
cisco Live!
Let's go
#CiscoLive

The bridge to possible

Journey to Hosting Containers on Meraki with ThousandEyes

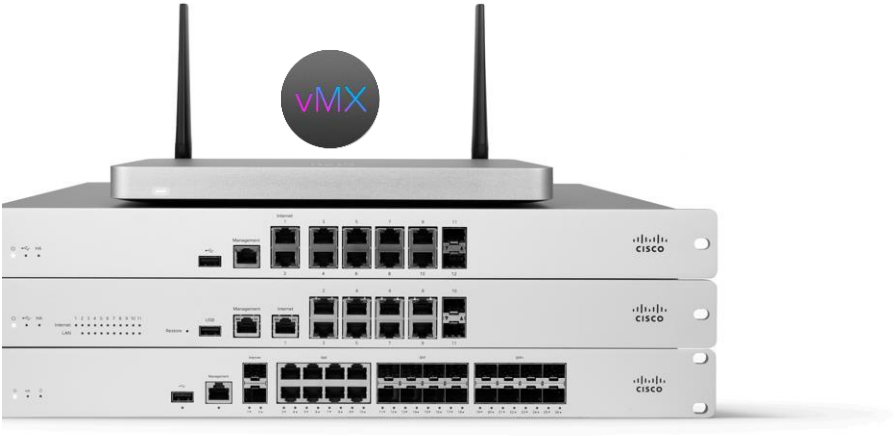
Tyler Duzan, Sr. Product Manager – ThousandEyes
Taylor Byrnes, Software Engineering Technical Leader - Meraki
BRKAPP-2727

Agenda

- Introduction
- The Meraki App Platform
- Building the Agent Container
- Demo
- Q&A

Delivered by Meraki building blocks

MX security and SD-WAN appliances



Highlights across models

- Up to ×4 WAN ports
- 3G / 4G / LTE USB as single-WAN or failover
- Models with embedded LTE modem
- High availability mode and automatic WAN failover
- Additional Ethernet ports with PoE/PoE+ options
- Virtual appliances for hybrid cloud

