

The background features a vibrant, multi-colored abstract design. On the left, there are overlapping, wavy bands of color in shades of orange, red, and yellow. On the right, a bright white light source emits a series of colorful rays in shades of blue, green, and yellow, creating a sunburst effect. The overall composition is dynamic and energetic.

cisco *Live!*

Let's go

#CiscoLive



The bridge to possible

Automating ThousandEyes

Tests and Agents

Bernie Clairmont - Product Solution Architect

@BernieClairmont

BRKAPP-2001



#CiscoLive



I'm a Product Solutions Architect at ThousandEyes, a Cisco Company, helping to innovate and revolutionize network, cloud and digital service reliability. I've been evangelizing network and performance management for over 25 years, starting out as a system administrator, and then moving into network engineering and network, server and storage management for telecommunications and cable providers. I love helping customers innovate the way they manage their network infrastructure.

Bernie Clairmont

Email: bclairmo@cisco.com

LinkedIn: <https://www.linkedin.com/in/bernardclairmont/>



Agenda

- ThousandEyes Overview
- ThousandEyes API Overview
- Benefits of Automation
- Ansible Enterprise Agent Deployment Demo
- Test and Snapshots API
- Postman API Demo
- Resources
- Conclusion

ThousandEyes Overview



Visibility and control is shrinking

The Challenge



Datacenter



Cloud

External dependencies are exponentially growing



App Stack



SaaS + API



Office



Home



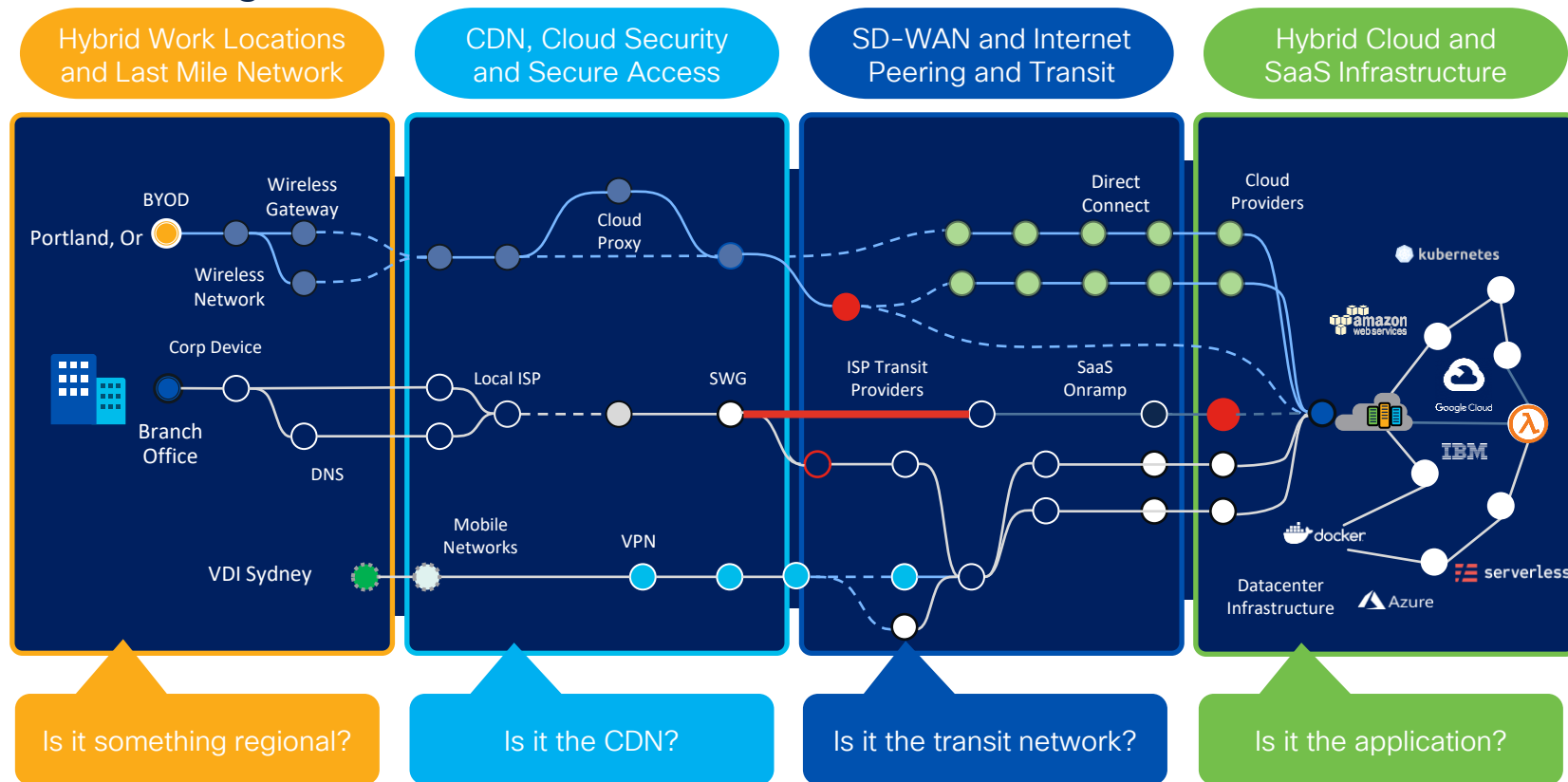
Network



Internet

Massive blind spots erode ability to manage

The Challenge



ThousandEyes Internet and Cloud Intelligence

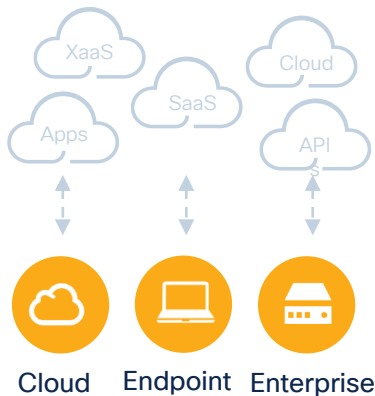
Our Solution

VISIBILITY

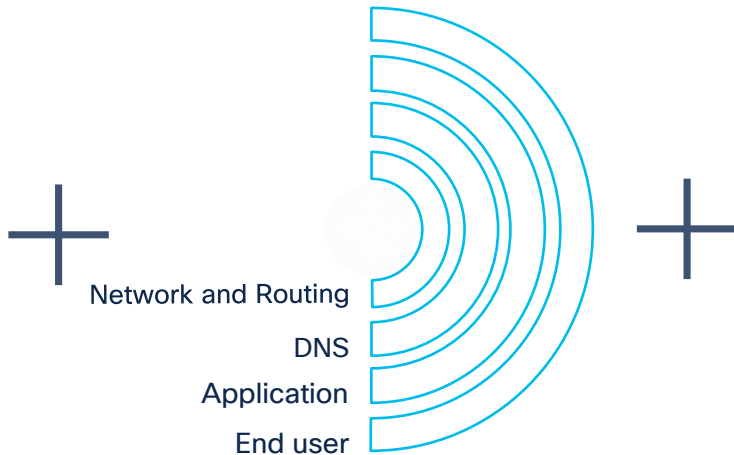
INTELLIGENCE

WORKFLOWS

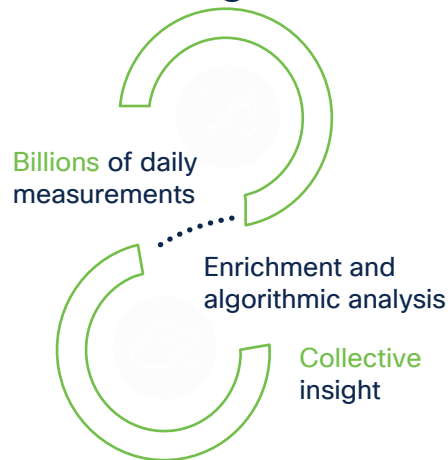
1000s Global Vantage Points



Unique X-layer Telemetry



Global Collective Insight



ThousandEyes Internet and Cloud Intelligence

Our Solution

VISIBILITY

INTELLIGENCE

WORKFLOWS



- REST API
- Native Integrations
- Custom webhooks
- Cloud templates
- Infra as Code tools
- Sharelinks for easy sharing across teams



SREs / App owners

Troubleshoot and optimize App, APIs and delivery services



NetOps (Service delivery)

Tune BGP and peering and monitor for anomalies, hijacks and sub-optimal routing



Customer Support

Build trust with timely, data-driven communication



IT Helpdesk

Rapidly prioritize, resolve and direct issues to the right team to ensure workforce productivity



Enterprise WAN Ops

Proactively manage and improve SD-WAN and cloud connectivity to assure app performance for users across sites



