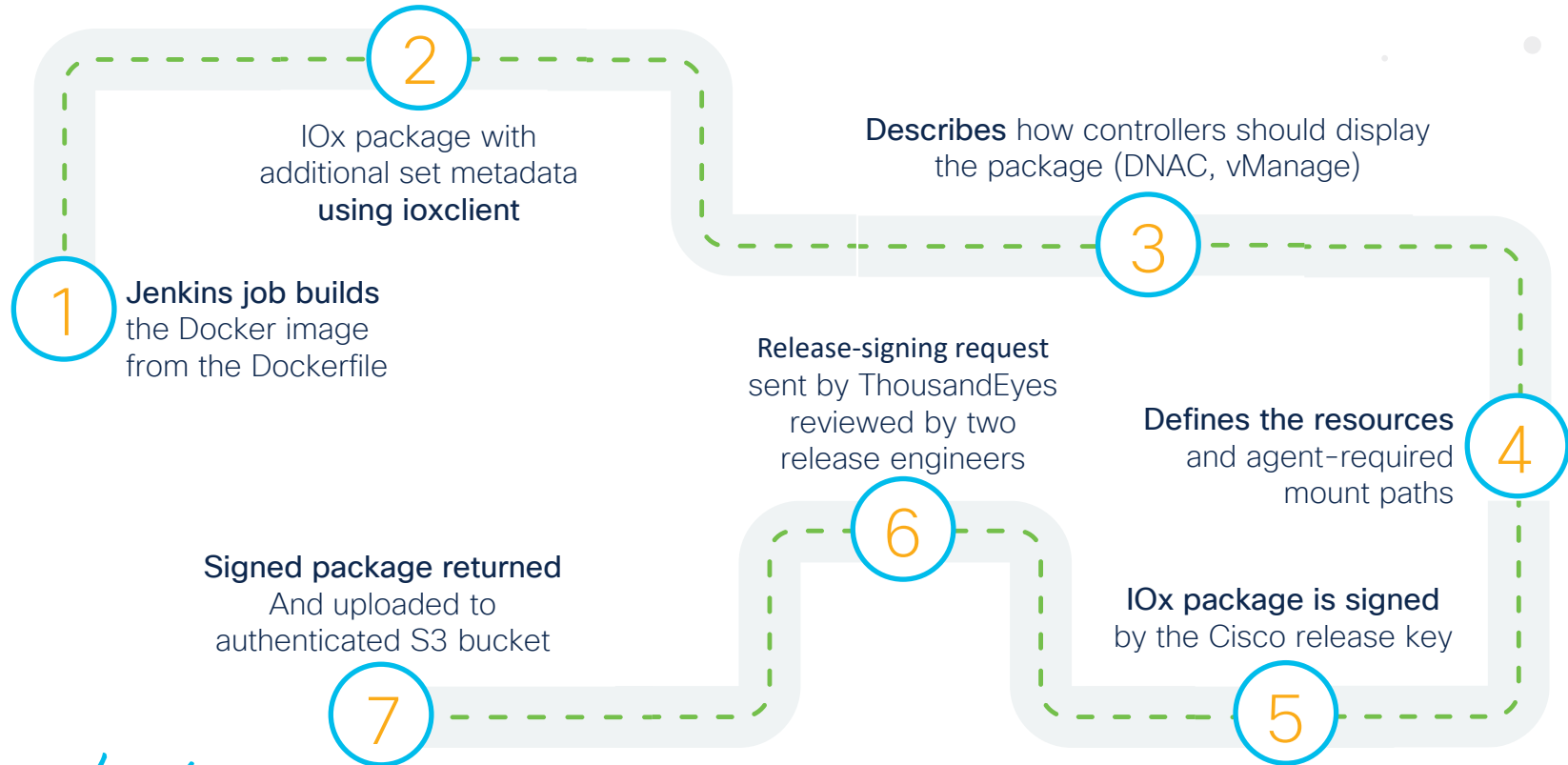
What Platforms Support CAF Today?

