TransFICC

Zero-Regression Network CI/CD for Finance-grade reliability

SREcon 2025 Finance InFocus Track

Co-presented by
Damian Krogul and David Ferrandez

Do you really know your network?

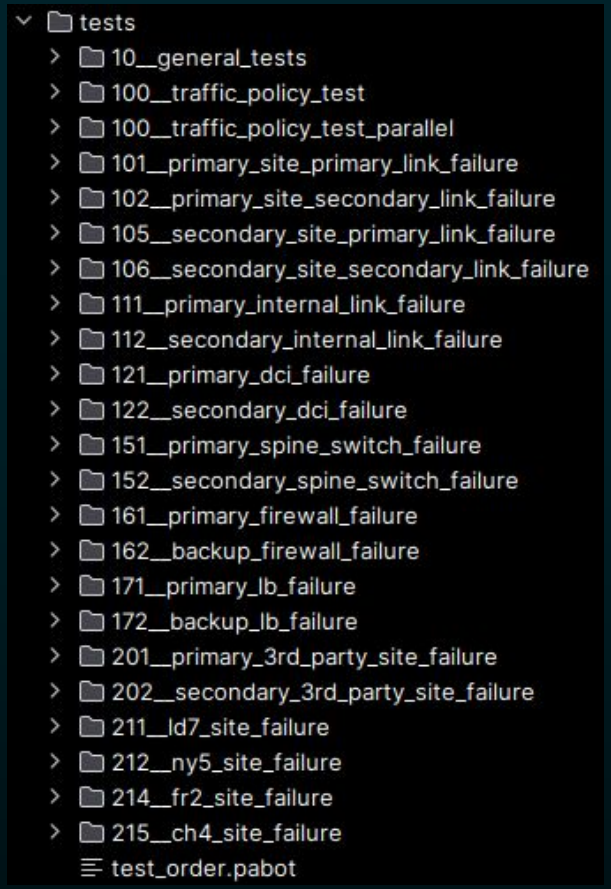
TransFICC

Agenda

TransFICC

- Introduction: The story
- Who are we?
- The Problem: High-stakes network changes in finance (risks & challenges)
- Solution: Zero-regression philosophy and CI/CD pipeline design
- Implementation: Key components
  - Virtual lab
  - Test automation
  - Failure simulations
  - Branching strategy
  - Zero-touch deployment
- Case Study: Real-world example of the pipeline preventing an incident
- Results, lessons, and Key takeaways: Impact achieved and insights gained

The end goal



- Achieve zero regressions by running full network tests upon every change
- Including exercising every redundancy mechanism by simulating failures of every component.

```
27250 tests, 25846 passed, 0 failed, 1404 skipped.
================================================
Output:  /home/ciagent/workspace/infra-netenv-acceptance-test/
Log:     /home/ciagent/workspace/infra-netenv-acceptance-test/
Report:  /home/ciagent/workspace/infra-netenv-acceptance-test/
```

Who are we?

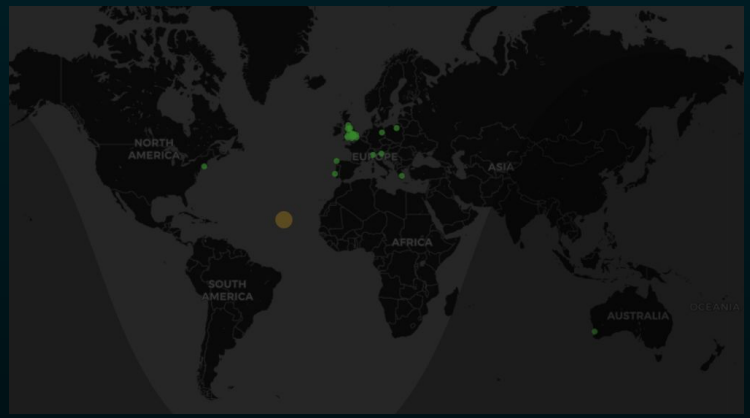
TransFICC

London based Fintech that facilitates Fixed Income trading to multiple venues by providing simplified connectivity and APIs.

Our customers are international financial institutions.
We provide the connectivity, the platform, and software.

Remote-first, with employees based in the UK, Europe, USA, and Australia.

We're hiring! https://transficc.com/join



The Problem

TransFICC

FINANCIAL NETWORKS = EXTREME RELIABILITY NEEDS

Near zero downtime

RISK OF MANUAL CHANGES

Error prone → outages

SLOW, CAUTIOUS RELEASES

Fear, infrequent changes, reviewboards → $$ costly

NEED FOR CHANGE

And grow fast without compromise

The Goal: Zero-regression changes

TransFICC

What is Zero-Regression? No change should break existing functionality, ever.

Shift in Mindset: Network Engineering as part of SRE.

TransFICC SRE covers all infrastructure, devops, and automation.

Network configuration, code and playbooks are a 1st class citizen

Network CI/CD: Foundation

TransFICC

Network configuration as code.
New features → code change.

A mix of: static non-mutable files.
YAML + Jinja2 templates + Python
→ Ansible / Terraform

These changes are tested: syntax + new features
+ regression testing

TDD: create test 1st, implement feature
→ test pass on completion

Network CI/CD: Foundation

TransFICC

Test all branches every day in virtual environments.

CI/CD pipeline → spins up labs and runs tests.

Emulate the _entire network + critical services.

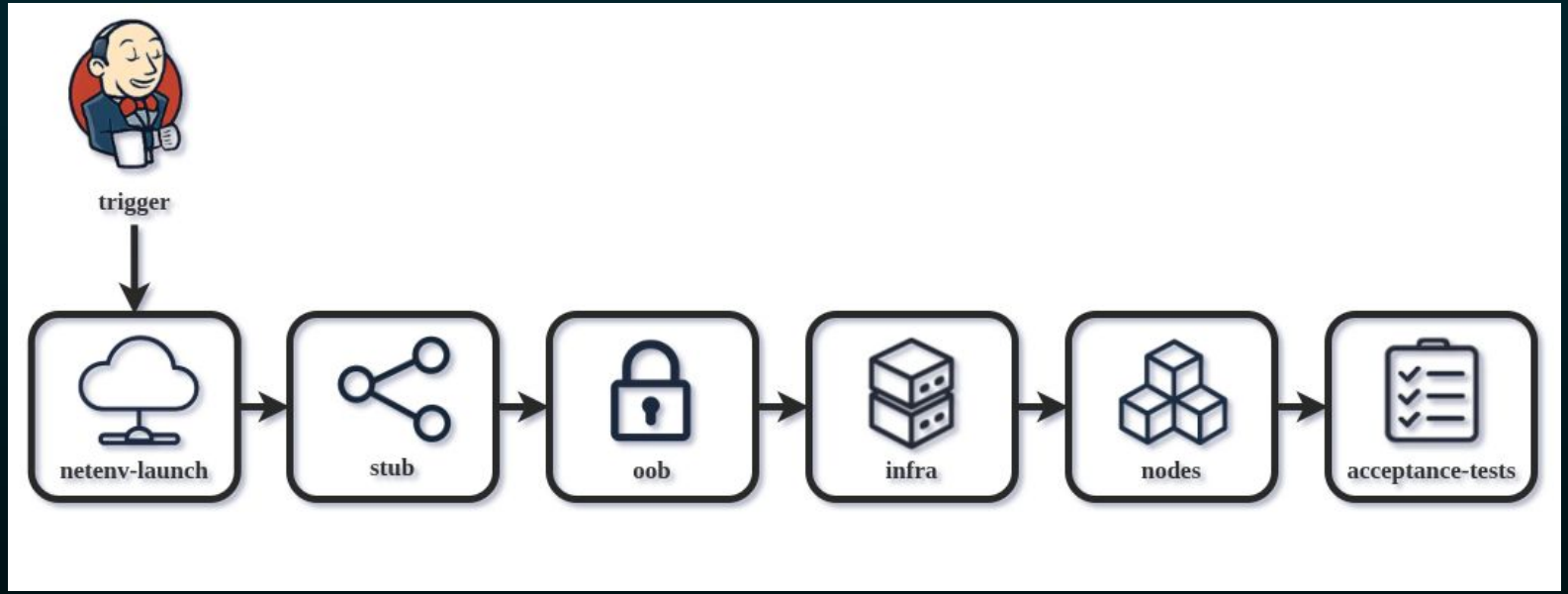
Exact mirror of production.

Deployed from scratch + from existing network configuration

Include all monitoring and alerting stack.

Network CI/CD: Foundation

TransFICC



Network CI/CD: Foundation

TransFICC

Any misconfiguration is caught immediately

No manual reviews

Lots of tests (27K), not only for the happy path but also introduce failures

Safe place for evaluating upgrades and new features or entirely new technologies

Prior incidents are coded as tests

Only commits that are green are ever pushed to production network

Network CI/CD: Automated testing

TransFICC

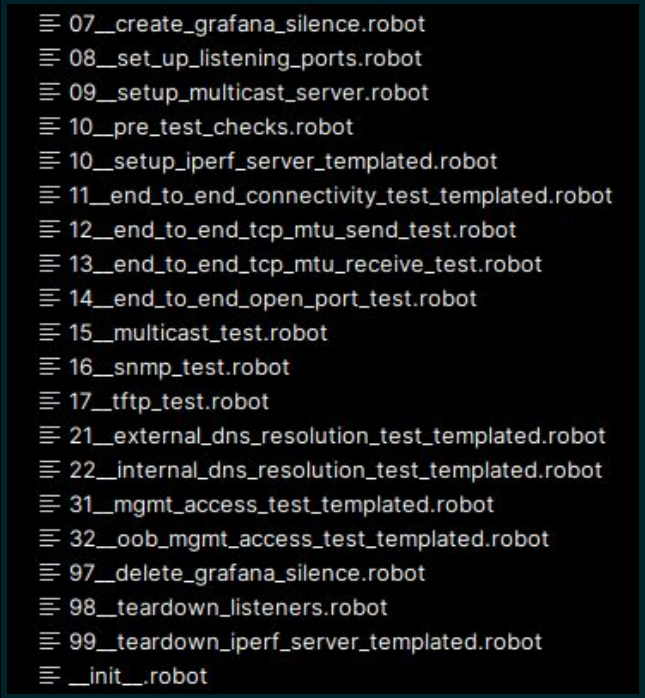
- Test defined using high level expressions/connectivity policies
- CSV based templated tests

