



The bridge to possible

# Testing and Deployment of NSO Use Cases with a CI/CD State of Mind

## Embracing NetDevOps as a life mantra

Alfonso Sandoval Rosas, Software Consulting Engineer

 @ponchotitlan

# 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.



# But first, a horror story ...



- “This custom NSO package for provisioning services is great! In fact, we want to add one more feature ...”

Network Operations team

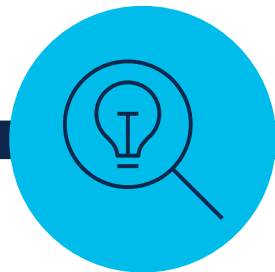


- “Absolutely. Let’s discuss the capabilities, estimate the effort, and so on”

Software Development team



Details were agreed



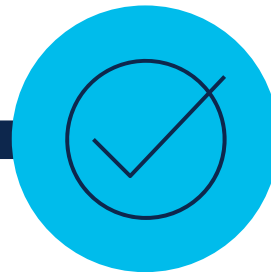
User Story US00012:  
“Adding support for  
interface provisioning”  
was created



Hands-on  
coding took  
place



Code was  
redeployed in production



New code was  
tested in the lab

Ticket FW001:  
“Syntax error  
when  
provisioning  
access lists”



– “There are very strange errors after  
last night deployment!”

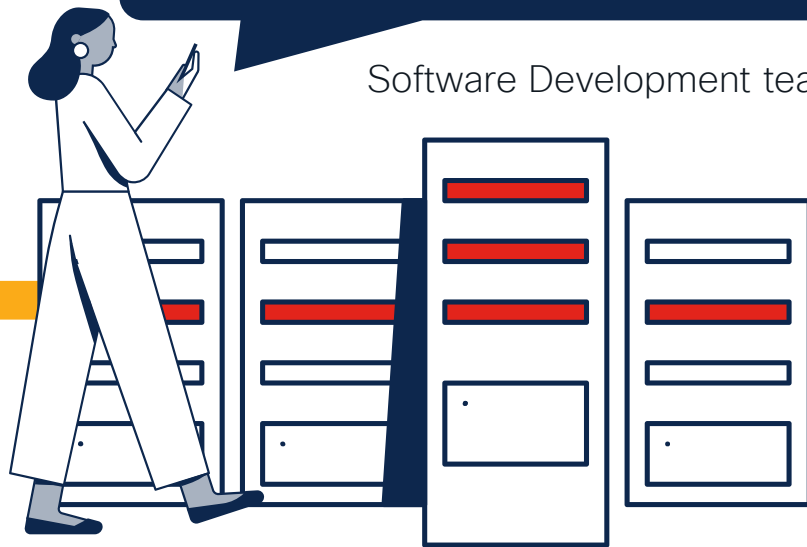
Network Operations team

Ticket FW025:  
“Standard  
config is not  
being pushed”



– “Guess this will be another night  
of troubleshooting ...”

Software Development team



# Post Mortem

- “How can the team prevent these situations from happening again?”



Syntax errors  
in some parts  
of the new  
code (corner  
cases)



Modified old  
code related  
to other User  
Stories



Didn't test  
the entire  
solution, just  
the newest  
features



Deployed the  
files by hand,  
and not the  
final version



The solution:  
Embracing NetDevOps  
as a "life mantra"





# Agenda

- NetDevOps at-a-glance
- NSO oriented pipeline design
- Platforms and tooling
- Demo
- Best practices
- References