- Cisco Industrial IoT Switches and Routers in the IE and IR families

- Cisco Catalyst Switches, both Access and Core

- Cisco Catalyst Routing

- Cisco Routing ASR and ISR

- Cisco Nexus Datacenter Switches in NX-OS standalone mode
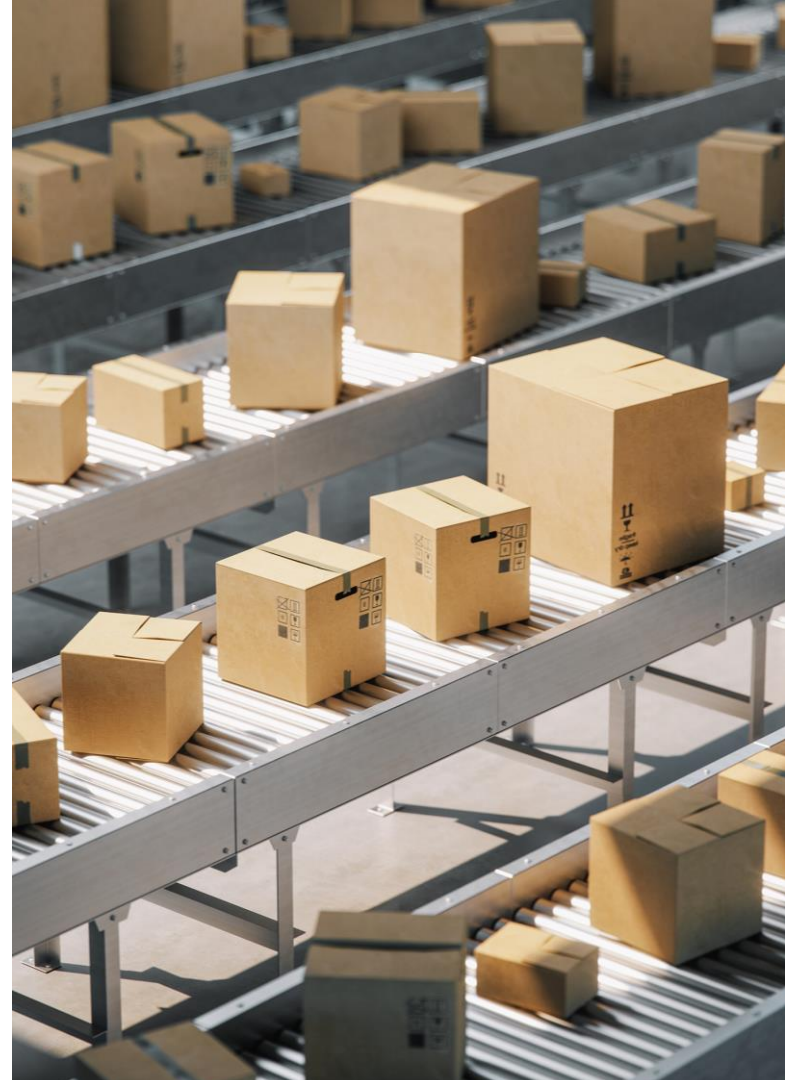
How We Containerize Now

The ThousandEyes Build Process

2

IOx package with
additional set metadata
using ioxclient

Describes how controllers should display
the package (DNAC, vManage)

3

1 **Jenkins job builds**
the Docker image
from the Dockerfile

Release-signing request
sent by ThousandEyes
reviewed by two
release engineers

Defines the resources
and agent-required
mount paths

4

6

Signed package returned
And uploaded to
authenticated S3 bucket

IOx package is signed
by the Cisco release key

7

5

Package-Centric Approach

- All components of the ThousandEyes Enterprise Agent are packaged as debs for Ubuntu Server LTS

- Containers are built by installing packages on top of the base, only.

- No non-package customization, other than the contents of the container init script

Simplistic Dockerfile

We use Phusion minimal Ubuntu base images

The Dockerfile process:

Pull in base image	→ Add our GPG keys for ThousandEyes package repository	→ Install packages from ThousandEyes and Ubuntu

Copy over the init script we run on container start	← Register our agent process with `runit`	← Copy over a pre-made log rotate configuration