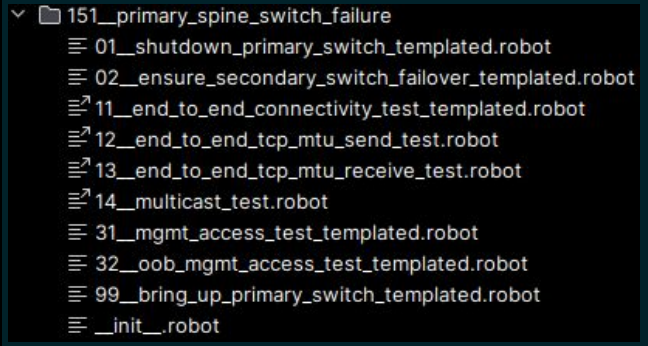
```
- from_tagged_as:
      - source_tagA
      - name: "source_network_name"
    to_tagged_as:
      - destination_tagB
      - name:
"destination_network_name"
```

OR

Network CI/CD: Automated testing

TransFICC





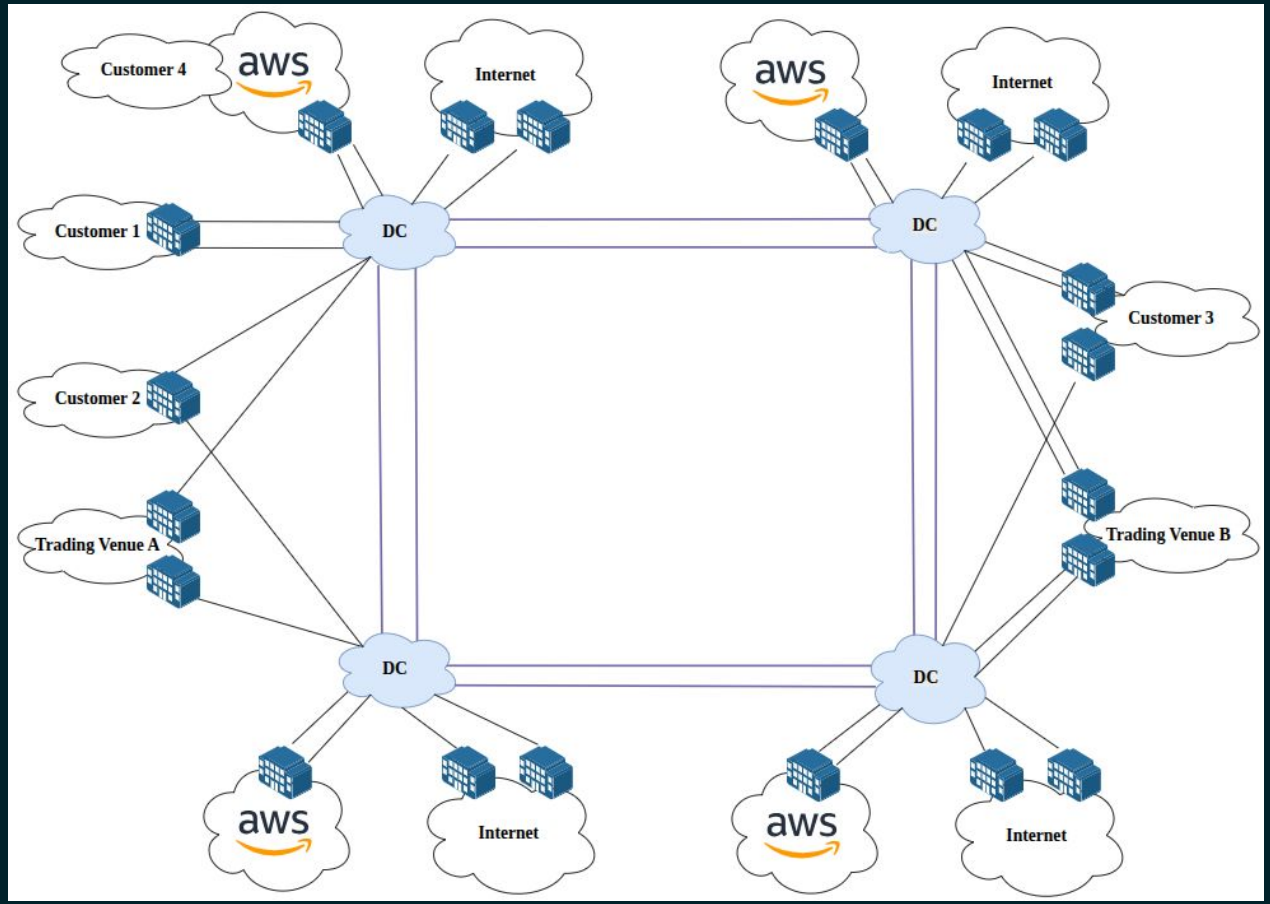
Network CI/CD: Automated testing

./tf netenv-acceptance-test --include primary_site_primary_link_failure

```
2025-10-05 17:49:38.926088 [PID:2869538] [21] [ID:1523] PASSED Tests.Primary Site Primary Link Failure.Post Test Ch
2025-10-05 17:49:38.996787 [PID:2869500] [75] [ID:1524] PASSED Tests.Primary Site Primary Link Failure.Post Test Ch
2025-10-05 17:49:39.181284 [PID:2869493] [14] [ID:1520] PASSED Tests.Primary Site Primary Link Failure.Post Test Ch
2025-10-05 17:49:39.236201 [PID:2869799] [68] [ID:13] EXECUTING Tests.General Tests.Teardown Listeners.Teardown nc
2025-10-05 17:49:39.236238 [PID:2869800] [36] [ID:14] EXECUTING Tests.General Tests.Teardown Iperf Server Templated
2025-10-05 17:49:39.236513 [PID:2869802] [53] [ID:15] EXECUTING Tests.General Tests.Delete Grafana Silence
2025-10-05 17:49:42.169181 [PID:2869851] [5] [ID:1559] EXECUTING Tests.General Tests.Teardown Listeners
2025-10-05 17:49:42.169834 [PID:2869853] [32] [ID:1564] EXECUTING Tests.General Tests.Teardown Iperf Server Templat
2025-10-05 17:49:42.169311 [PID:2869855] [62] [ID:1562] EXECUTING Tests.General Tests.Teardown Iperf Server Templat
2025-10-05 17:49:46.278607 [PID:2869851] [5] [ID:1559] PASSED Tests.General Tests.Teardown Listeners in 4.1 seconds
2025-10-05 17:49:46.279432 [PID:2869855] [62] [ID:1562] PASSED Tests.General Tests.Teardown Iperf Server Templated
2025-10-05 17:49:46.376073 [PID:2869840] [73] [ID:1563] PASSED Tests.General Tests.Teardown Iperf Server Templated
2025-10-05 17:49:46.377316 [PID:2869841] [41] [ID:1558] PASSED Tests.General Tests.Teardown Listeners in 4.2 second
2025-10-05 17:49:46.378125 [PID:2869845] [50] [ID:1557] PASSED Tests.General Tests.Teardown Listeners in 4.2 second
2025-10-05 17:49:46.378945 [PID:2869847] [25] [ID:1560] PASSED Tests.General Tests.Teardown Listeners in 4.2 second
2025-10-05 17:49:46.379731 [PID:2869853] [32] [ID:1564] PASSED Tests.General Tests.Teardown Iperf Server Templated
2025-10-05 17:49:46.477048 [PID:2869843] [27] [ID:1561] PASSED Tests.General Tests.Teardown Iperf Server Templated
2025-10-05 17:49:46.578221 [PID:2869849] [1] [ID:1556] PASSED Tests.General Tests.Teardown Listeners in 4.4 seconds
1580 tests, 1525 passed, 0 failed, 55 skipped.
==============================================
Output:  test_results/output.xml
Log:     test_results/log.html
Report:  test_results/report.html
Stopping PabotLib process
Robot Framework remote server at 127.0.0.1:8270 stopped.
PabotLib process stopped
Total testing: 4 hours 1 minute 0.10 seconds
Elapsed time:  5 minutes 13.94 seconds
```