## This session is about

CI/CD and NetDevOps 101

CI/CD best practices for NSO

CI/CD essential tooling for NSO

Real-life Use Case examples,  
code and demos

## This session is NOT about

In-depth CI/CD theory

NSO services development

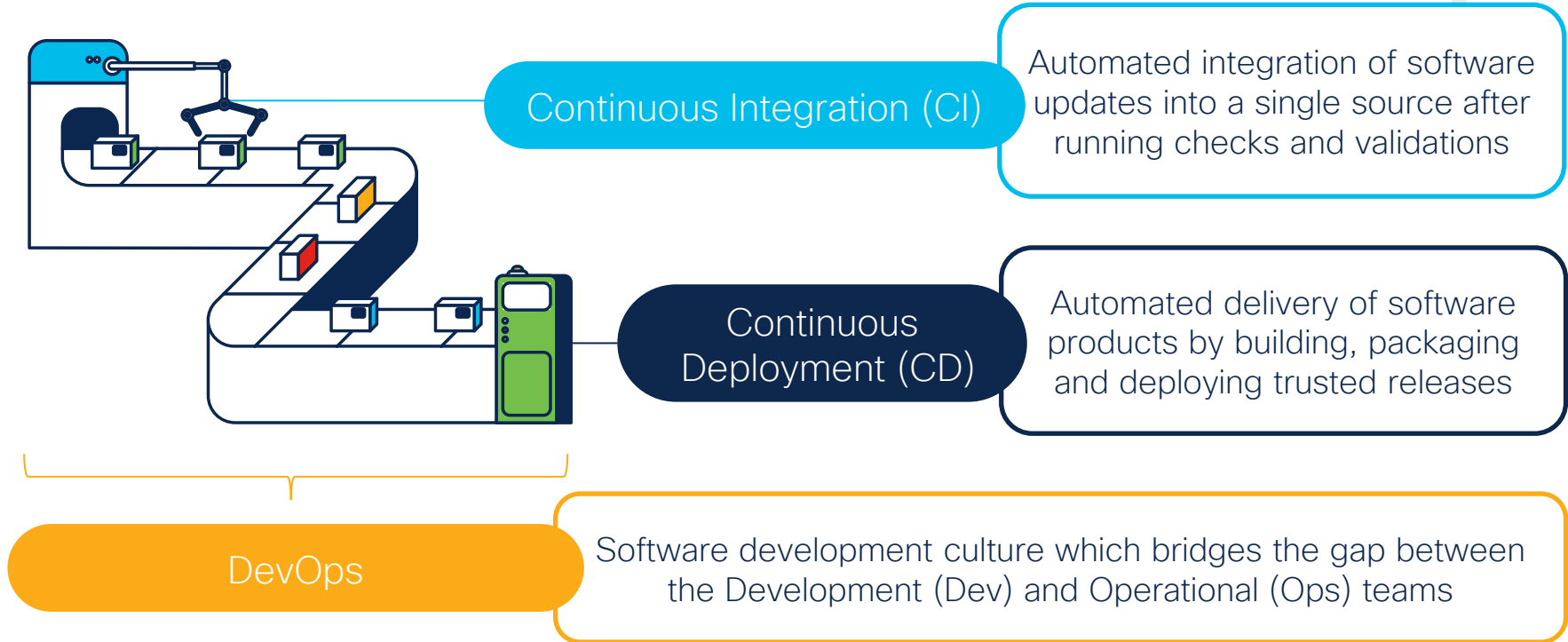
General Software development

Software delivery methodologies

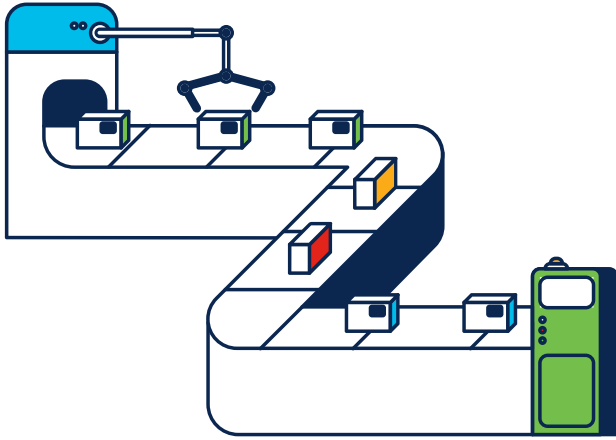