Network Requirements

SMALL BRANCH

MX67/68

Up to 50 users
600 Mbps FW throughput
Wi-Fi & PoE

MX67C/68CW

Up to 50 users
600 Mbps FW throughput
Wi-Fi & PoE
CAT 6 LTE modem

MX75

Up to 250 users
1 Gbps FW throughput
WAN PoE

MEDIUM TO LARGE BRANCH

MX85

Up to 250 users
1 Gbps FW throughput

MX95

Up to 500 users
2 Gbps FW throughput

MX105

Up to 750 users
3 Gbps FW throughput

CAMPUS / LARGE ENTERPRISE

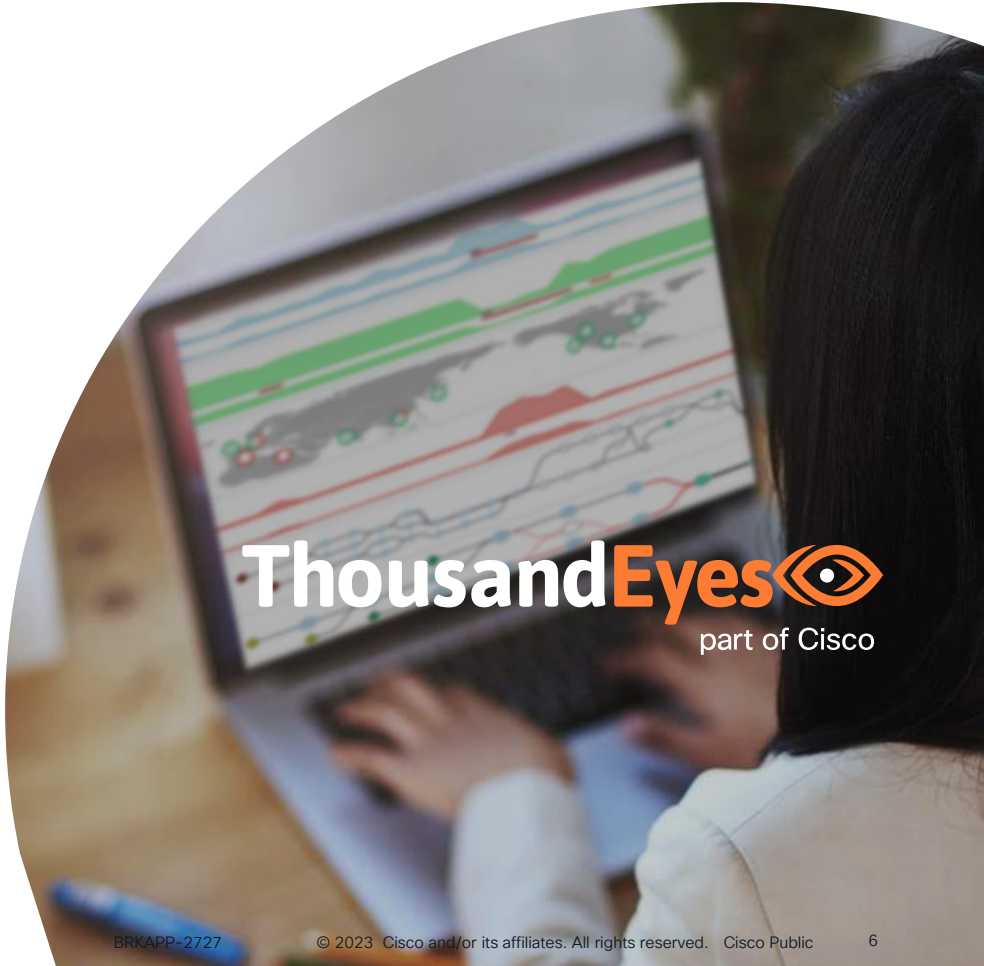
MX250

Up to 2,000 users
4 Gbps FW throughput

MX450

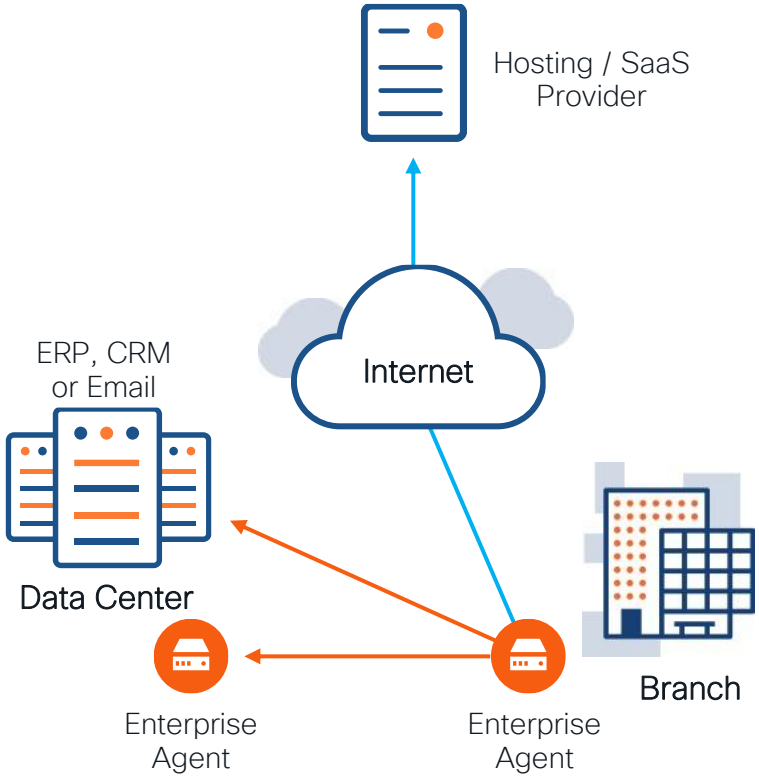
Up to 10,000 users
6 Gbps FW throughput

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



Fast Scalable SaaS App Monitoring

Benefits of ThousandEyes with Meraki – **USE CASES**

Instantly activate at **Scale**

Reduce operation workstreams with Meraki Dashboard

Continuously monitor

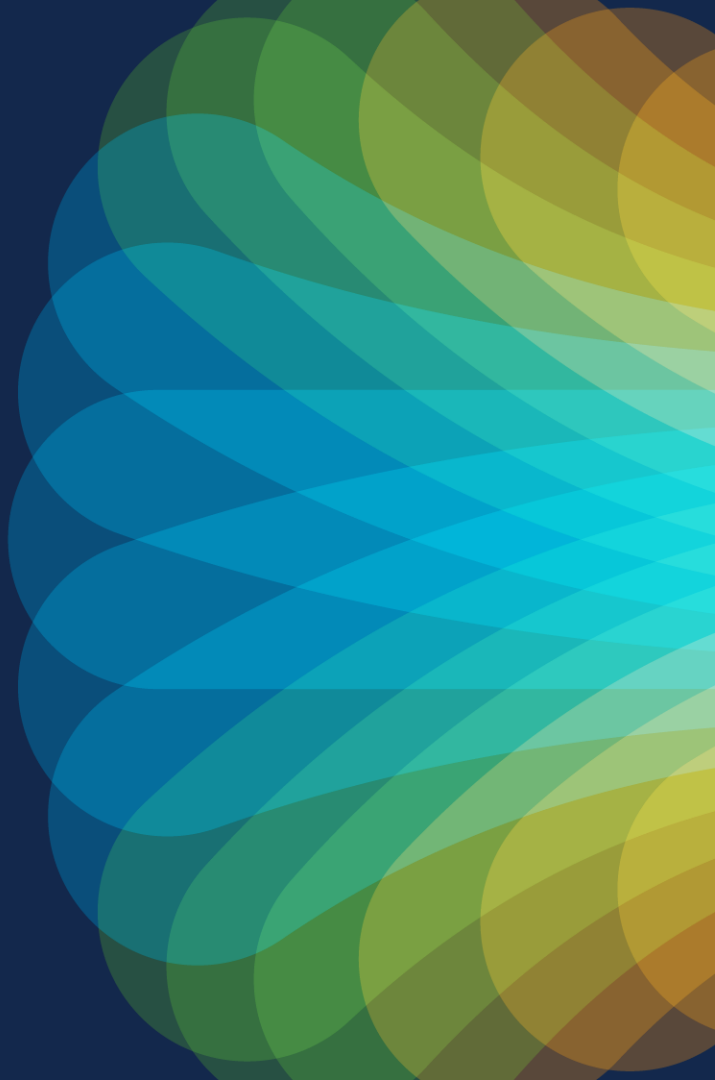
Provide visibility to SaaS apps at **regular intervals** to give IT admins more validity of the issue in addition to real time user traffic with app health

Reduce Adverse Impact

Proactive monitoring to **reduce impact** and helps communicate up and across.

Across SD-Branch Cloud Platform

The Meraki App Platform



A Disclaimer

This session is:

- A peek behind the curtain at how Meraki delivers our new integration with ThousandEyes
- A deep dive into a number of technical details that customers don't interact with
- To satisfy your curiosity

This session is not:

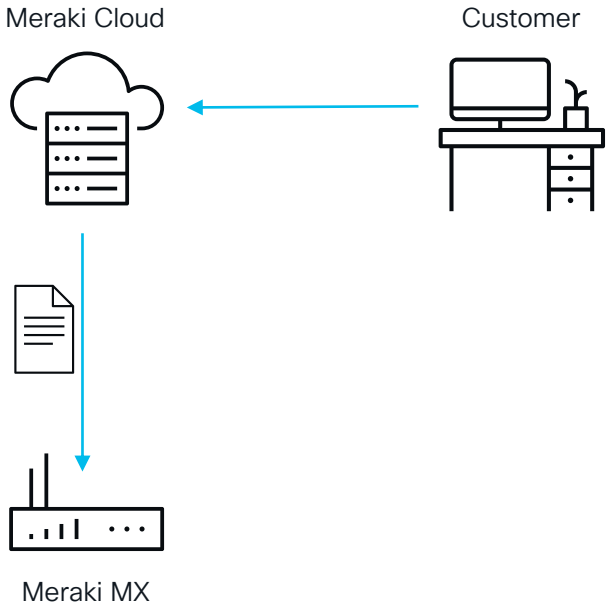
- Something customers need to understand to deploy ThousandEyes on MX
- The launch of a general-purpose app hosting platform

Why an app platform?