Dealing With Storage Using tmpfs

Key Challenge

Requirements for log storage and results cache on Cisco platforms

Our Solution

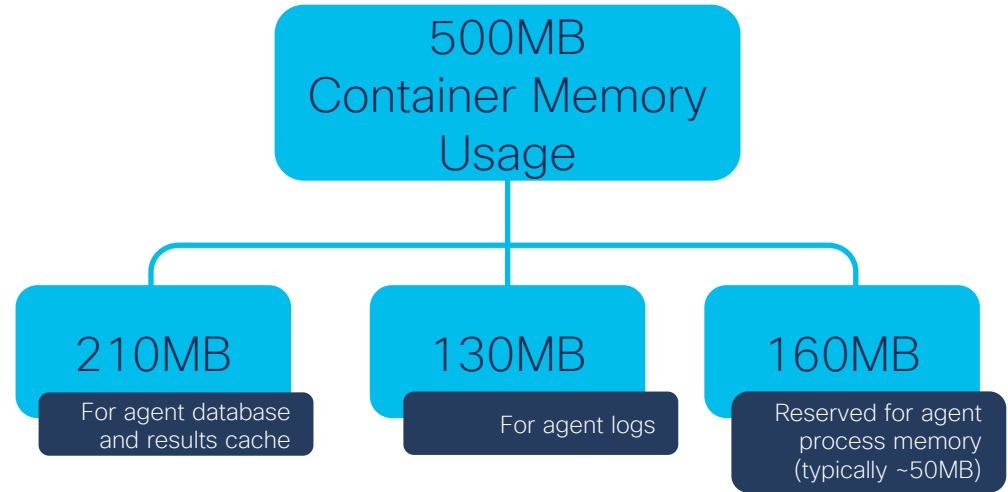We write results to database after each interval, then purge after its ingested

Things to Note
When connectivity is lost, agent caches several hours of test results and ingests them for backfilling when the connection is restored

The agent logs extensively, requiring the use of log rotation

Dealing With Storage Using tmpfs

- tmpfs ensures our high disk churn won't damage flash storage

- All storage in memory and accounted as part of container memory

- CAF sets cgroups rules from our container manifest and enforces memory limits

500MB
Container Memory
Usage

210MB

For agent database and results cache

130MB

For agent logs

160MB

Reserved for agent process memory (typically ~50MB)

Networking Capabilities Required

Enterprise Agent uses raw sockets to generate TCP probes
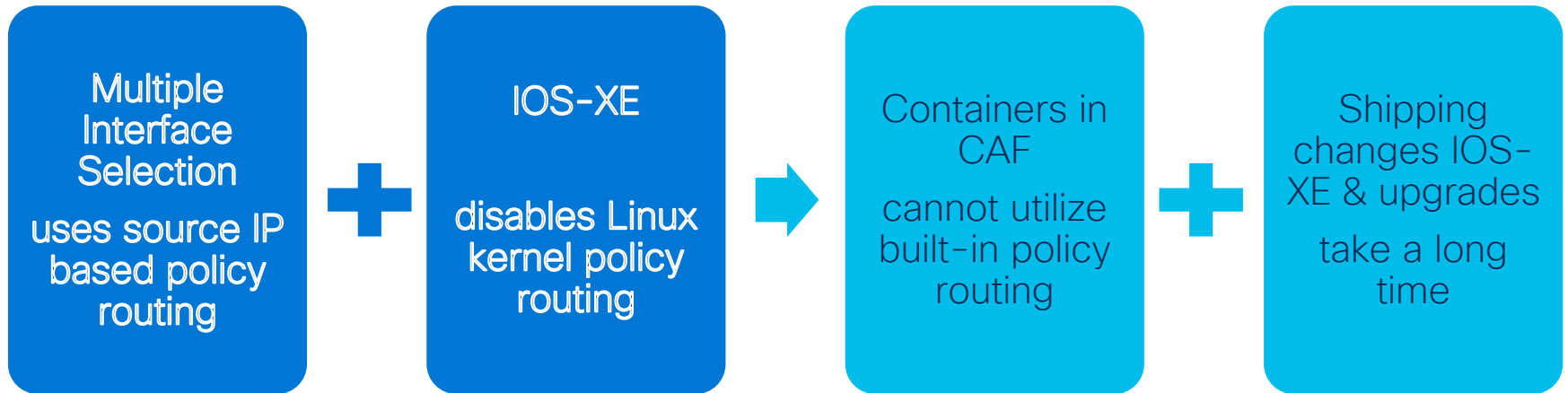
Network configuration inside container via init script