# NetDevOps at-a-glance



# What DevOps and CI/CD stand for



# Key benefits



Early detection of defects



No error risk due to manual work



Delivery quality and reliability assurance

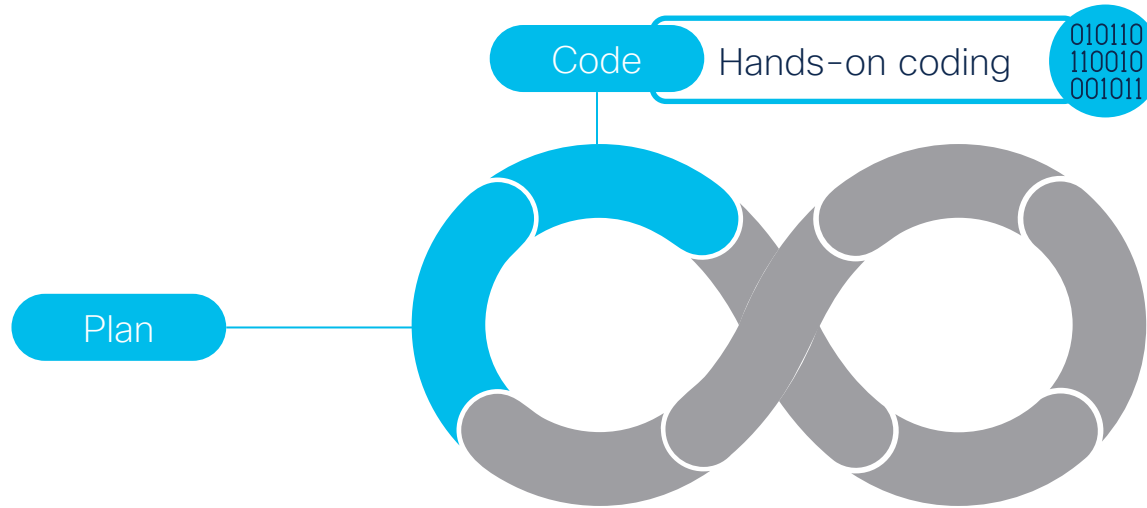


Faster, more efficient software delivery

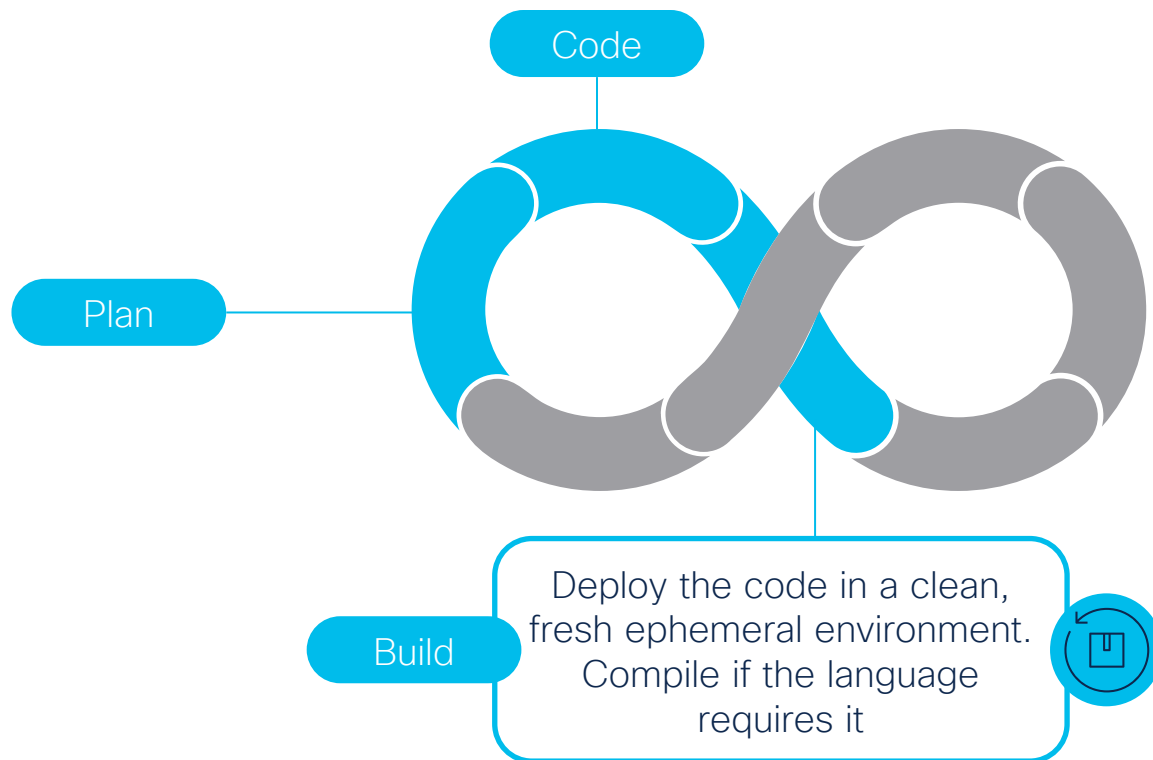
# Continuous Integration (CI) most common stages