- Meraki firmware upgrades require a device reboot
- ThousandEyes releases Enterprise Agent updates on a two-week cycle
- Containers provide isolation and resource limiting to assure stability

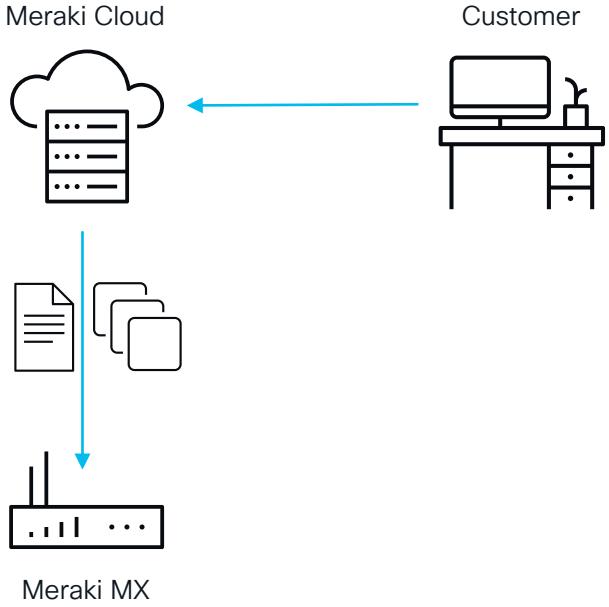
The MX Today

1. Customer changes a setting on Meraki Dashboard
2. Meraki Cloud generates a new config file
3. Meraki MX implements the new configuration



The MX, with Apps

1. Customer enables an app on Dashboard (i.e. ThousandEyes)
2. Meraki Cloud generates a new config file, which includes which apps to run
3. Meraki MX implements the new configuration, including downloading the specified apps



4 Big Questions

1. What is an app?
2. How are apps created?
3. How do apps get configured?
4. How does the MX install and run an app?

What is an App?

Meraki YAML Manifest

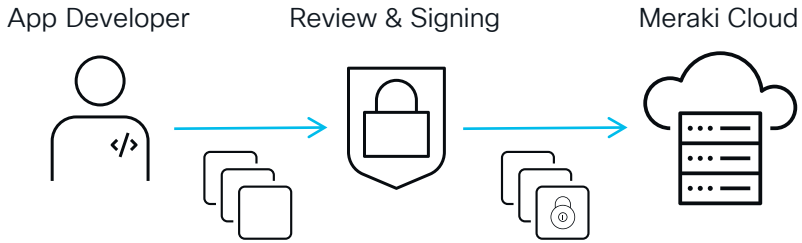
- Describe the app's system requirements and configuration
  - Limits on RAM, CPU, IO, etc.
  - Networking and storage needs
  - Linux capabilities required
- Digitally signed, and contains the container image SHA256 hash

Docker/OCI Container Image

- Industry-standard container image format
- Must support both arm64 and amd64 CPUs

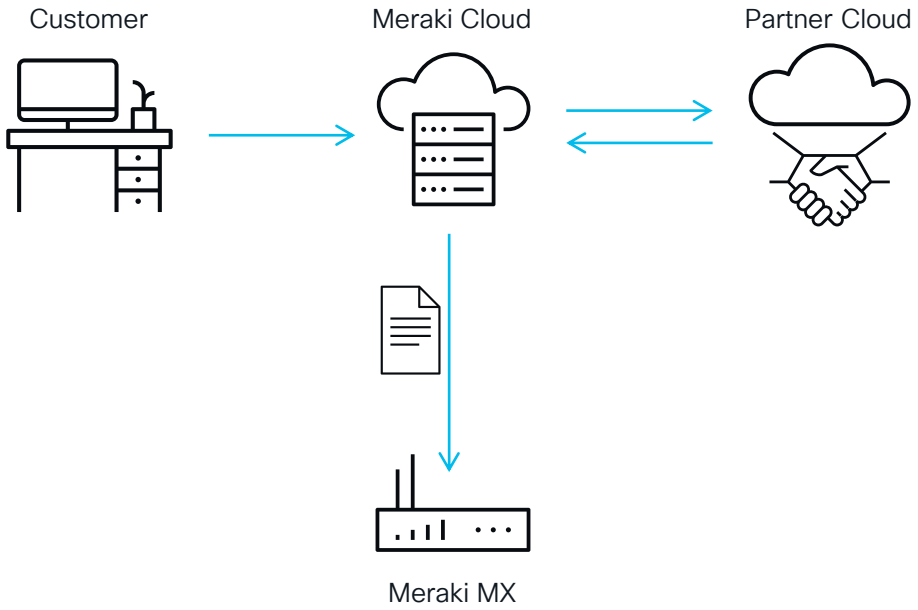
Publishing Apps

1. ThousandEyes builds a Docker/OCI container image, and writes a YAML manifest
2. The app is uploaded to Meraki
3. Meraki reviews the app, and if approved signs it for release