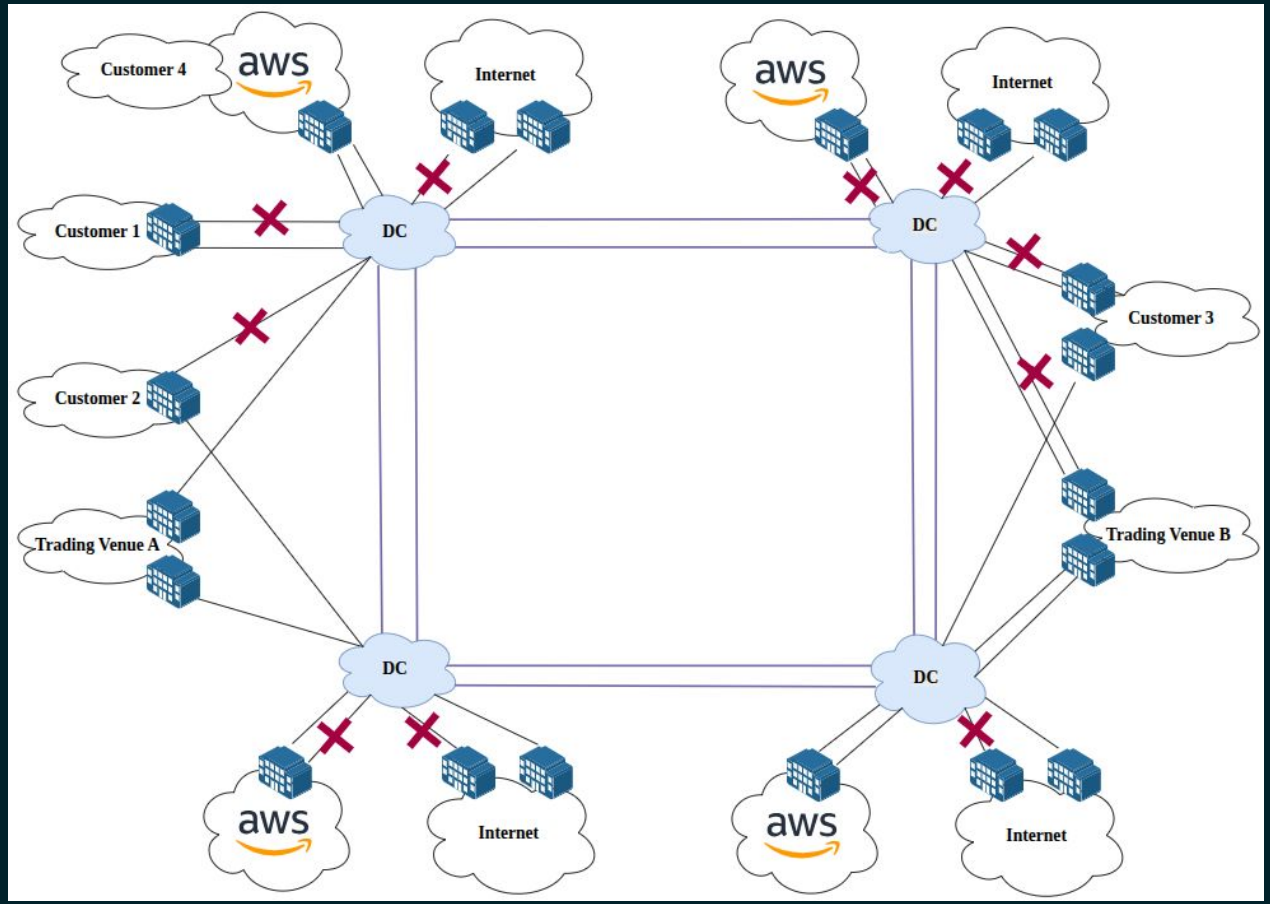
Network CI/CD: Automated testing

TransFICC



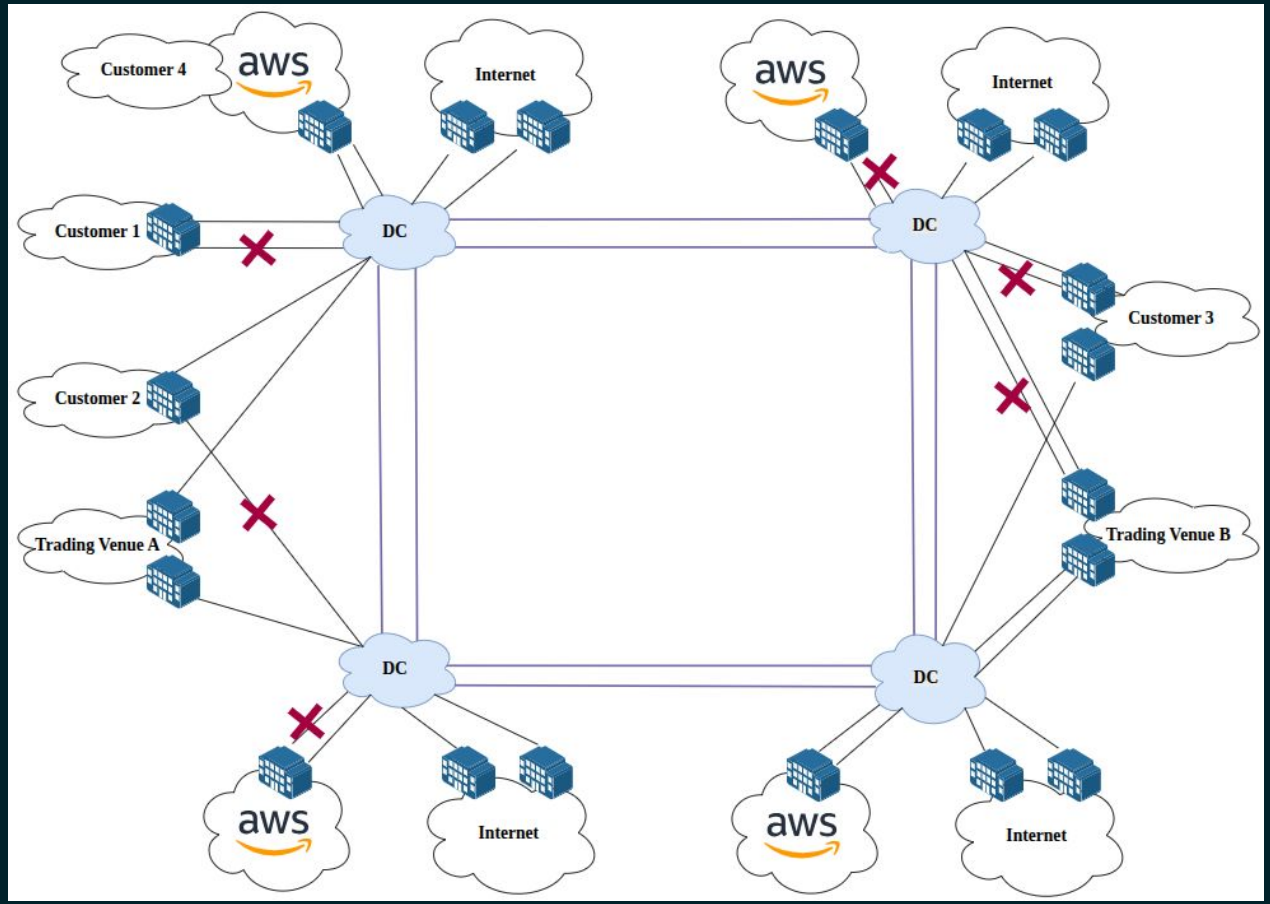
- Links are grouped based on tags
- Tags:
  - function
  - site
  - sequence
  - left_dc
  - right_dc

Network CI/CD: Automated testing



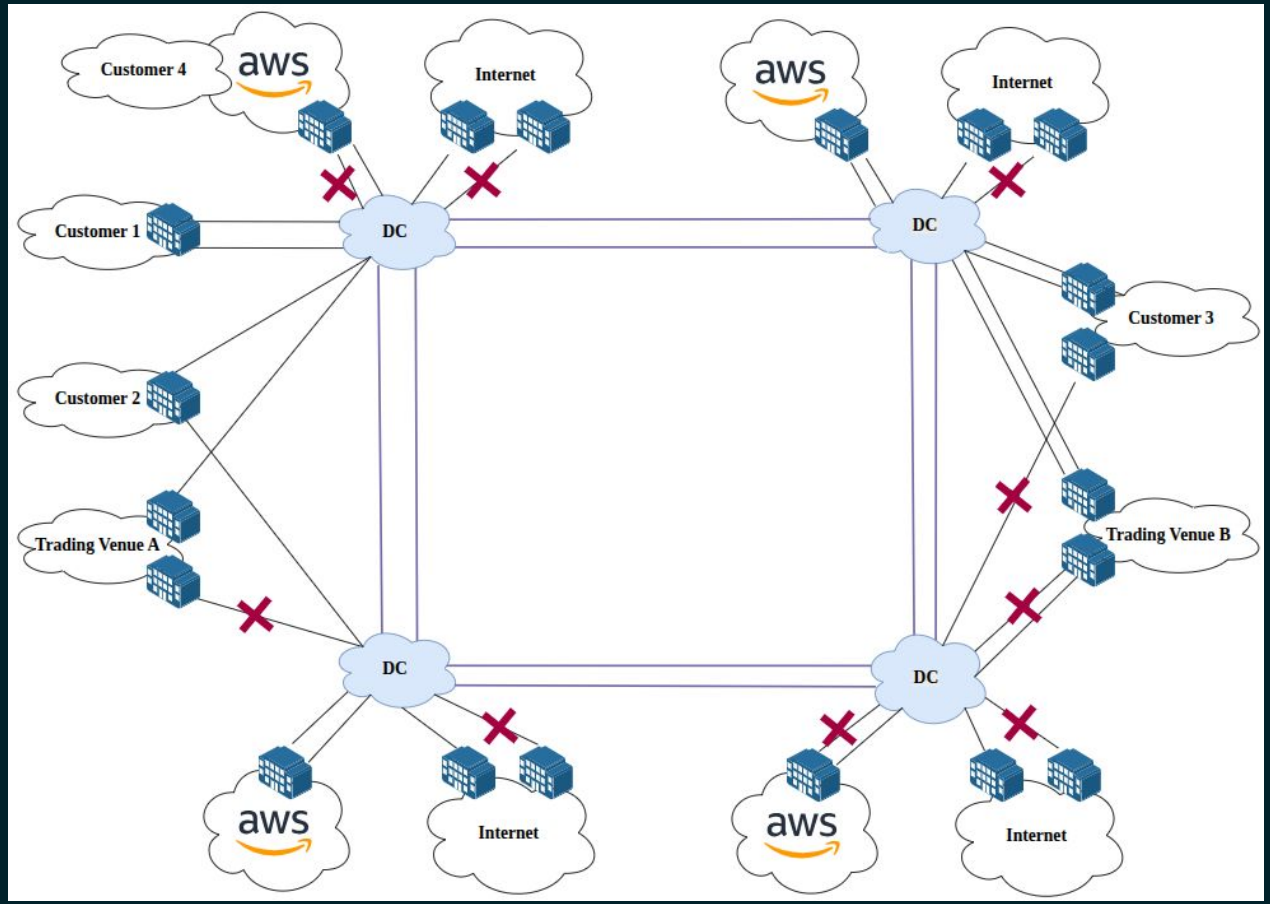
- Tags are used to shutdown links of the same function - allows to scale
- function=3rd_party_handoff
- site=primary
- sequence=primary