External Teams

Receive detailed, actionable information to quickly resolve issues

ThousandEyes API Overview

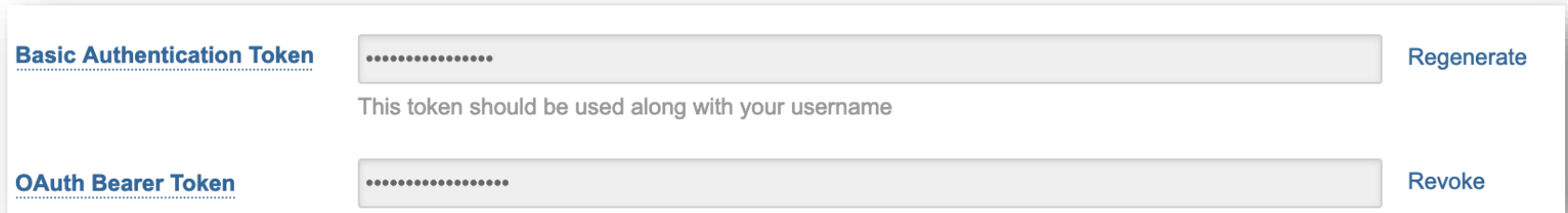
Why use APIs when the ThousandEyes UI is so awesome?

- Integration with other monitoring tools for a single pane of glass
- On demand tests and agents as part of troubleshooting sessions
- User interface customizations
- ThousandEyes as part of the DevOps toolset
- Automated network sanity checks after network upgrades or changes
- Data Lake for historical storage and analysis

<https://docs.thousandeyes.com/product-documentation/getting-started/getting-started-with-the-thousandeyes-api>

ThousandEyes Programmable Interfaces

- Base URL is <https://api.thousandeyes.com>
- Authentication type can be basic or bearer token, operation set with a mix of HTTP method and URL *
- XML and JSON formats are supported



The screenshot shows a user interface for managing API tokens. It contains two sections: 'Basic Authentication Token' and 'OAuth Bearer Token'. Each section has a text input field with masked characters (dots), a 'Regenerate' button for the basic token and a 'Revoke' button for the OAuth token, and a descriptive text line below the input field.

Token Type	Token Value	Action	Description
Basic Authentication Token	Regenerate	This token should be used along with your username
OAuth Bearer Token	Revoke	

* Basic Auth is not available in EU region

ThousandEyes Programmable Interfaces


- Available API endpoints:
 - Tests & Instant Tests
 - Snapshots
 - Alerts and Notifications
 - Enterprise Agents
 - Endpoint Agents
 - Labels and Dashboards
 - Much, much more...

Documentation at <https://developer.thousandeyes.com>


Benefits of Automation: Enterprise Agent Deployment

Enterprise Agents Install Types

- OVA – For manual install (it can be automated, but there are better options)
- Linux installer – Script available in the dashboard
- Docker – Steps available in the dashboard
- More options available

Add New Enterprise Agent 

Appliance **Custom Appliance** **Cisco Application Hosting** **Linux Package** **Docker** **IaaS Marketplaces**

Account Group Token  Copy

Virtual Appliance

Automated Install Examples

- Terraform and cloud-init can be used to install pre-requirements and software into a new VM
- Ansible can be used to install Linux agent or a docker agent in existing virtual machines
- Agents API can be used to validate that the agent has registered correctly

```
33 {
34   "agentId": 203041,
35   "agentName": "lwr04-xrv-vrr-02-te-agent-216796",
36   "agentType": "Enterprise",
37   "countryId": "US",
38   "enabled": 1,
39   "keepBrowserCache": 0,
40   "verifySslCertificates": 1,
41   "ipAddresses": [
42     "10.90.54.188",
43     "1.1.1.3"
44   ],
45   "lastSeen": "2021-03-26 21:04:54",
46   "location": "Atlanta Area",
47   "network": "Cisco Systems, Inc. (AS 109)",
48   "prefix": "173.38.0.0/17",
49   "publicIpAddresses": [
50     "173.38.117.77"
51   ],
52   "targetForTests": "10.90.54.188",
53   "agentState": "Offline",
54   "ipv6Policy": "FORCE_IPV4",
55   "hostname": "lwr04-xrv-vrr-02-te-agent",
56   "createdDate": "2021-02-22 22:52:46",
57   "errorDetails": []
58 }
```