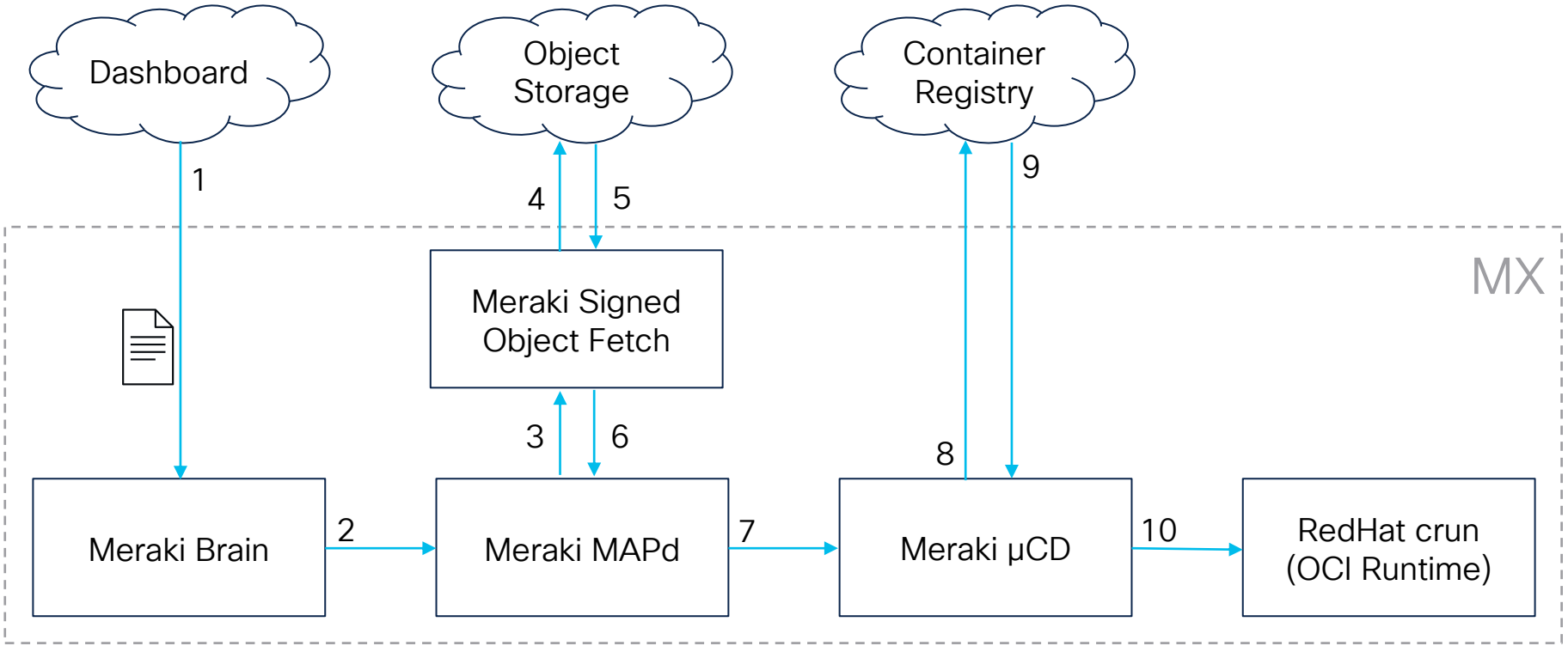


Configuring Apps

1. Install is clicked on Dashboard
2. Meraki cloud interfaces to ThousandEyes cloud, and generates an app config
   - Creates the ThousandEyes agent
   - Config includes the agent's identity
   - This replaces agent self-registration
3. App config is included in the Meraki config file



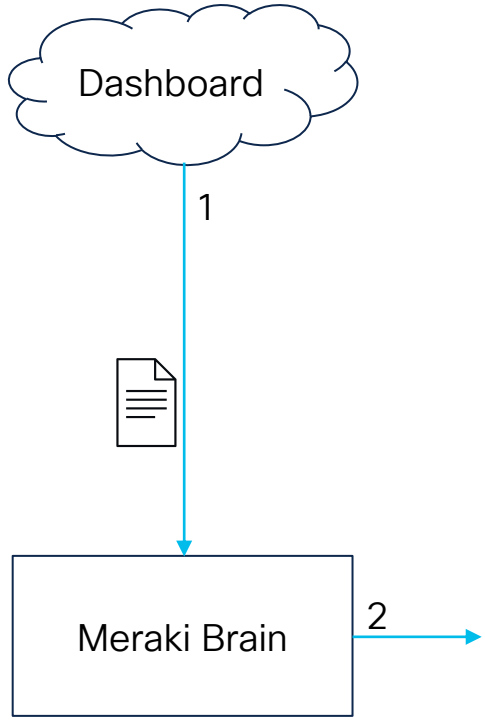
Installing & Running Apps



Installing & Running Apps, Part 1

Config Downloading

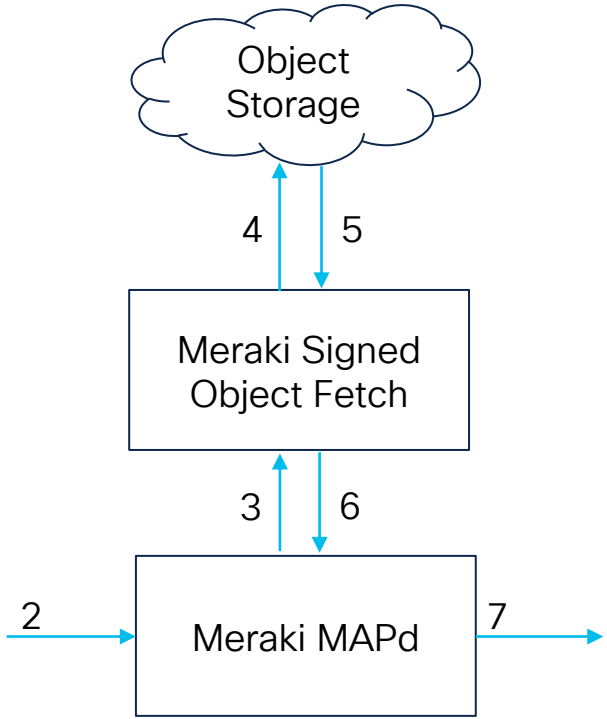
1. Meraki Dashboard delivers a new config file, containing:
   - Which apps to run
   - Provisioning information for all apps
2. Brain forwards the Meraki App Platform section to MAPd



Installing & Running Apps, Part 2

Meraki App Platform Daemon (MAPd)

- MAPd **drives system state**
  - Compares current state to desired state, and causes state transitions
- RPC interfaces to SOF and µCD
  - SOF responsible for downloading Meraki YAML Manifest files, and verifying signatures
  - µCD downloads and unpacks container images
- Makes provisioning information available to apps



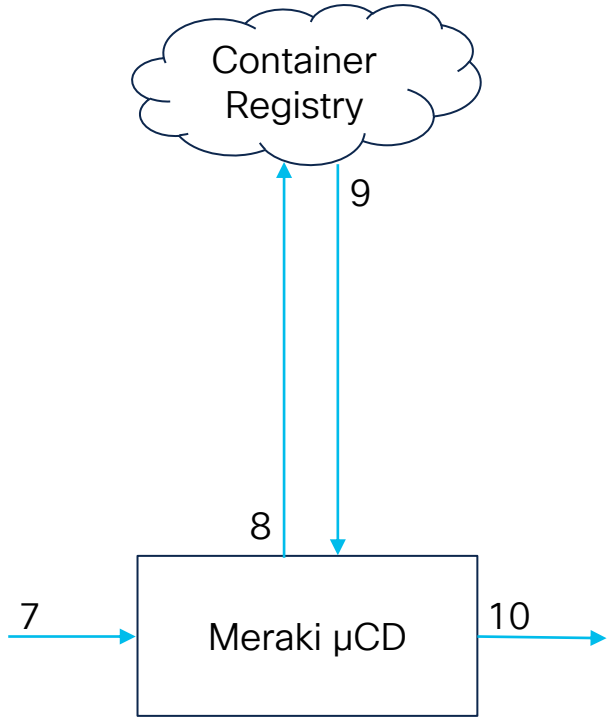
Installing & Running Apps, Part 3

Meraki Micro Container Daemon (µCD)