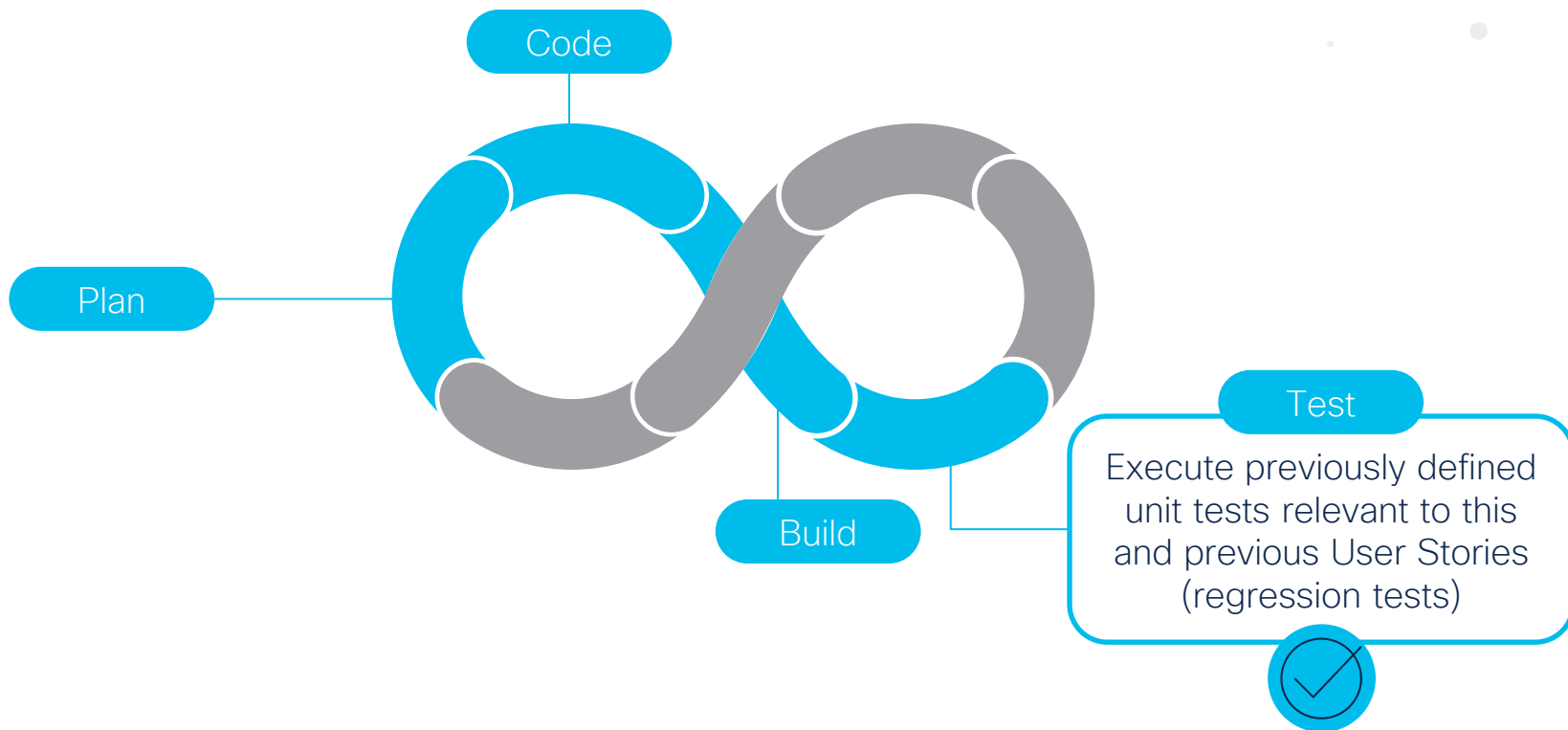
# Continuous Integration (CI) most common stages