<https://api.thousandeyes.com/v6/agents.json?agentTypes=ENTERPRISE>

Ansible Demo: Enterprise Agent Deployment

Test and Snapshots API

ThousandEyes Test APIs

- Recurrent Test: <https://api.thousandeyes.com/v6/tests>
- Instant Test: <https://api.thousandeyes.com/v6/instant>
- The URL is different per test type – for example:

Name	URL
Agent to Server	/v6/tests/ agent-to-server /new.json
Agent to Agent	/v6/tests/ agent-to-agent /new.json
BGP	/v6/tests/ bgp /new.json
Page Load	/v6/tests/ page-load /new.json
Transactions	/v6/tests/ transactions /new.json

Full list at https://developer.thousandeyes.com/v6/tests/#/test_creation

Create: HTTP Server Test

Test type

Operation

Test URL

Format

The screenshot shows a REST client interface with the following configuration:

- Method:** POST
- Test URL:** `https://api.thousandeyes.com/v6/tests`
- Test type:** http-server
- Operation:** new
- Format:** json
- Body:**

```
{
  "interval": 60,
  "agents": [
    {
      "agentId": 387771
    }
  ],
  "testName": "API server test addition for http://192.168.20.22",
  "url": "http://10.90.54.152:8080",
  "protocol": "TCP",
  "alertsEnabled": 0,
  "bgpMeasurements": 0
}
```

Retrieve: Test Metrics

- API Endpoint:
https://api.thousandeyes.com/v6/net/metrics/{test_id}
- Test ID needs to be obtained first via either:
 - Get tests and filter: <https://api.thousandeyes.com/v6/tests>
 - Response body from test creation

```
"metrics": [  
  {  
    "avgLatency": 74.0,  
    "loss": 0.0,  
    "maxLatency": 78.0,  
    "jitter": 0.73440033,  
    "minLatency": 73.0,  
    "serverIp": "10.94.158.206",  
    "agentName": "sfloresk-lwr01",  
    "countryId": "US",  
    "date": "2022-03-08 22:12:02",  
    "agentId": 347216,  
    "roundId": 1646777520,  
    "permalink": "https://app.thousandeyes.com/view/tests?__a=190644&testId=2622584&roundId=1646777520&agentId=347216"  
  }  
]
```

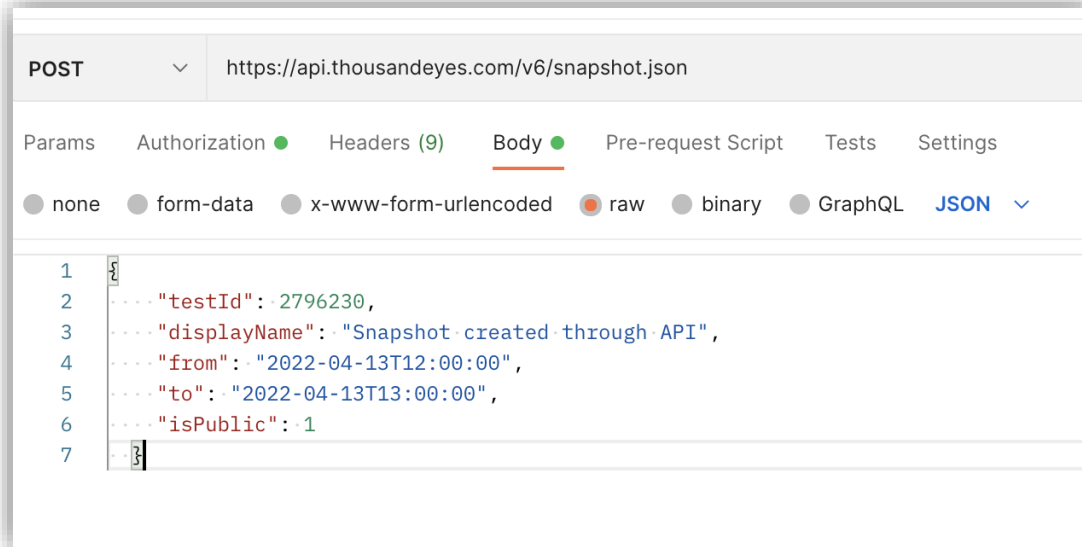
Retrieve: Network Path

- https://api.thousandeyes.com/v6/net/metrics/{test_id}/{agent_id}/{round_id}
- Round ID and agent ID need to be obtained first via network metrics or network path APIs

```
"routes": [  
  {  
    "pathId": "20657547682632584681731393106177968817249522579839969819205053747",  
    "hops": [  
      {  
        "hop": 1,  
        "ipAddress": "10.90.54.130",  
        "responseTime": 10,  
        "location": "North Carolina, US"  
      },  
      {  
        "hop": 2,  
        "ipAddress": "10.90.54.129",  
        "responseTime": 3,  
        "location": "North Carolina, US"  
      },  
      {  
        "hop": 3,  
        "ipAddress": "10.90.244.13",  
        "rdns": "lwr01-00-lab-gw2-ten5-5.cisco.com",  
        "responseTime": 0,  
        "location": "North Carolina, US"  
      }  
    ]  
  }  
]
```