Network CI/CD: Automated testing



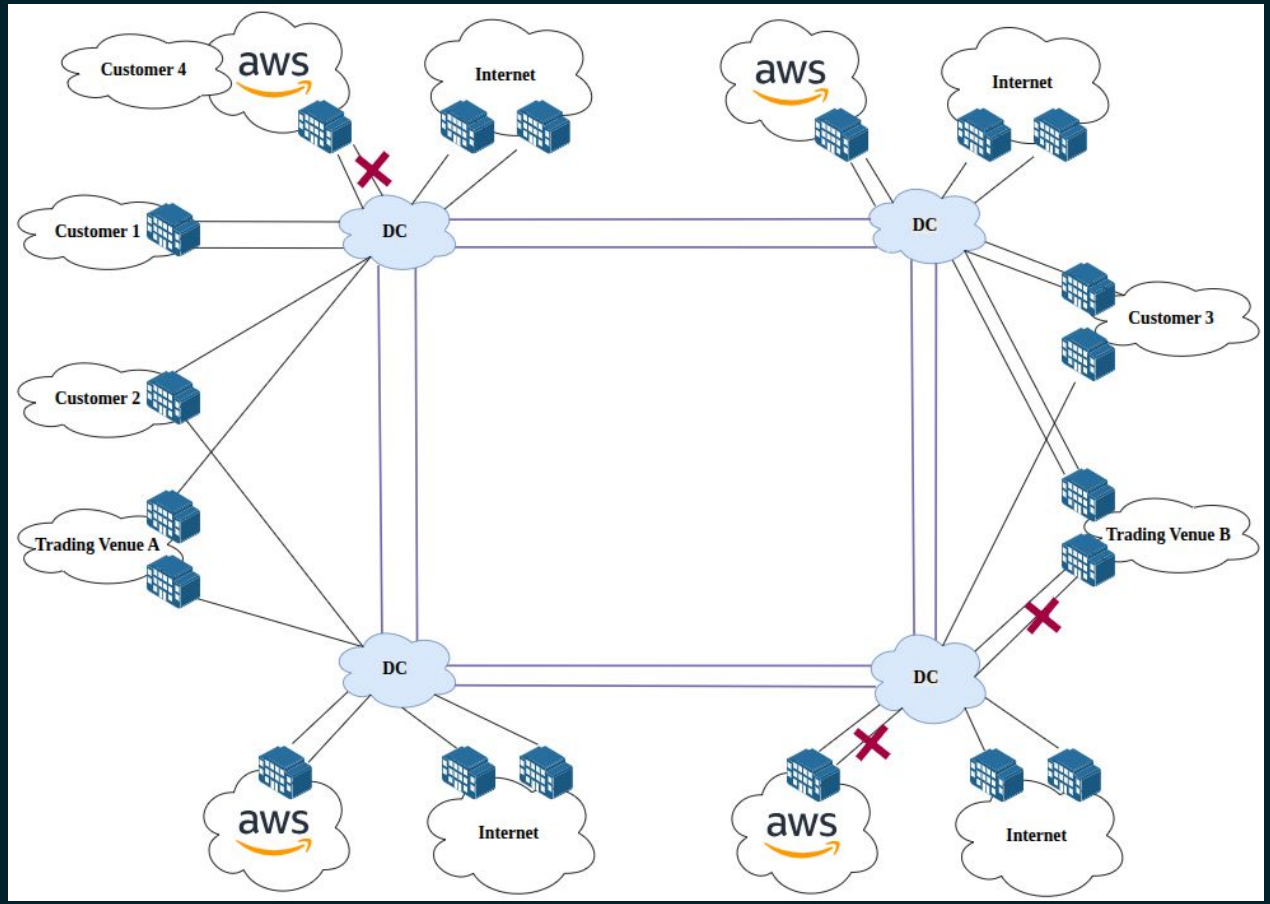
- function=3rd_party_handoff
- site=primary
- sequence=secondary

Network CI/CD: Automated testing



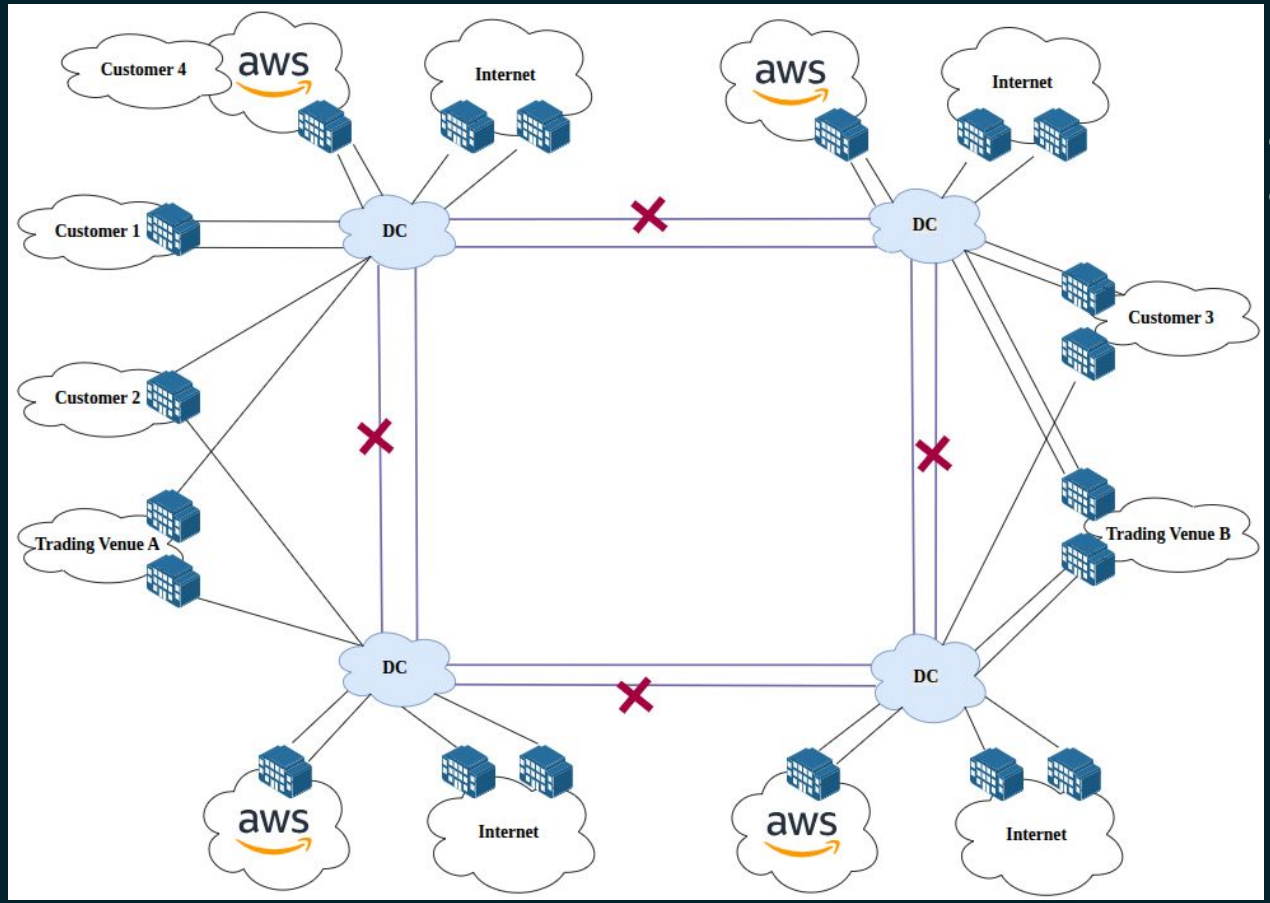
- function=3rd_party_handoff
- site=secondary
- sequence=primary

Network CI/CD: Automated testing



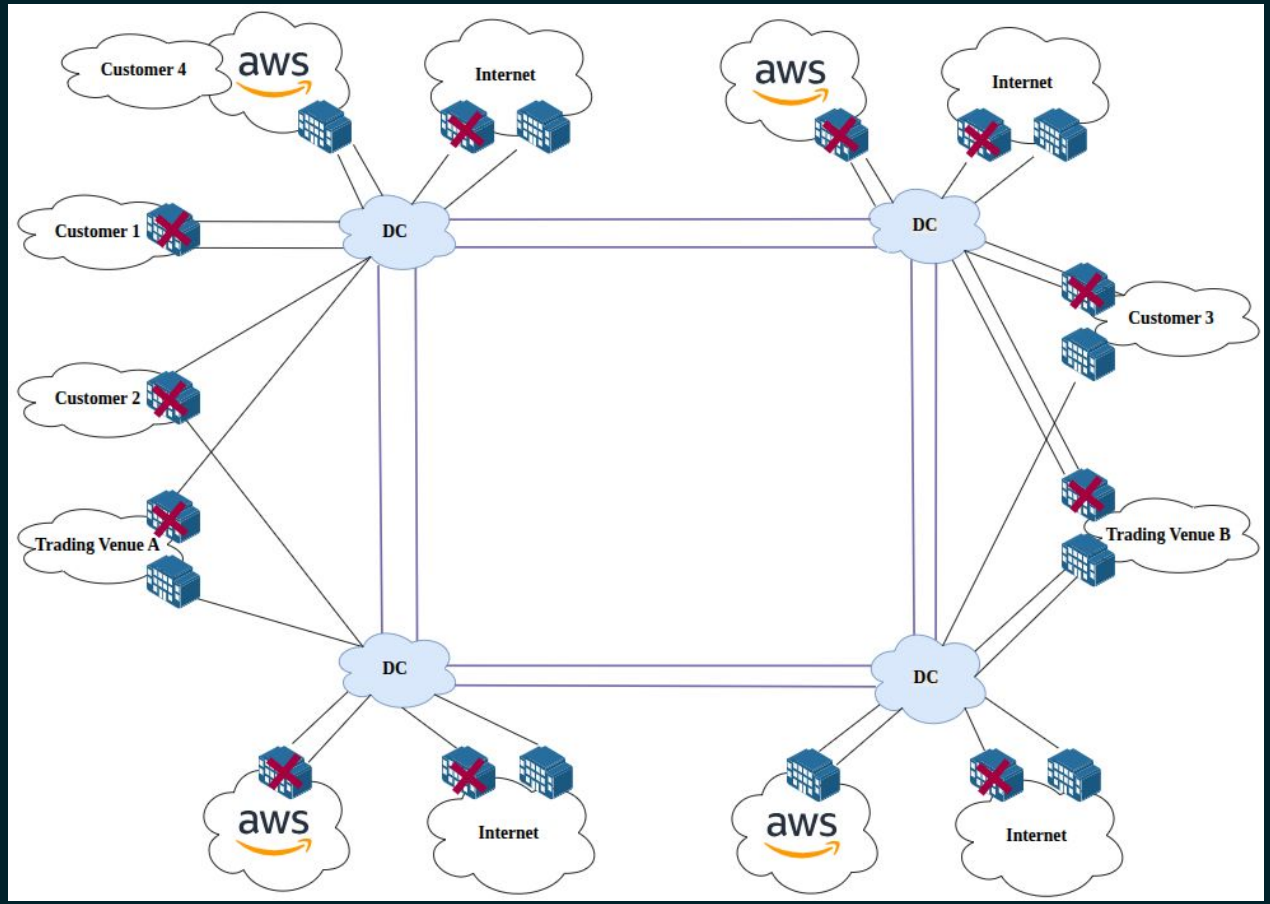
- function=3rd_party_handoff
- site=secondary
- sequence=secondary