# Continuous Integration (CI) most common stages

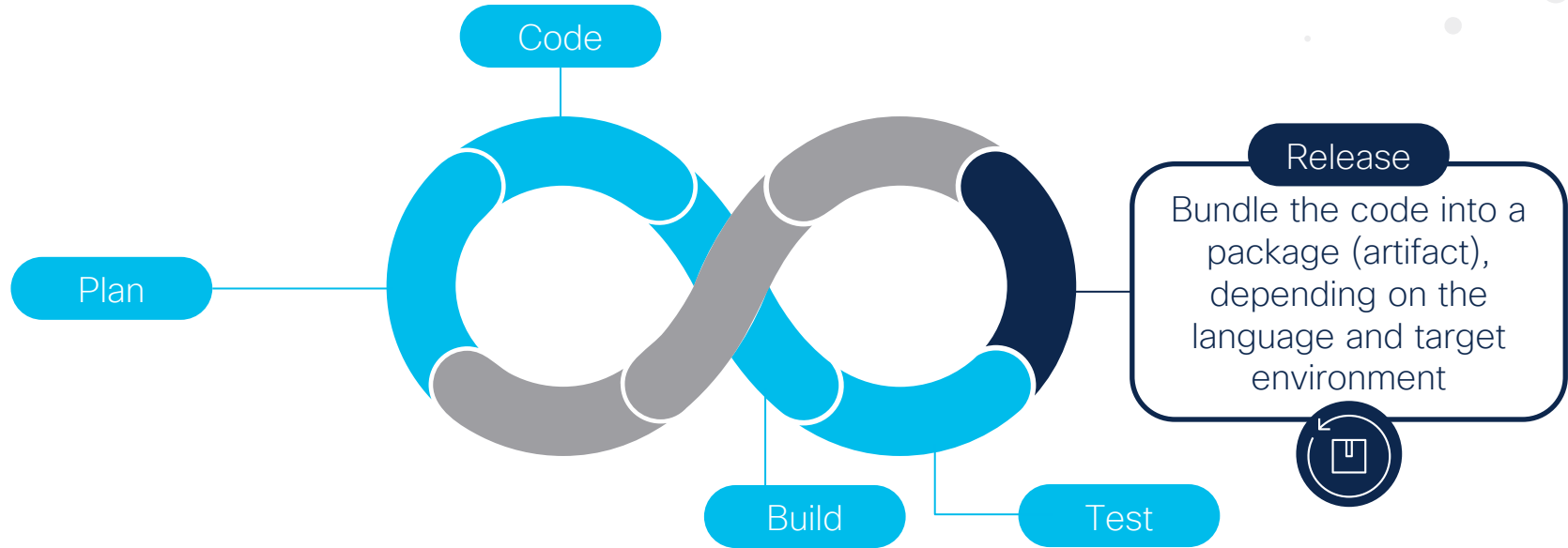


# Continuous Integration (CI) most common stages