Create: Snapshot API

- <https://api.thousandeyes.com/v6/snapshot.json>
- Requires test ID, a name and the date range
- Returns a URL that can be viewed and shared to external users



Test API Summary

- 1 Retrieve agent IDs
- 2 Create test using agent IDs – save the test ID
- 3 Retrieve metrics and path overview using test ID – retrieve round ID
- 4 Retrieve path hops using test ID, agent ID and round ID

Postman Demo: Test and Snapshot API

Alerts and Webhooks API

Alerts Rules and Alerts

Alert rules

Defines alert attributes

Very flexible expressions
response time $\leq 500\text{ms}$

Customizable triggers

All conditions are met by the same agents 3
times out of 5 test in a row

Alerts

Instance of an alert rule for a given test

Triggers notifications such as emails or
webhooks

Can be “paused” for maintenance
windows

<https://docs.thousandeyes.com/product-documentation/getting-started/getting-started-with-alerts>

Alerts and Alert Rules API

- Retrieve active alerts
- Create, update and delete alert rules
- The test assignment to an alert rule is done from the alert rule directly
- Webhooks need to be created in the dashboard in advance
 - API endpoint (future)
- Custom Webhooks can integrate into things like Webex and Microsoft Teams
 - Custom Webhooks can have alert logic (if, then, else, etc.)
- Tests need to have the “**alertsEnable**” parameter set to 1

Alert Rules API Data Model – Reference

- https://developer.thousandeyes.com/v6/alerts/#/alert_rule_metadata

Field	Creation	Update	Data Type	Accepts	Notes/Example
alertType	required	required	string	Acceptable test types, verbose names.	Page Load, HTTP Server, End-to-End (Server), End-to-End (Agent), Voice, DNS+ Domain, DNS+ Server, DNS Server, DNS Trace, DNSSEC, Transactions, Web Transactions, BGP, Path Trace, FTP, SIP Server
ruleName	required	required	string	name of the alert rule	the name of the alert rule
default	optional	optional	bit	to set the rule as a default, set this value to 1.	Defaults to 0
expression	required	required	string	Expression string should be a valid JSON string.	See examples in the table below. Use examples found on the Alert Rule Detail endpoint.
notifications	optional	optional	object with notification properties	See notifications endpoint for examples	<pre>"notifications": { "email": { "message": "NA", "recipient": ["noreply@thousandeyes.com"] }, "thirdParty": [{ "integrationId": "sl-101", "integrationType": "SLACK" }, { "integrationId": "pgd-21", "integrationType": "PAGER_DUTY" }], "webhook": [{ "integrationId": "wb-201", "integrationType": "WEBHOOK" }] },</pre>

Alert Rule Expressions

- Conditionals that when they evaluate to true, the alert is triggered
 - Can be combined with OR/AND operators
- Not all expressions are available or applicable for all alert types
 - ftpReplyCode does not apply to HTTP tests

Alert Rules Expressions – Reference

- https://developer.thousandeyes.com/v6/alerts/#/alert_rule_metadata

Alert type	Metric	Operator	Data type	Example	Notes
HTTP Server	responseCode	[<, <=, ==, !=, >=, >]	Integer	<code>((responseCode >= 400))</code>	HTTP response code returned by the test
HTTP Server	dnsTime	[<, <=, ==, !=, >=, >]	Integer + units	<code>((dnsTime >= 500 ms))</code>	ensure that you set units in the expression
HTTP Server	connectTime	[<, <=, ==, !=, >=, >]	Integer + units	<code>((connectTime >= 500 ms))</code>	ensure that you set units in the expression
HTTP Server	sslTime	[<, <=, ==, !=, >=, >]	Integer + units	<code>((sslTime >= 500 ms))</code>	ensure that you set units in the expression

Alerts API Summary

- 1 Create a test and save the test id
- 2 Create a new alert rule with the test recently created or update an existing one (If using webhooks, retrieve webhook information prior to creating the rule)
- 3 Listen for webhook notifications (recommended) or retrieve active alerts periodically (not recommended)