CAP_NET_ADMIN and CAP_NET_RAW is required to run our containers

Case Study: Multiple Interface Selection with CAF

Multiple Interface Selection

uses source IP based policy routing

+

IOS-XE

disables Linux kernel policy routing

→

Containers in CAF

cannot utilize built-in policy routing

+

Shipping changes IOS-XE & upgrades

take a long time

Case Study: Multiple Interface Selection with CAF

We identified an alternative pathway…

1. Specified a different gateway with a different metric for each interface

2. Set up networking, pairing interfaces to gateways

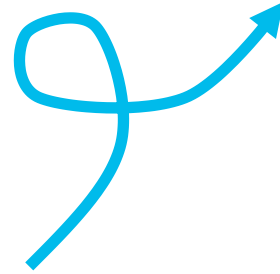3. Use SO_BINDTODEVICE to specify the interface socket it opened on

Case Study: Multiple Interface Selection with CAF

SO_BINDTODEVICE

Forces traffic associated to that socket out a particular interface

Responses on the same interface required

Traffic sent with correct source IP and gateway

Allows you to create multiple guest interfaces for container

Case Study: Multiple Interface Selection with CAF

```
app-hosting appid cat9k402
 app-vnic AppGigabitEthernet trunk
  vlan 21 guest-interface 0
   guest-ipaddress 10.100.21.65 netmask 255.255.255.0
  vlan 22 guest-interface 1
   guest-ipaddress 10.100.22.65 netmask 255.255.255.0
  vlan 23 guest-interface 2
   guest-ipaddress 10.100.23.65 netmask 255.255.255.0
  vlan 24 guest-interface 3
   guest-ipaddress 10.100.24.65 netmask 255.255.255.0
 app-default-gateway 10.100.21.1 guest-interface 0
 app-resource docker
  prepend-pkg-opts
  run-opts 1 "-e TEAGENT_ACCOUNT_TOKEN=
  run-opts 2 "--hostname cat9k-multi"
  run-opts 4 "-e TEAGENT_DEF_IPV4_GW_ETH1=10.100.22.10"
  run-opts 5 "-e TEAGENT_DEF_IPV4_GW_ETH2=10.100.23.10"
  run-opts 6 "-e TEAGENT_DEF_IPV4_GW_ETH3=10.100.24.10"
```

Default Interface is configured by 'app-default-gateway'

Every interface needs to specify its gateway IP

▼ ☑ cat9k-multi
  ⦿ Default interface selection
  ◯ eth0 10.100.21.65
  ◯ eth1 10.100.22.65
  ◯ eth2 10.100.23.65
  ◯ eth3 10.100.24.65

What's Coming: Building on Alpine

Why Switch to Alpine?

1. Security Surface Area

   Absolute minimum base, only direct dependencies

2. Minimize Disk and Memory Footprint

   ~400MB to ~40MB

3. Simplify Support / Maintenance

   No significant re-architecture

4. Improved deployment performance

   Reduced size increases speed

How Far Along Are We?

prototype

We have a prototype working with the base Enterprise Agent running on x86-64 and ARM using Alpine

porting

We are still working on porting BrowserBot (component for web browser-based tests) to Alpine

validation

Additional validation on Cisco platforms is required once we are ready to begin full testing

What's Coming: ARM, Nexus, and IoT

Nexus Supports CAF in NX-OS 10.3(2)F

NX-0S 10.3(2)
Supports CAF in *stand-alone mode*

Full IPv6
& multi-interface selection

Est.

Spring '23

Nexus ACI Shipping CAF Soon

- No strict ETA

- ACI-mode firmware images on a 16.x base

- We are working on ensuring that multi-interface support works well within the complexities of an ACI fabric

- Intent is to make it easy to use the ACI Terraform provider and the ThousandEyes Terraform provider together to deploy tests across your fabric in an automated way.

ARM Enterprise Agents and IoT Platforms

- Prototype ARM agent today
  - Intending to get production ready and released this calendar year
- Once we ship ARM containers, the IoT platforms (IE switches and IR routers) already support CAF
- Specific device support details are still being worked out
  - Goal is to make the Enterprise Agent installable on devices with 4GB+ of memory and at least 4GB of usable flash