1. Downloads and unpacks container images
2. Generates OCI Runtime configuration file

Ultra-low resource equivalent to Docker containerd or CNCF CRI-O

Driven by RPC from MAPd

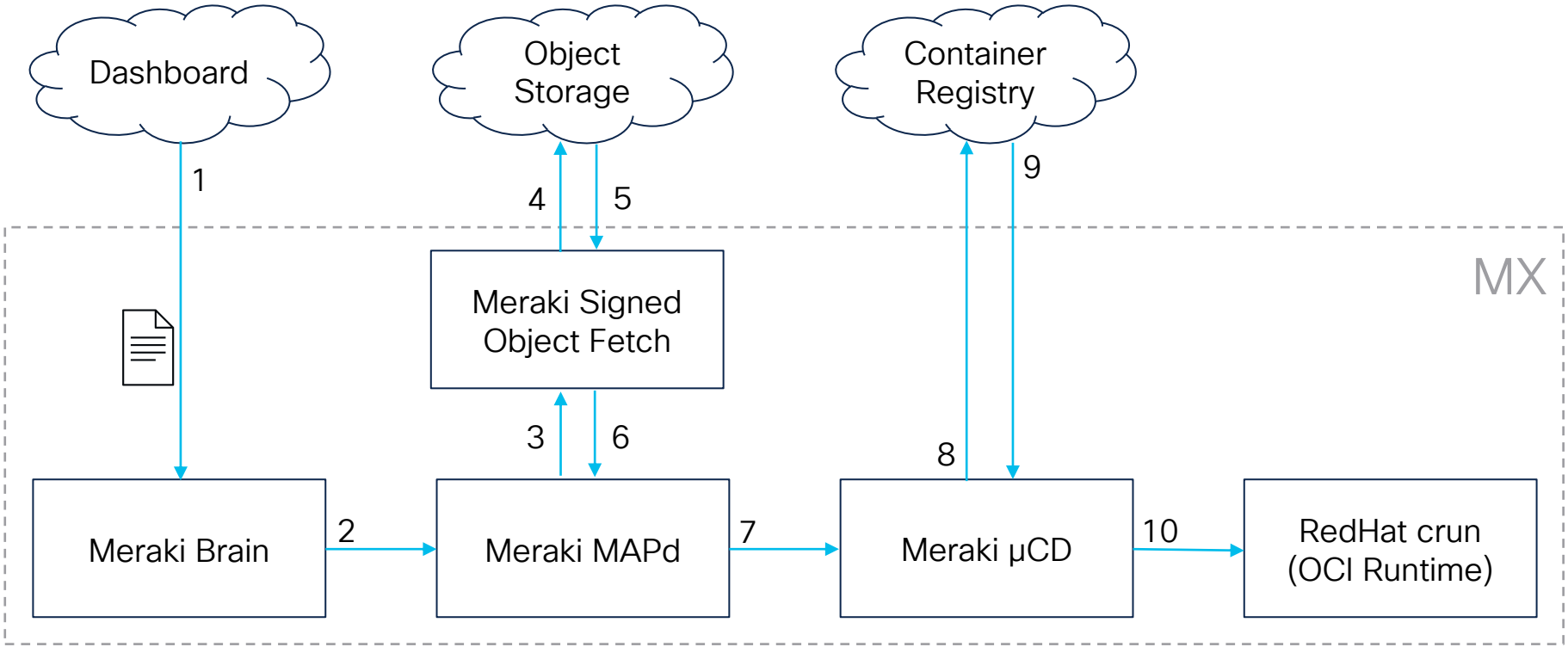


Installing & Running Apps, Part 4

OCI Runtime

- Executes an OCI Filesystem Bundle (created by µCD)
- Performs the low-level operations that create a container (e.g. Linux namespace creation)
- Meraki App Platform uses RedHat's crun, but compatible with any compliant OCI Runtime

Installing & Running Apps

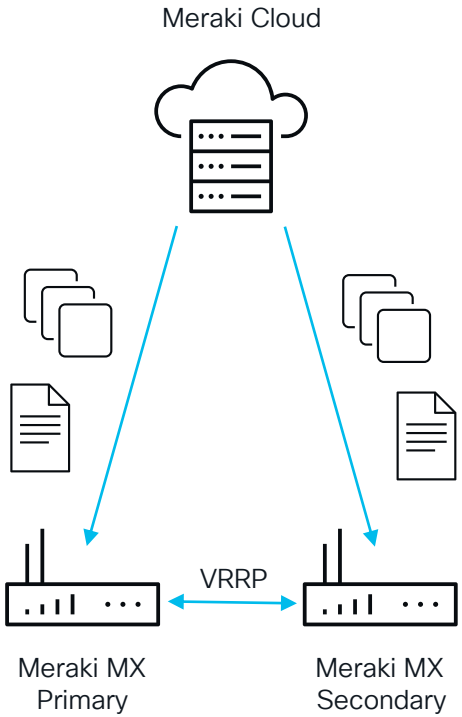


Bringing Meraki Magic to Apps

- Automatic HA failover
- Pre-configuration of apps before hardware setup
- Seamless hardware swaps and upgrades
- SaaS-like vendor-managed app updates

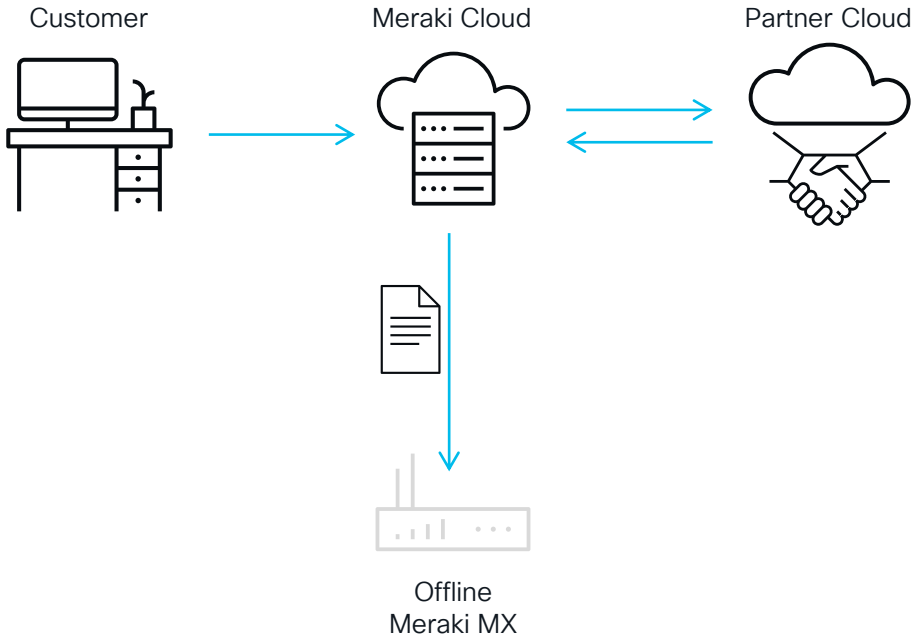
HA Failover

- App + config is delivered to both primary and secondary MX
- MAPd continuously monitors VRRP state and starts/stops apps as required
- Single agent identity is automatically transferred