Network CI/CD: Automated testing



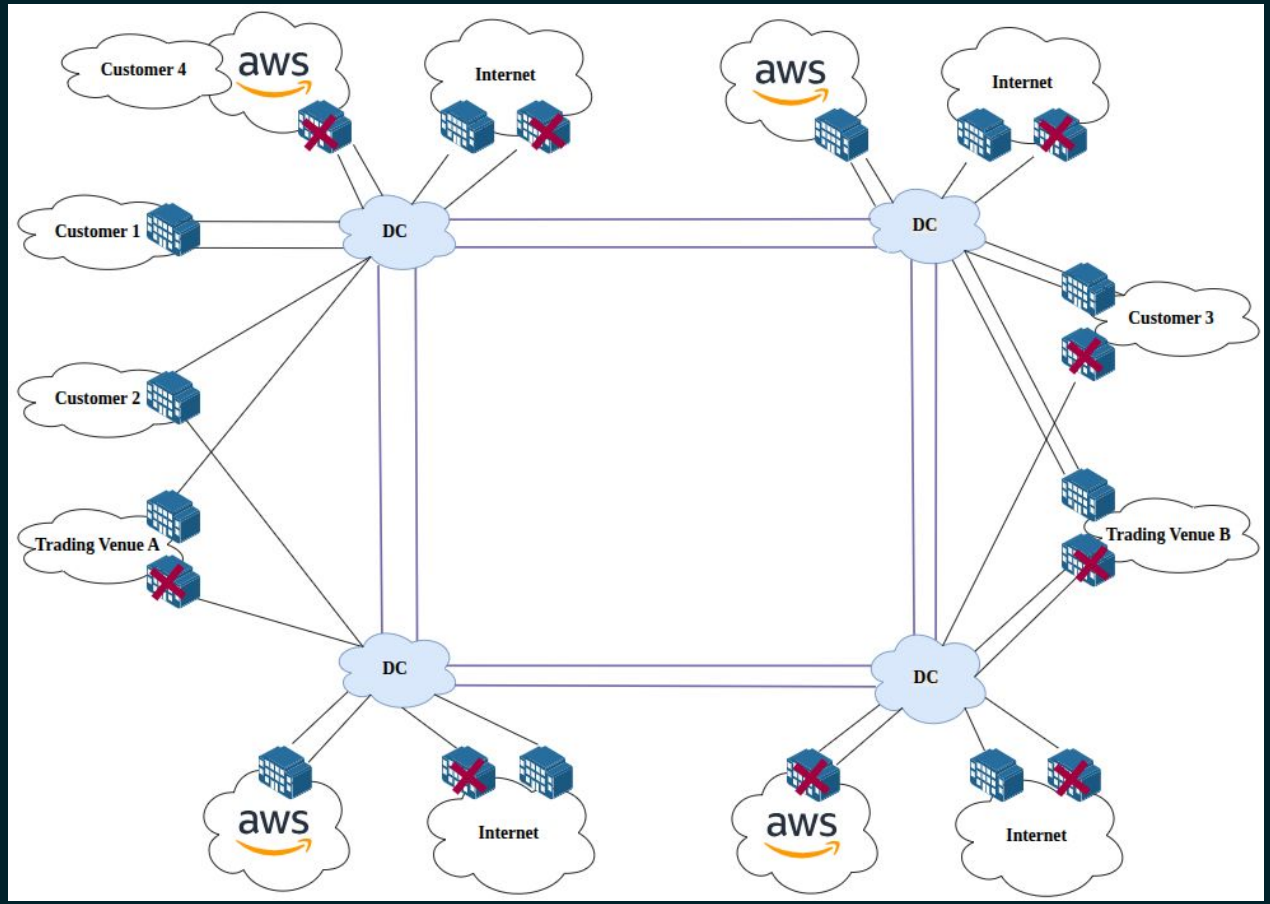
- function=dci/internal
- sequence=primary/secondary