IoT Creates New Use Cases
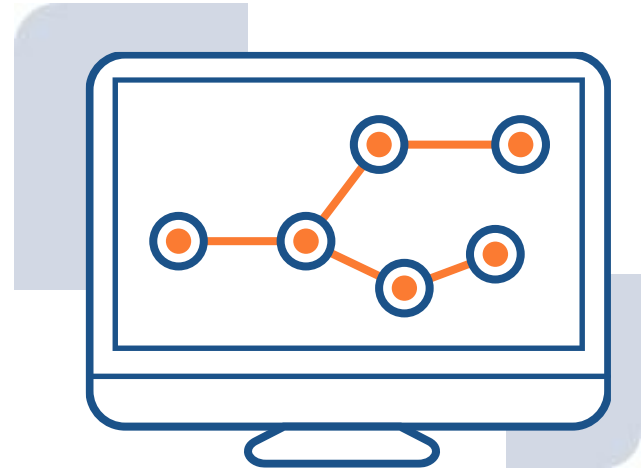
Monitoring Sensor Reachability

- Helping customers minimize truck rolls for sensor networks in the field, such as in oil and gas industry.
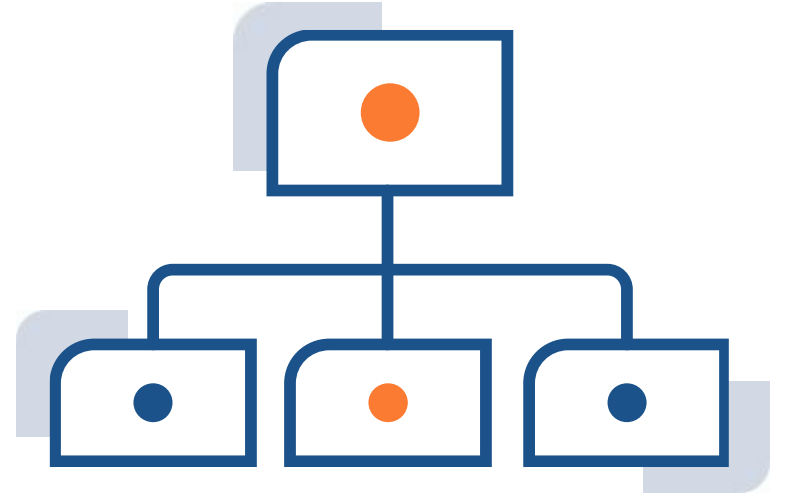
IoT Creates New Use Cases

Monitoring Robotics Safety Devices

- Use ThousandEyes agents to measure network performance that directly impacts safety limits imposed on robotic manufacturing and warehousing

IoT Creates New Use Cases

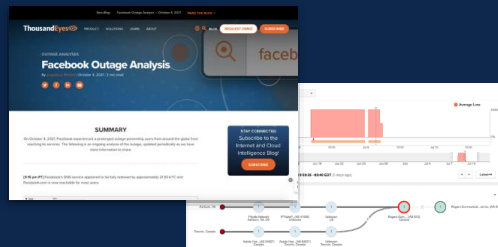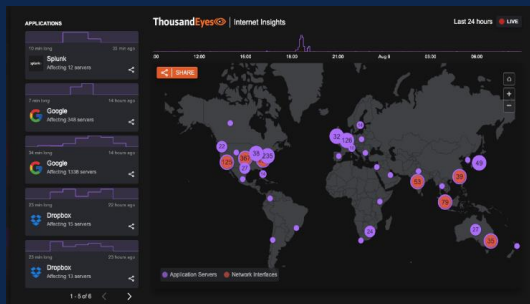## Monitoring Transportation Networks and Mobile Reachability

- Trains, Buses, and other transport systems need reliable monitoring of core backend systems such as payments, ticketing, and rider-facing wireless.

Internet outages and research
Available 24x7

ThousandEyes Internet Intelligence

Outage map: www.thousandeyes.com/outages
Outage analysis: www.thousandeyes.com/blog
Internet research: www.thousandeyes.com/research

Subscribe to our blog

Follow us on

https://www.linkedin.com/company/thousandeyes/

@thousandeyes

Complete your Session Survey

- Please complete your session survey after each session. Your feedback is important.

- Complete a minimum of 4 session surveys and the Overall Conference survey (open from Thursday) to receive your Cisco Live t-shirt.

- All surveys can be taken in the Cisco Events Mobile App or by logging in to the Session Catalog and clicking the "Attendee Dashboard" at
https://www.ciscolive.com/emea/learn/sessions/session-catalog.html

Continue Your Education

Visit the Cisco Showcase for related demos.

Book your one-on-one Meet the Engineer meeting.

Attend any of the related sessions at the DevNet, Capture the Flag, and Walk-in Labs zones.

Visit the On-Demand Library for more sessions at ciscolive.com/on-demand.

Thank you