Postman Demo: Creating Alert-Rules and Retrieving Active Alerts

References

- API docs: <https://developer.thousandeyes.com>
- Test: https://developer.thousandeyes.com/v6/tests/#/test_creation
- Alerts: <https://docs.thousandeyes.com/product-documentation/alerts>
- Alert metadata:
https://developer.thousandeyes.com/v6/alerts/#/alert_rule_metadata
- Custom Webhooks: <https://docs.thousandeyes.com/product-documentation/alerts/integrations/webhooks/custom-webhooks>
- Postman collection:
<https://www.postman.com/ciscoddevnet/workspace/cisco-devnet-s-public-workspace/documentation/134222-0715abb3-9d2d-4724-9544-69d2dac6feff>

Conclusion

Why ThousandEyes

See, understand and improve digital experience everywhere

Broad Data Collection

Deliver ubiquitous visibility

E2E visibility,
synthetic, and cloud
network visibility
across every
vantage point that
matters

Proactive Intelligence

Identify and isolate issues quickly

Event detection,
correlation, and
causal analysis for
intelligent
troubleshooting

Operational Workflows

Integrate with a connected ecosystem

Unlock intelligence
and data across
ecosystems via open
standards, APIs and
automation

Cisco ThousandEyes assuring every element of modern digital experience



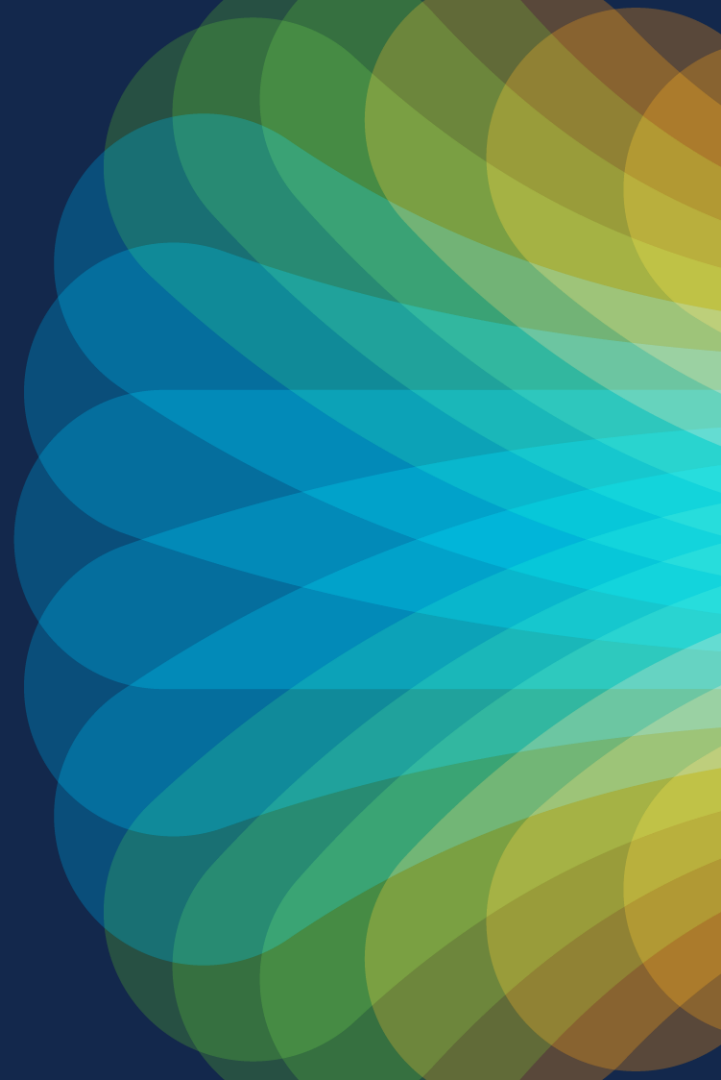


The bridge to possible

Thank you

CISCO *Live!*

#CiscoLive



The background is a vibrant, abstract graphic. It features a central bright white light source from which numerous colorful rays emanate, creating a sunburst or starburst effect. The rays transition through a spectrum of colors: yellow, orange, red, and then various shades of blue and green. Overlaid on this are several large, semi-transparent, wavy shapes in similar color tones, giving the overall image a sense of motion and energy.

cisco *Live!*

Let's go

#CiscoLive