Pre-Deployment Configuration

- ThousandEyes can be enabled, and the agent created, before the MX is online
- Meraki Cloud stores the app configuration, and provides it to the MX on first boot
- Allows pre-deployment configuration of tests

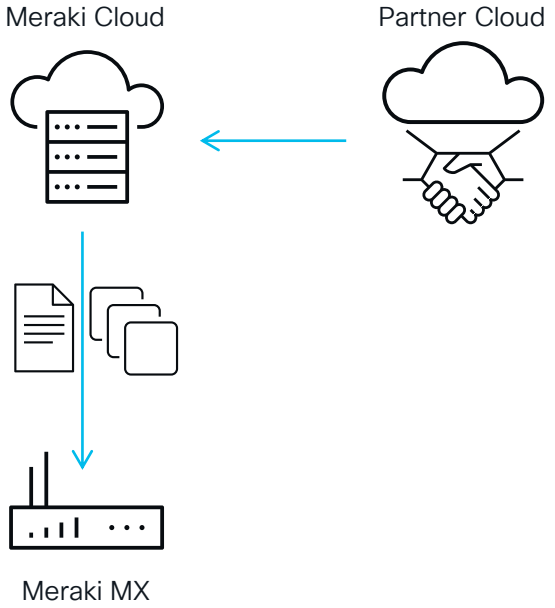


Seamless Hardware Replacement

- App config is automatically transferred on MX replacement
- Meraki cloud is the authoritative source of all app configuration
  - Data is never linked to specific hardware, only to the network
  - ThousandEyes agent identity is seamlessly transferred

SaaS-like App Updates

- After app version approval by Meraki, individual agent updates are initiated by ThousandEyes
  - Follows standard ThousandEyes progressive release process
  - Not the typical self-updating ThousandEyes agent model
- Apps are sandboxed during execution to ensure MX stability



The Meraki App Platform significantly simplifies deployment of Cisco software at the network edge

Building the Agent Container

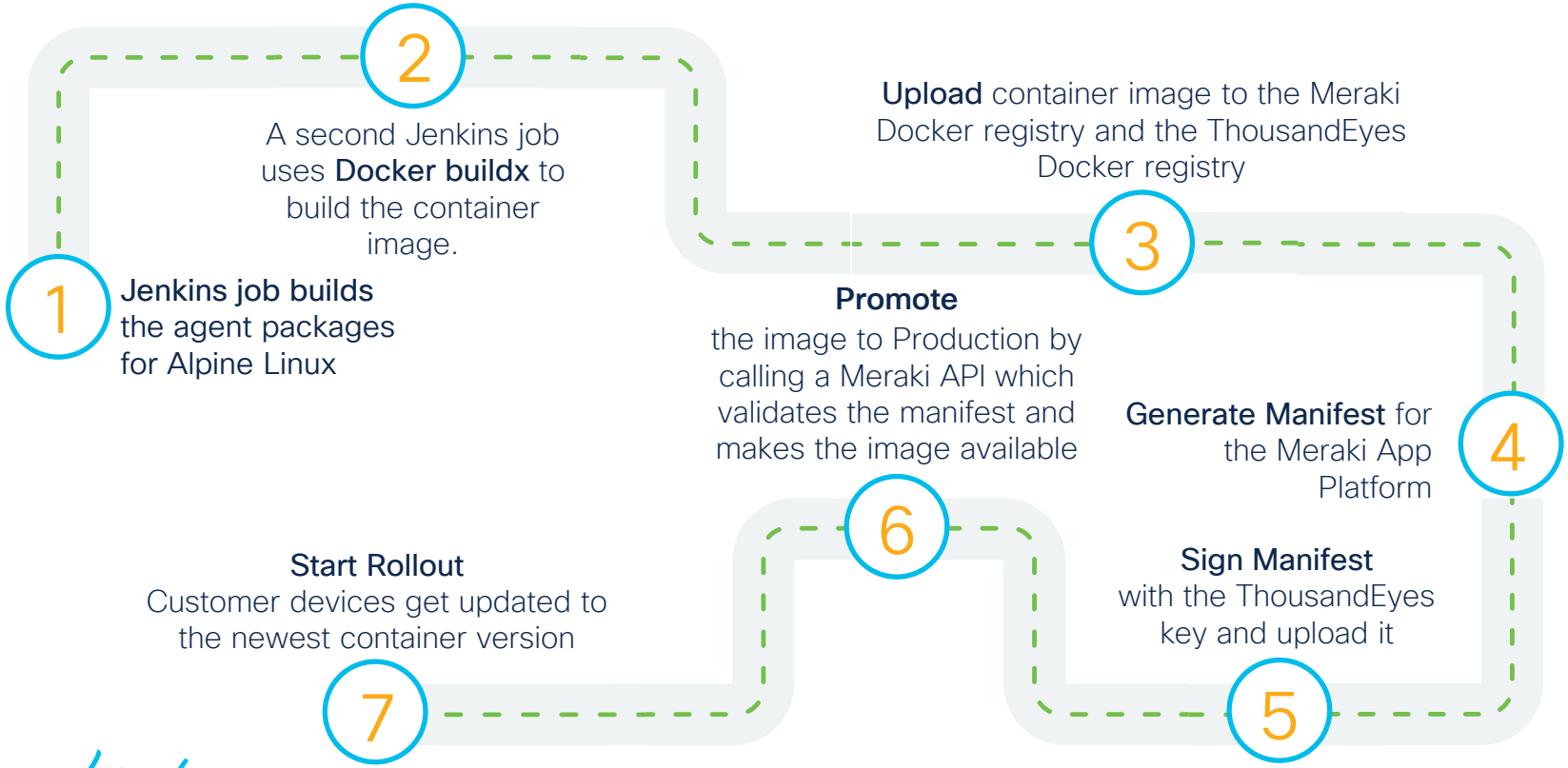
Constraints & Requirements

- We needed to support devices which only have 4GB of physical memory
- Our agent container must fit within less than 100MB on disk
- We must support both ARM64 and X86 architectures
- We need to run on a read-only filesystem
- The Enterprise Agent cannot self-register to our platform
- We cannot exceed 400MB of memory utilization during runtime, including tmpfs

Why Switch to Alpine?