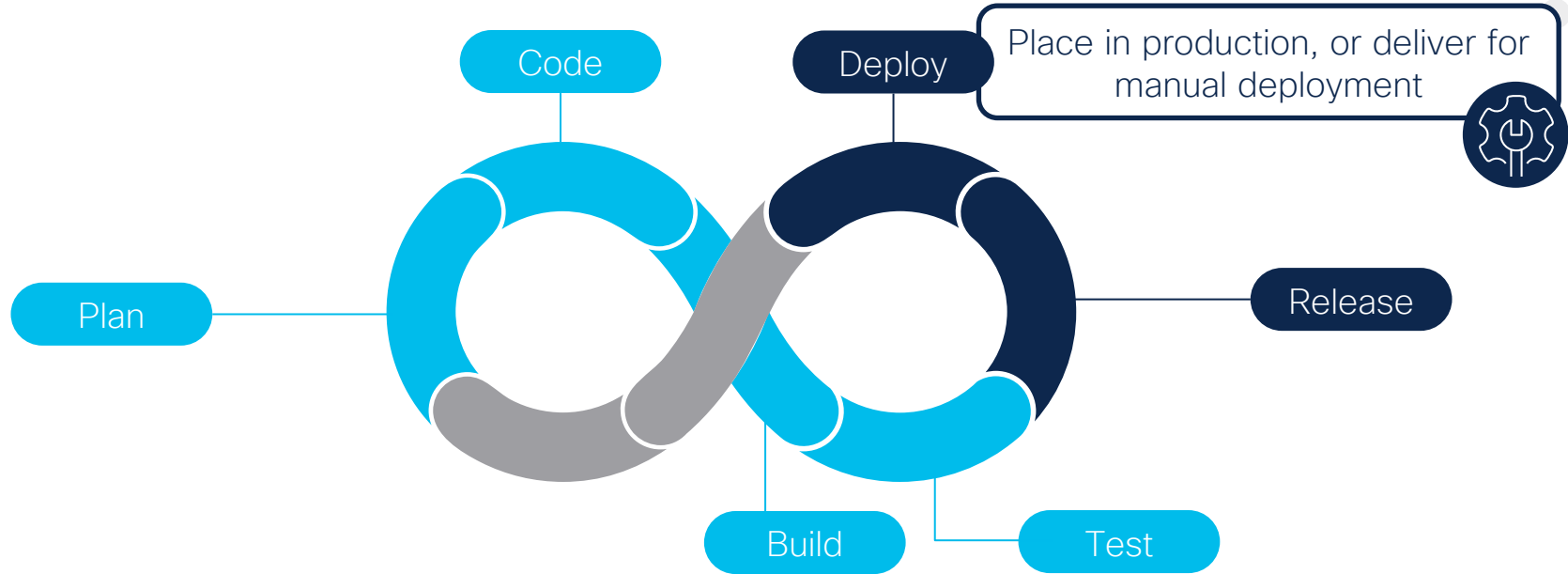




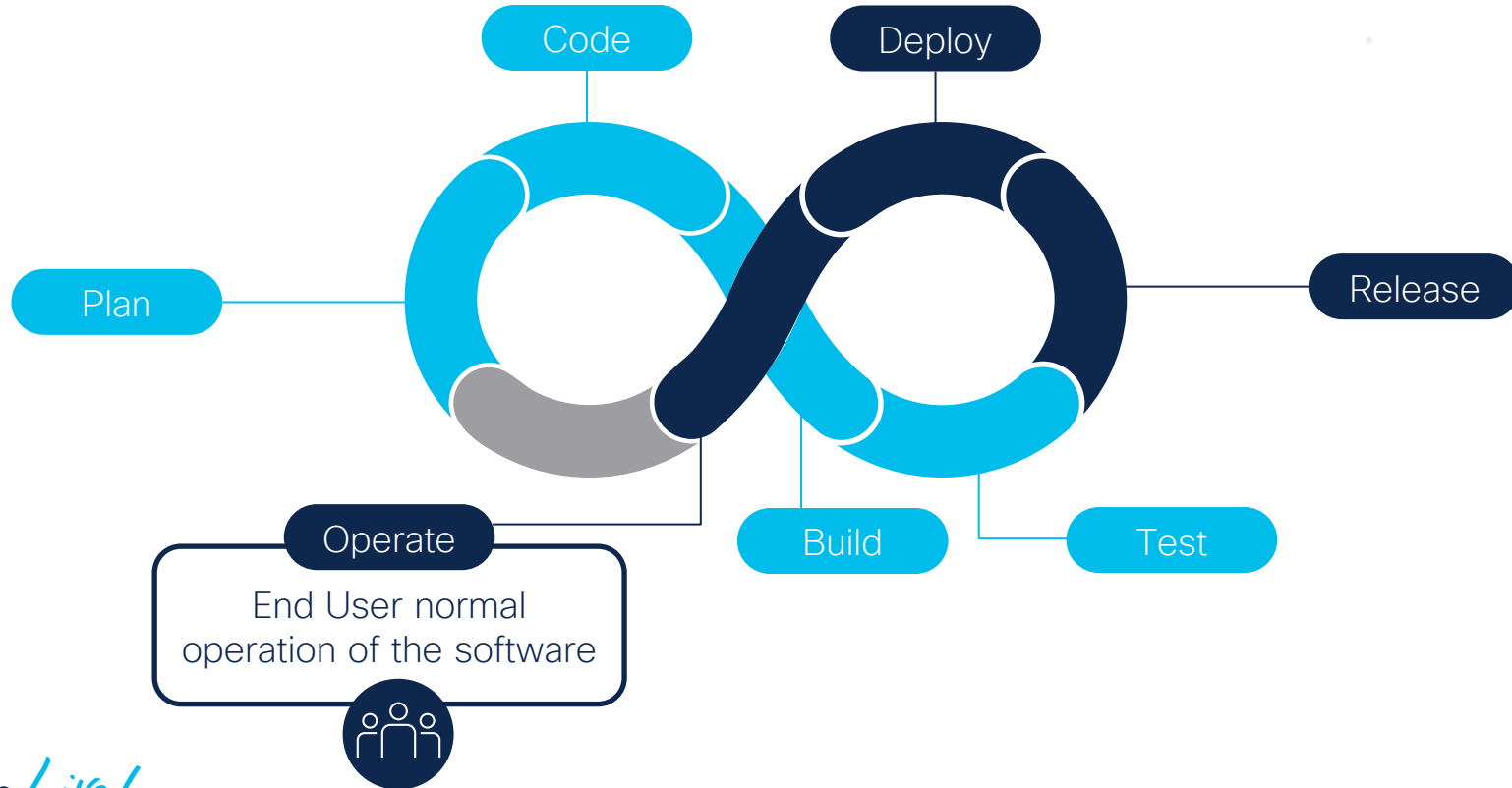
# Continuous Deployment (CD) most common stages