Network CI/CD: Automated testing



- function=3rd_party_handoff
- site=primary

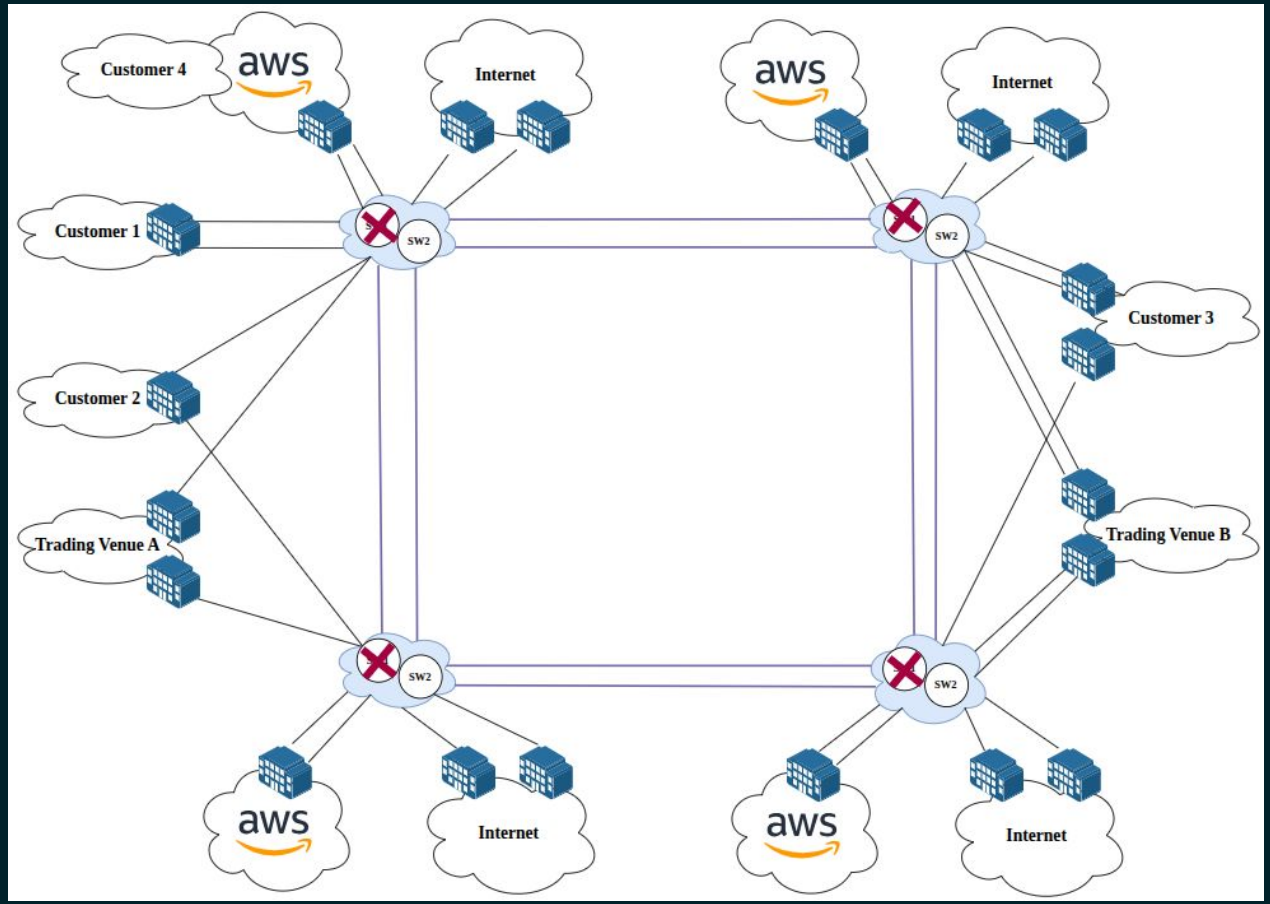
Network CI/CD: Automated testing



- function=3rd_party_handoff
- site=secondary

Network CI/CD: Automated testing

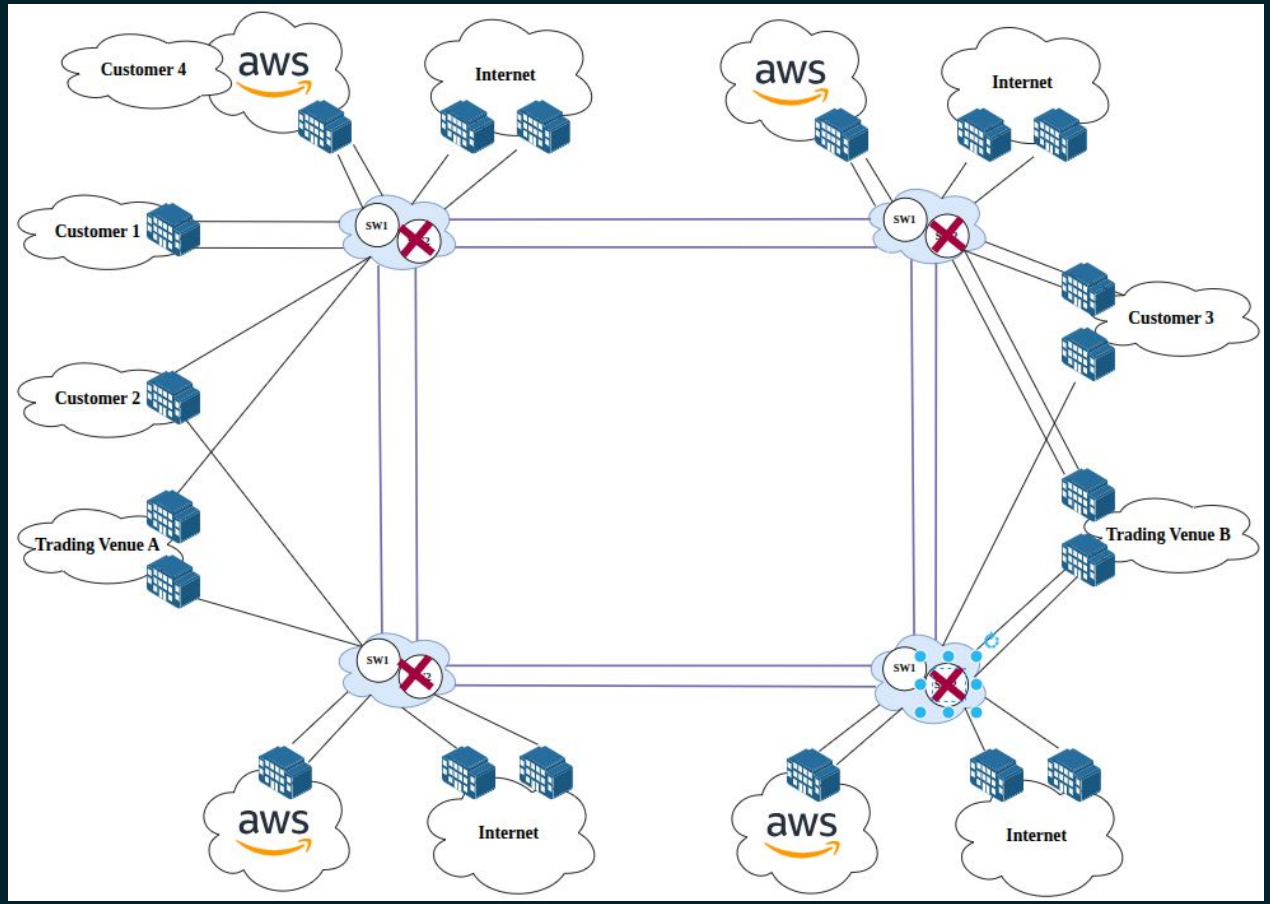
TransFICC



- role=spine_switch
- spine_switch_role=primary
- Same for FWs and LBs

Network CI/CD: Automated testing

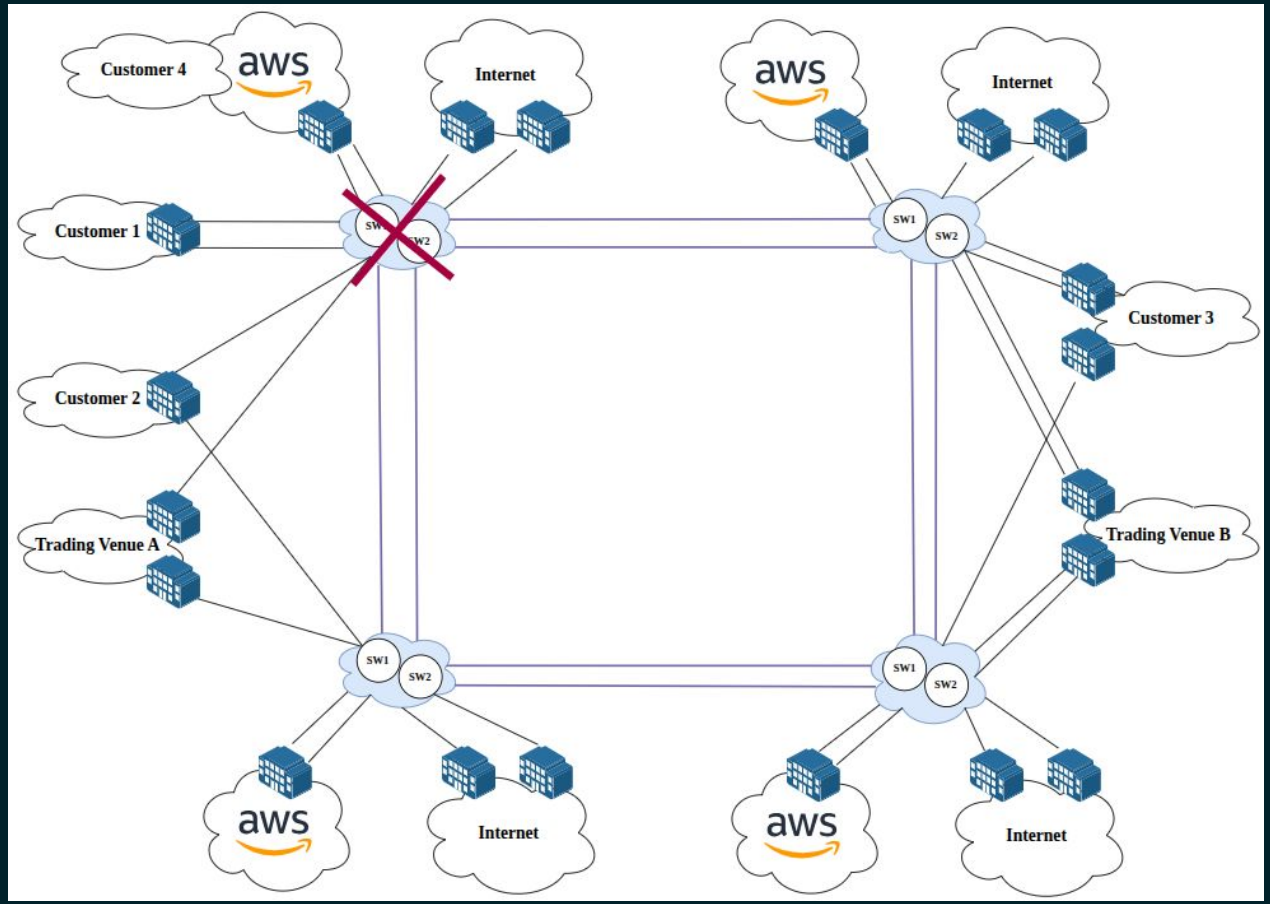
TransFICC



- role=spine_switch
- spine_switch_role=secondary
- Same for FWs and LBs

Network CI/CD: Automated testing

TransFICC

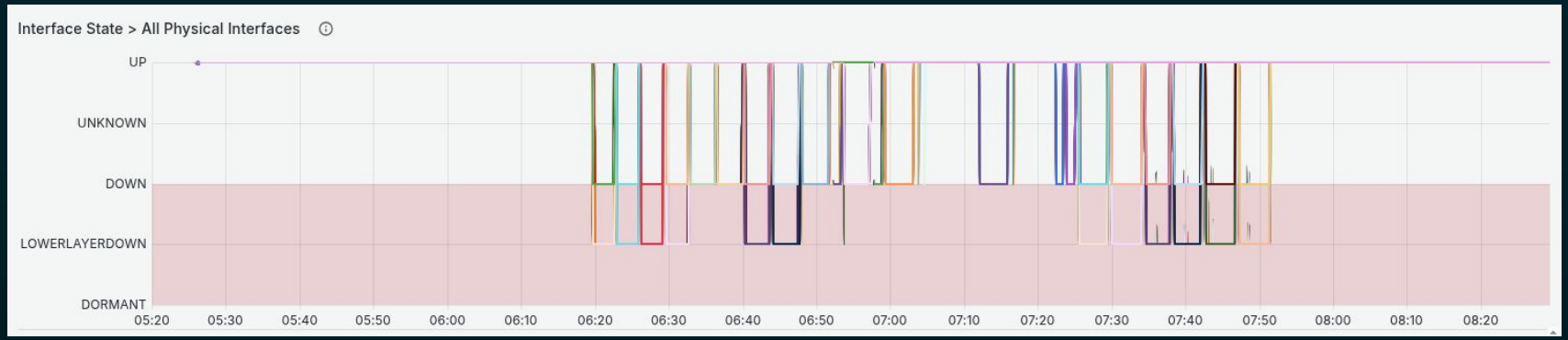


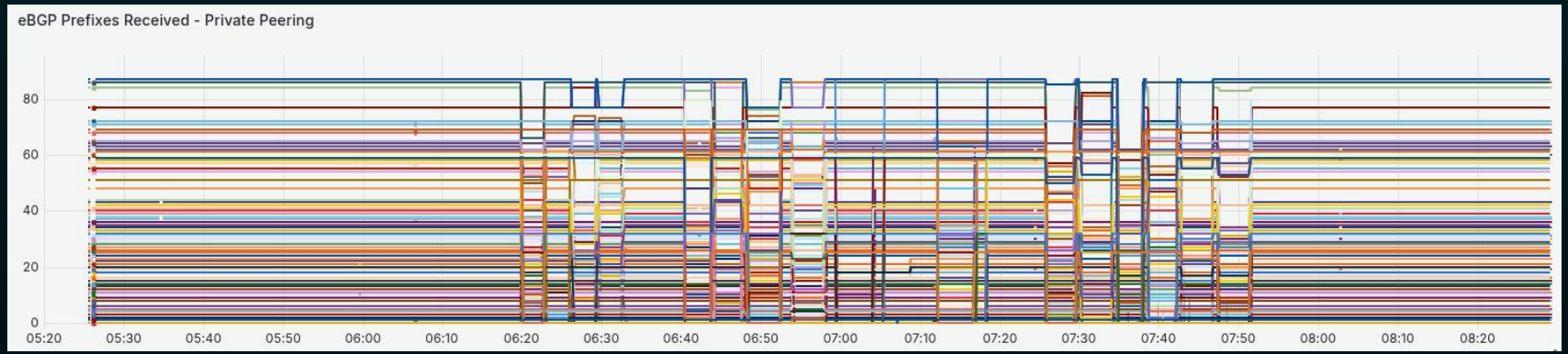
- left_dc or right_dc =dcA

Network CI/CD: Automated testing

Monitoring in the lab: Interface failures -> BGP failures

TransFICC





Known Weak Spots & Gaps

TransFICC

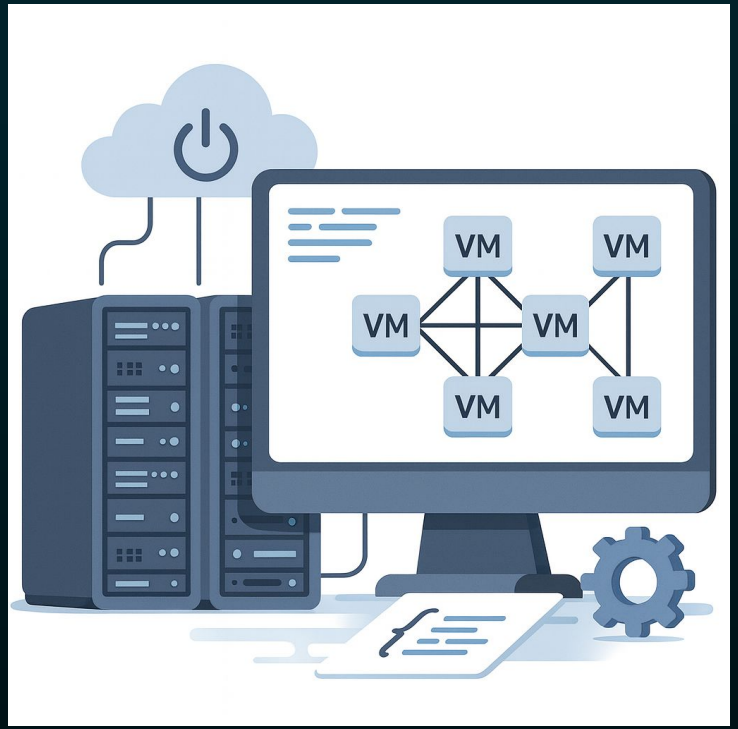
- YAML file for test exceptions
- Acts as a “known exceptions register”
- Highlights vulnerable areas and lack of redundancy
- Makes risk visible across teams (Engineering + Business)
- Drives remediation prioritisation and accountability

```yaml
skipped_tests:
  tests: <80 items>
- suite_name: "Tests.Primary Dci Failure.Mgmt Access Test Templated"
  tests: <2 items>
- suite_name: "Tests.Secondary Dci Failure.End To End Connectivity Test Templated"
  tests: []
- suite_name: "Tests.Primary Firewall Failure.End To End Connectivity Test Templated"
  tests: []
- suite_name: "Tests.Backup Firewall Failure.End To End Connectivity Test Templated"
  tests: []
- suite_name: "Tests.Primary 3Rd Party Site Failure.End To End Connectivity Test Templated"
  tests: <30 items>
- suite_name: "Tests.Primary 3Rd Party Site Failure.Oob Mgmt Access Test Templated"
  tests: <24 items>
- suite_name: "Tests.Secondary 3Rd Party Site Failure.End To End Connectivity Test Templated"
  tests: []
- suite_name: "Tests.Ld7 Site Failure.End To End Connectivity Test Templated"
  tests: <51 items>
- suite_name: "Tests.Ld7 Site Failure.Multicast Test"
  tests: <2 items>
- suite_name: "Tests.Ny5 Site Failure.End To End Connectivity Test Templated"
  tests: []
- suite_name: "Tests.Ny5 Site Failure.Mgmt Access Test Templated"
  tests: []
- suite_name: "Tests.Fr2 Site Failure.End To End Connectivity Test Templated"
  tests: []
- suite_name: "Tests.Ch4 Site Failure.End To End Connectivity Test Templated"