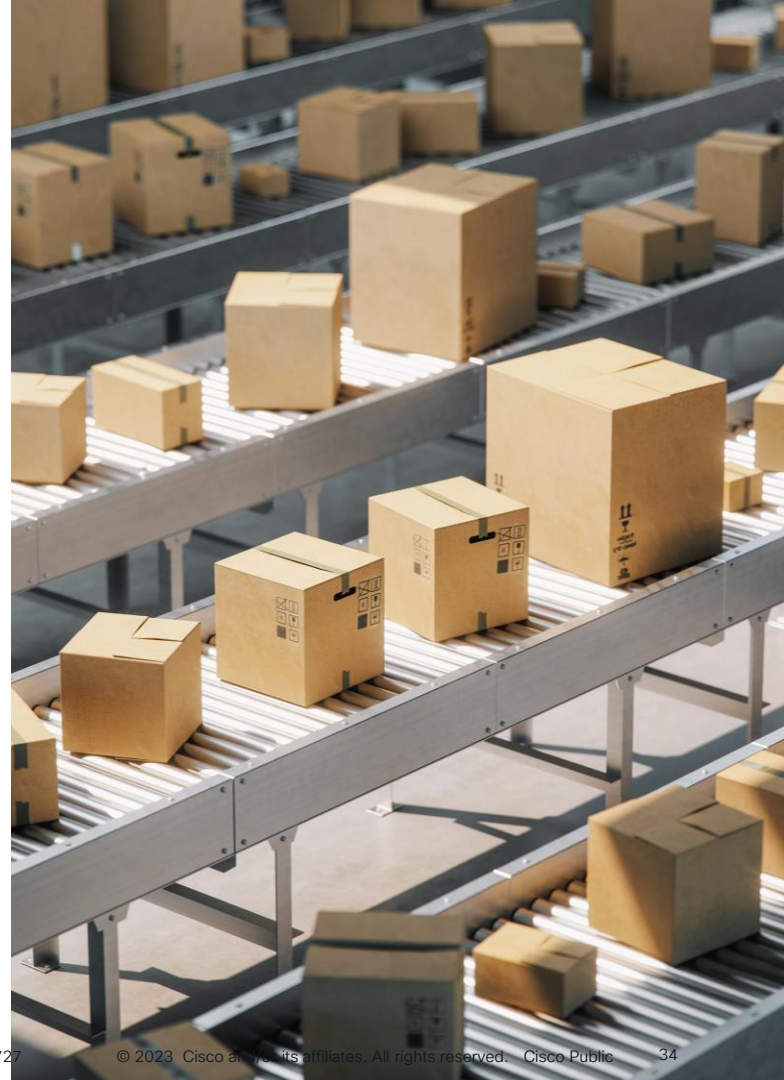
1. Security Surface Area
   Absolute minimum base, only direct dependencies
2. Minimize Disk Footprint
   ~400MB to ~40MB on disk
3. Simplify Support / Maintenance
   No significant re-architecture
4. Improved deployment performance
   Reduced size increases speed

The ThousandEyes Build Process for Meraki



Package-Centric Approach

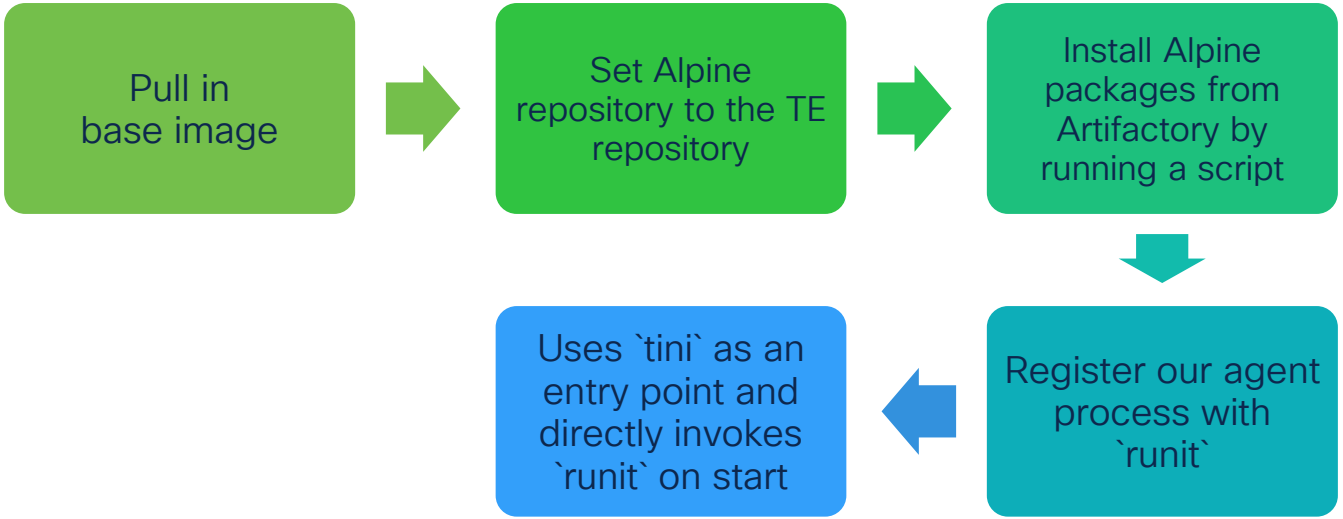
- All components of the ThousandEyes Enterprise Agent are packaged as APKs for Alpine Linux
- Containers are built by installing packages using a script run during the build process.
- No non-package customization, and the container runs on a read-only filesystem



Simplistic Dockerfile

We use the official images for Alpine Linux

The Dockerfile process:



Dealing With Storage Using tmpfs

Key Challenge

Balancing logs and results cache size with memory utilization

Our Solution

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

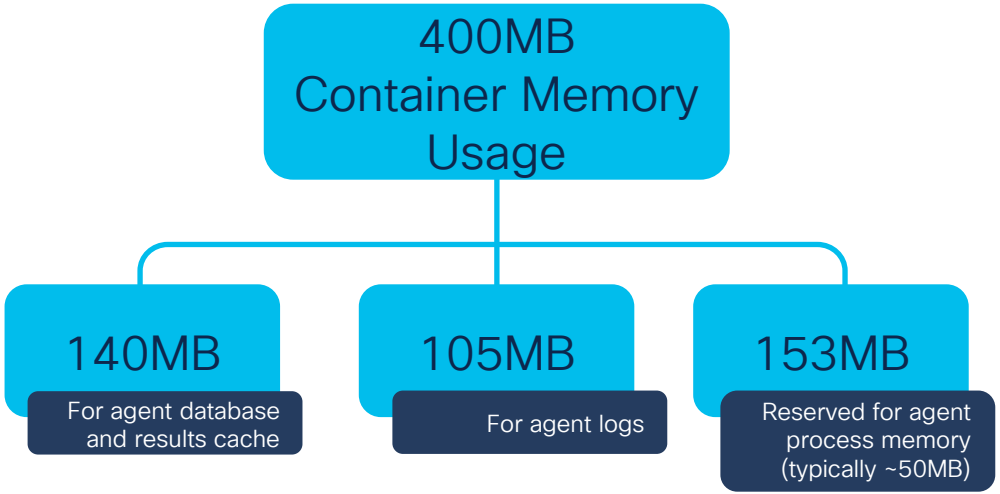
Things to Note

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 agent can work with a read-only filesystem
- All storage in memory is accounted as part of container memory usage
- MAP sets cgroups rules from our container manifest and enforces memory limits



Networking Capabilities Required

Enterprise Agent uses raw sockets to generate TCP probes

CAP_NET_RAW is required to run our containers

Demo

Fill out your session surveys!

Attendees who fill out a minimum of four session surveys and the overall event survey will get **Cisco Live-branded socks** (while supplies last)!

Attendees will also earn 100 points in the **Cisco Live Challenge** for every survey completed.

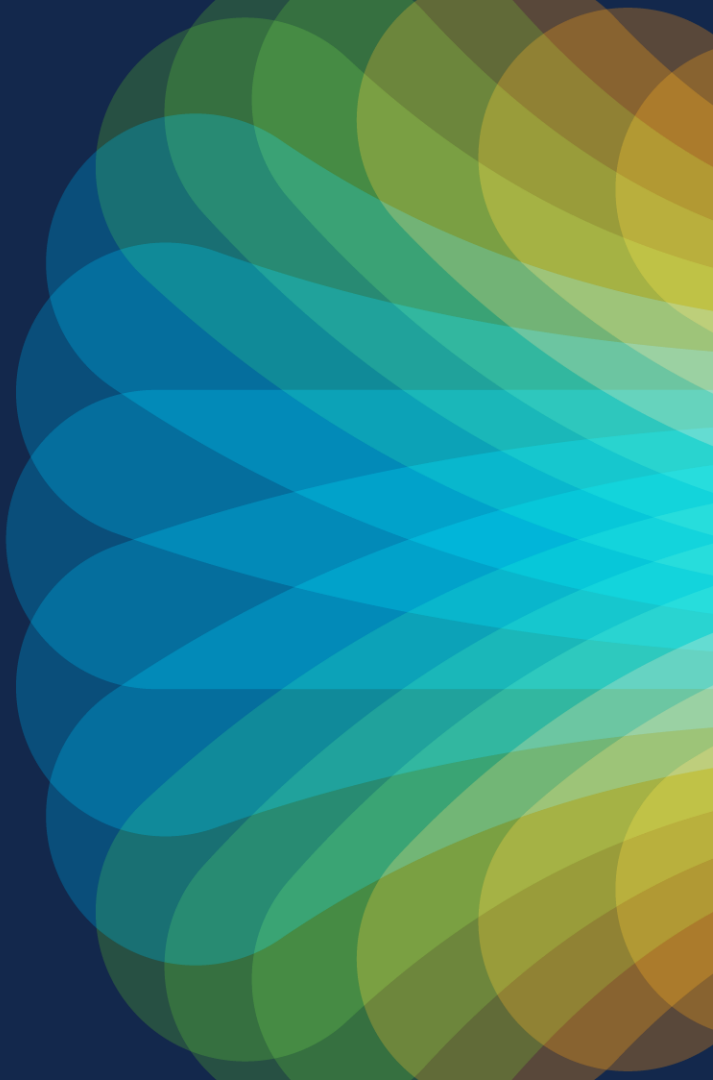
These points help you get on the leaderboard and increase your chances of winning daily and grand prizes

Continue your education

- Visit the Cisco Showcase for related demos in Cisco Networking booth #3303
- Book your one-on-one Meet the Engineer meeting
- Attend the interactive education with DevNet, Capture the Flag, and Walk-in Labs
- Visit the On-Demand Library for more sessions at www.CiscoLive.com/on-demand

Got app use cases?

map-public@cisco.com



The bridge to possible

Thank you

cisco Live!

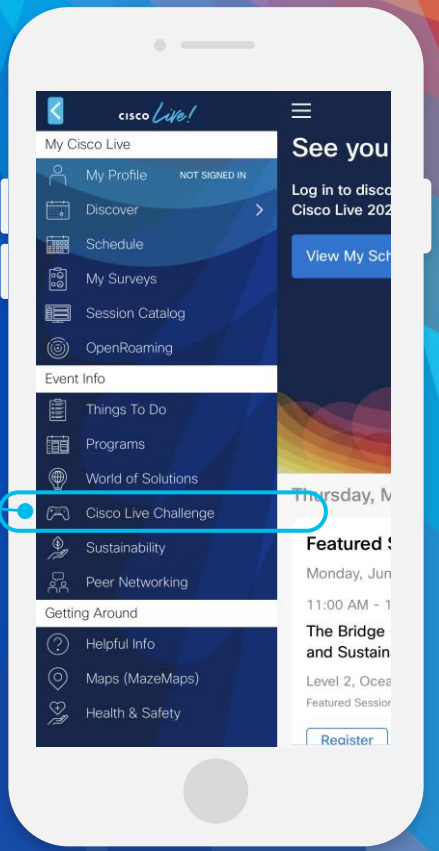
Cisco Live
Challenge

Gamify your Cisco Live experience!

Get points for attending this session!

How:

1. Open the Cisco Events App.
2. Click on 'Cisco Live Challenge' in the side menu.
3. Click on View Your Badges at the top.
4. Click the + at the bottom of the screen and scan the QR code:



cisco Live!


cisco Live!
Let's go
#CiscoLive