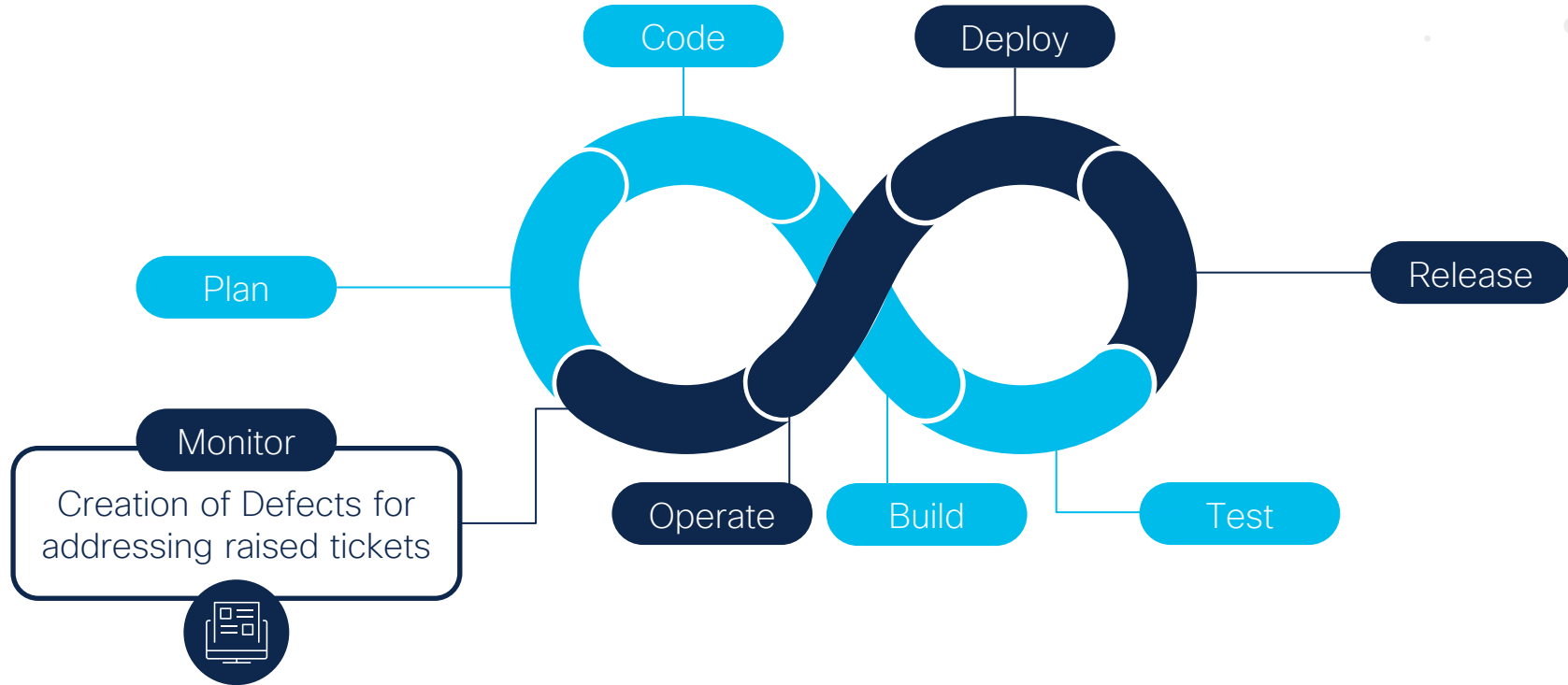
# Continuous Deployment (CD) most common stages



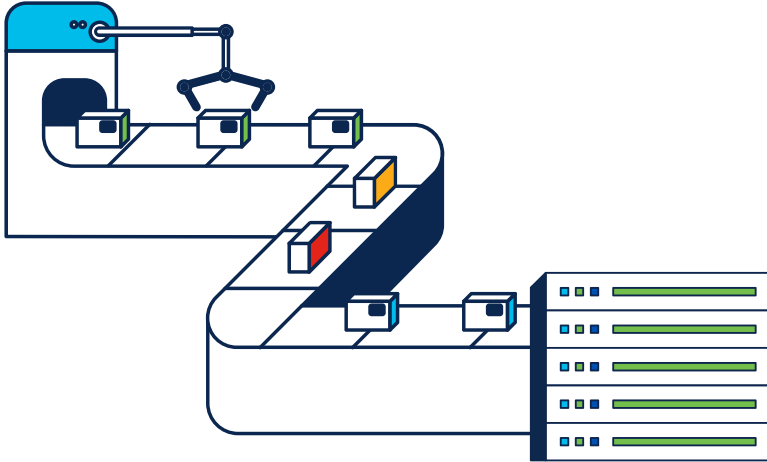
# Continuous Deployment (CD) most common stages



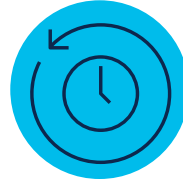
# Continuous Deployment (CD) most common stages