  tests: []
```

How we achieved that - NetEnv!

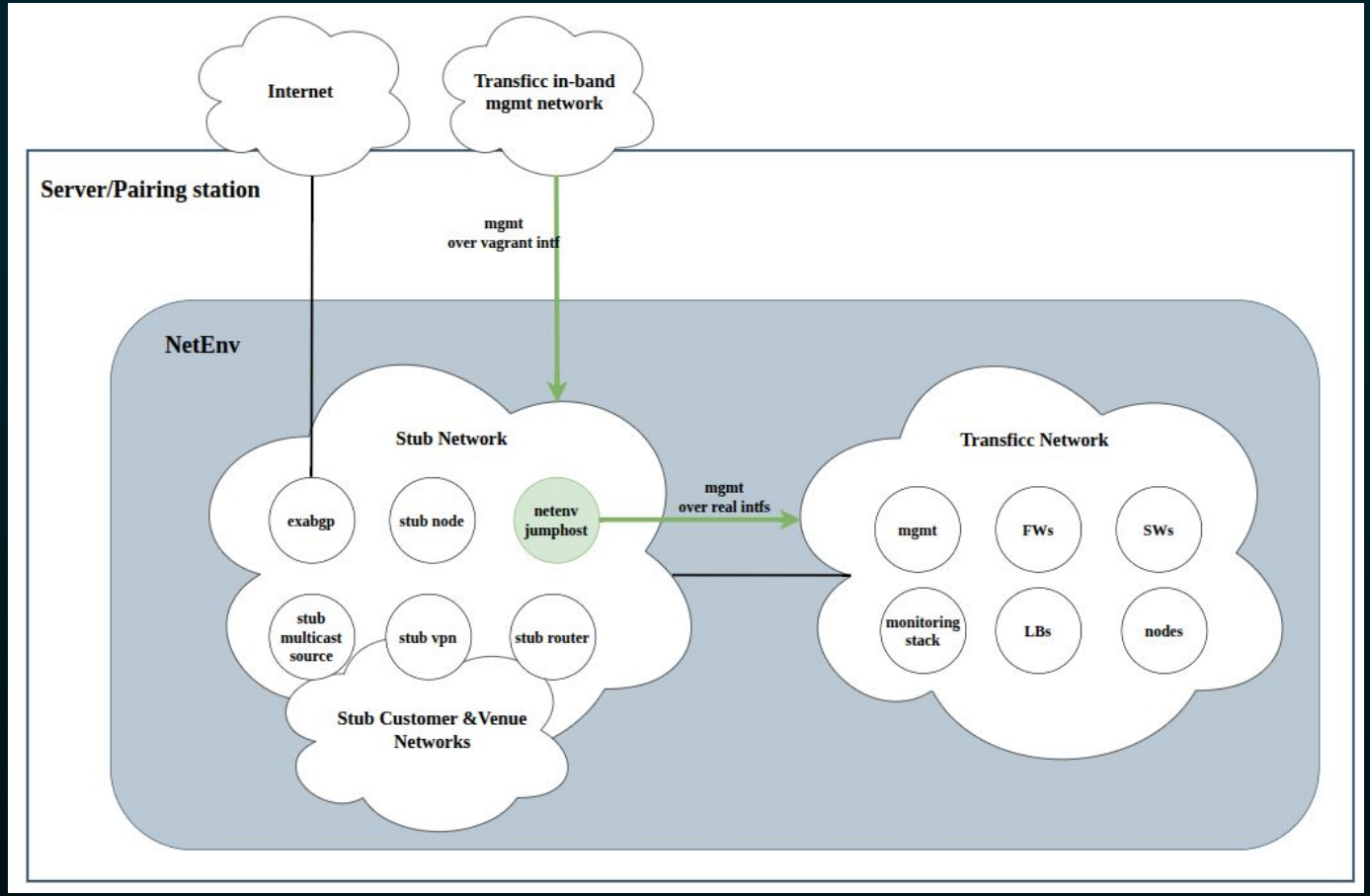
TransFICC

- Netenv our virtual environment - mirror of production



How we achieved that - NetEnv!

TransFICC



NetEnv Topology

TransFICC

- DOT (graph description language)
- Netenv graph for defining VMs
- DC graphs
- Converted to Vagrant file and run with libvirtd (netenv-initialise stage)

```
graph netenv {
  "switch-1" [
    ports=52 memory="1024"
    os="transficc"
    ssh_insert_key=false
    config="./helper_scripts/extra_switch_config.sh"
    graphics="vnc"]
  "node-1" [
    memory="1024"
    os="almalinux/9"
    graphics="vnc"
    config="./helper_scripts/extra_server_config_netenv.sh"]
```

```
graph ld7 {
  hostnametype="fqdn"
  LLDP="match_hostname=fqdn"
  "switch-ld7-1":"swp35" -- "internet-router-ld4-1":"0/0/17" [
    netenv_right_device_override="router-netenv-1.stub.transficc.net"
    netenv_right_device_intf_override="swp105"
    function="3rd_party_handoff" site="primary" sequence="primary"
    left_dc="ld7" right_dc="ld4"]
```

NetEnv Pipeline

TransFICC

- Jenkins pipeline
- Ansible + custom inventory plugins, one for stub and one for netenv jumphost

```
pipeline {
  agent { label "$TRANSFICC_NETENV_HYPERVISOR_HOST" }
  environment {...}
  stages {
    stage('Update name and description for this build') {...}
    stage('Clean and reinitialise the libvirt environment') {...}
    stage('Setup stub hosts') {...}
    stage('Setup wireguard server') {...}
    stage('Setup oob network, internal and archive hosts') {...}
    stage('Setup infra network') {...}
    stage('Setup the rest of the environment') {...}
    stage('Run netenv post-launch extras') {...}
    stage('Run acceptance tests') {...}
  }
  post {...}
}
```

```
function netenv-stub() {
  # 1. setup all stub hosts including the netenv jump host
  time netenv-setup -l '*.stub.*,!pairing*'
}
```

```
function netenv-infra-switches() {
  # wait for infra switches to be reachable
  (
    for _ in {1..5}; do...done && exit "$ping_return_code"
  ) &&
  # 6. setup infra network: 1. setup wireguard on infra switches
  (time netenv-jumphost-run netenv-wg-setup -l '*switch*.infra.*') &&
  # 7. setup infra network: 2. setup infra switches
  (time netenv-jumphost-run netenv-setup -l '*switch*.infra.*')
}
```

NetEnv: Our Network CI/CD

TransFICC

PATCH MANAGER
infrastructure management software

Topology converter

Jenkins

Vagrant
/libvirtd/qemu

Ansible

BASH

tc

ROBOT
FRAME
WORK/

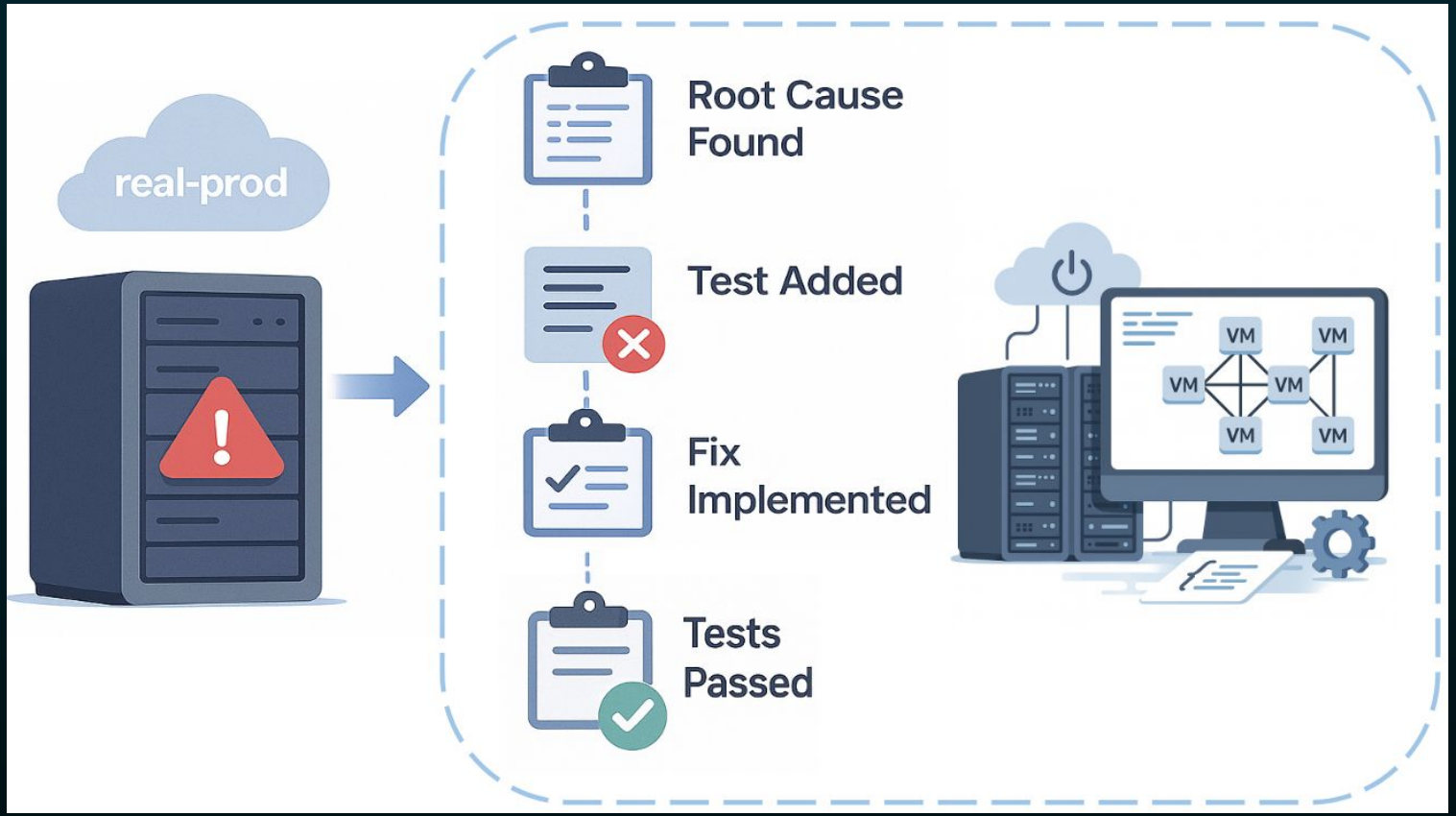
git

NetEnv: Our Network CI/CD

TransFICC

Real-world example

TransFICC



Real-world example

TransFICC

```
*** Settings ***
Resource            ../../lib/robot.resource
Library             DataDriver    file=../../lib/end_to_end_tcp_mtu_test.csv    reader_class=csv_reader    encoding=utf_8    optimize_pabot=Atomic
Variables           ../../opt/device_inventory.yml

Suite Teardown      Close All Connections
Test Template       Templated Keyword

Force Tags          mtu


*** Test Cases ***
Testing TCP and send MTU from IP: ${src_ip_address}, tags: ${src_tags} --> IP: ${dst_ip_address}, tags: ${dst_tags}


*** Keywords ***
Templated Keyword
    [Arguments]    ${hostname}    ${src_ip_address}    ${src_tags}    ${dst_ip_address}    ${dst_tags}    ${vrf}    ${skipped_test_suites}
    Suite Setup Hooks    ${SUITE NAME}    ${TEST NAME}    ${skipped_test_suites}
    Open Connection And Log In    ${inventory}[${hostname}][ip_address]
    Wait Until Keyword Succeeds
    ...    3 times
    ...    5 sec
    ...    Test TCP Connection and MTU From Source To Destination And Verify Return Code
    ...    dst_ip_address=${dst_ip_address}
    ...    port=5001
    ...    src_ip_address=${src_ip_address}
    ...    vrf=${vrf}
    ...    data_flow_direction=out
    ...    extra_args=-M 1460 -b 1M --connect-timeout 3000 -t 3 --snd-timeout 2000 --rcv-timeout 2000
```

Real-world example

TransFICC

SUITE **Shutdown Primary Link Templated**

SUITE **Multicast Test**

SUITE **End To End Tcp Mtu Send Test**

Full Name: Tests.Primary Site Primary Link Failure.End To End Tcp Mtu Send Test
DataDriver: Tests.Primary Site Primary Link Failure.End To End Tcp Mtu Send Test.Testing TCP and send MTU from IP tags: api_app on_prem,or
Source: /home/ciagent/workspace/infra-netenv-acceptance-test/python/netenv-acceptance-test/src/netenv_acceptance_test/network/datacentres/tests/101__primar
Start / End / Elapsed: 20250915 08:26:43.392 / 20250915 08:26:56.735 / 00:00:13.343
Status: 7 tests total, 7 passed, 0 failed, 0 skipped

TEARDOWN SSHLibrary.**Close All Connections**

TEST **Testing TCP and send MTU from IP: tags: api_app, ,on_prem,origin_dc:ld7,prod,public_ip --> IP tags: leased_line**

TEST **Testing TCP and send MTU from IP: , tags: api_app, n_prem,origin_dc:ld7,prod,public_ip --> IF tags ,prod**

TEST **Testing TCP and send MTU from IP: 1 , tags: api_app ,on_prem,origin_dc:ny5,prod,public_ip -->**

Full Name: Tests.Primary Site Primary Link Failure.End To End Tcp Mtu Send Test.Testing TCP and send MTU from IP: s: api_app i,on_pre
Tags: external_link_failure, link_failure, mtu, primary_external_link_failure, primary_site_primary_link_failure
Start / End / Elapsed: 20250915 08:26:46.783 / 20250915 08:26:51.211 / 00:00:04.428
Status: PASS
Message: Passed

KEYWORD **Templated Keyword** hostname=node-netenv-1.stub.transficc.net src ip address= src tags=api app i,on prem,origin dc:ny5,prod,public vrf=

Start / End / Elapsed: 20250915 08:26:46.784 / 20250915 08:26:51.211 / 00:00:04.427

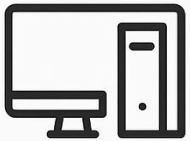
KEYWORD robot.**Suite Setup Hooks** ${SUITE NAME} ${TEST NAME} ${skipped_test_suites}

KEYWORD robot.**Open Connection And Log In** ${inventory}[${hostname}][ip_address]

KEYWORD BuiltIn.**Wait Until Keyword Succeeds** 10 times 5 sec Test TCP Connection and MTU From Source To Destination And Verify Return Code dst_ip_address= data_flow_direction=out extra_args=-M 1460 -b 1M --connect-timeout 3000 -t 3 --snd-timeout 2000 --rcv-timeout 2000

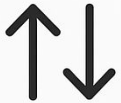
Challenges

TransFICC



Network is growing fast: Laptop simulation → large servers

Memory optimization through careful allocation + stubbing



Vertical vs. horizontal scaling strategies

Partial topologies for faster dev feedback loops

Testbed compatibility affects vendor selection

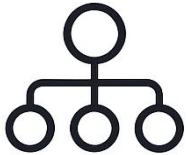
Small differences between virtual and HW: e.g. FPGAs, ASIC ACLs

Our roadmap

TransFICC

Running tests directly in production.
Compare connectivity from edge through network before and after push.

Expand horizontal scaling:
Spin up DC/Zone/Continent partial networks per server



Patchmanager synchronization
with topology DOT files

Results and Takeaways

TransFICC

High confidence in every change

Less stress and anxiety

Velocity without bureaucracy. Netenv:
Evidence of working changes

No more manual late-night pushes

Rapid rebuilds at the device or PoP level

A platform for verifying new network tech or ideas.

Vendor selection: support for virtualisation
and testability

Q/A

TransFICC

- Questions?

Thank you!!

TransFICC