# Networks + DevOps practices = NetDevOps



Small but frequent changes in the network services



Roll service updates with more reliability



Mitigate disruption risks caused by bad code or manual intervention

# NSO oriented pipeline design



# What is a pipeline?



A CI/CD pipeline is a series of processes that run when triggered by an action



The processes are defined in stages, which are executed sequentially



# Pipeline stages

Source



SCM

Branch  
checkout

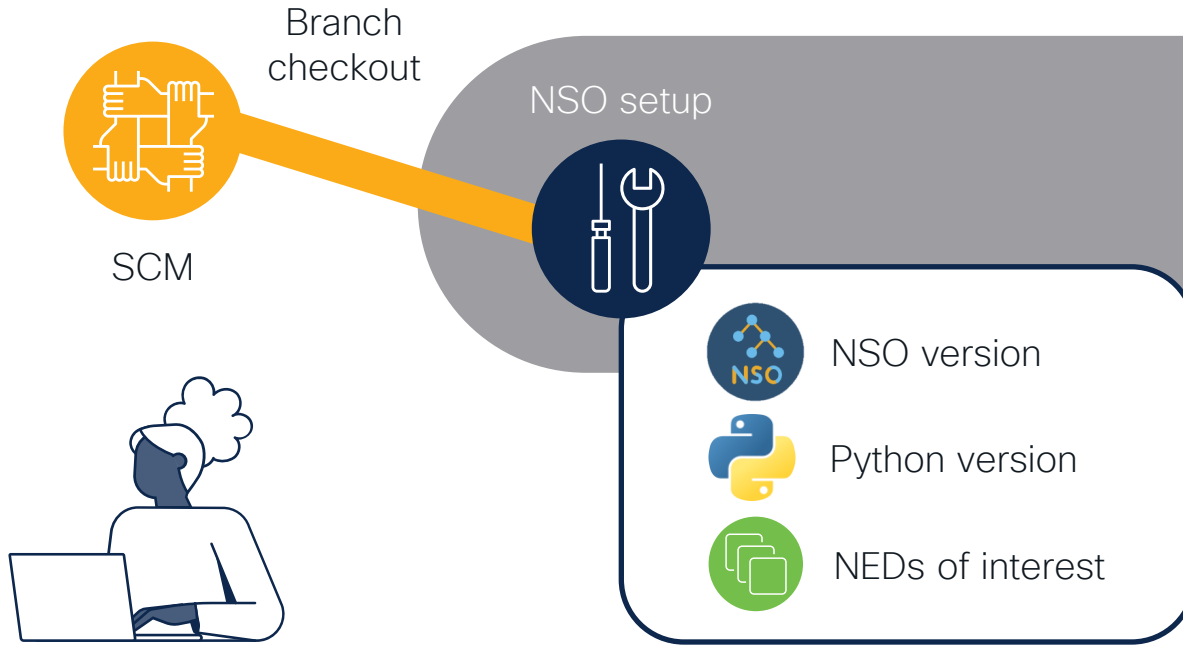


Source Code Management system for code  
versioning and collaboration



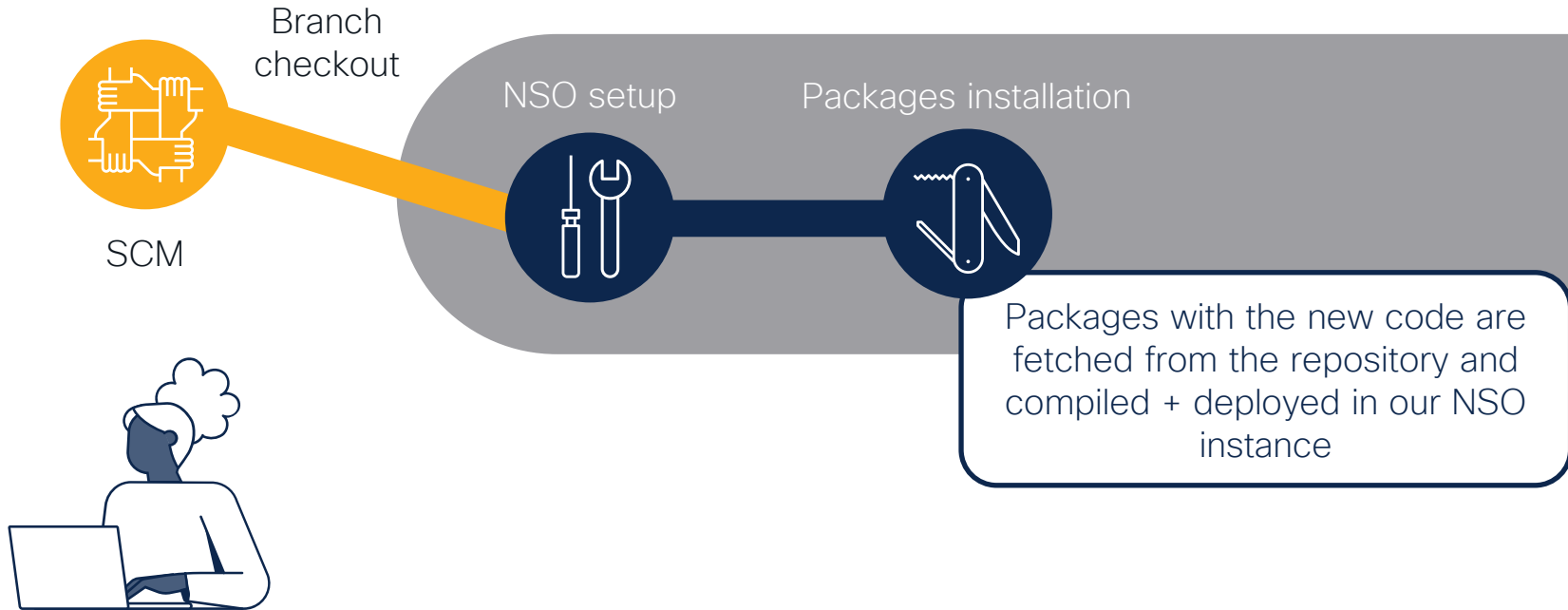
# Pipeline stages

Build



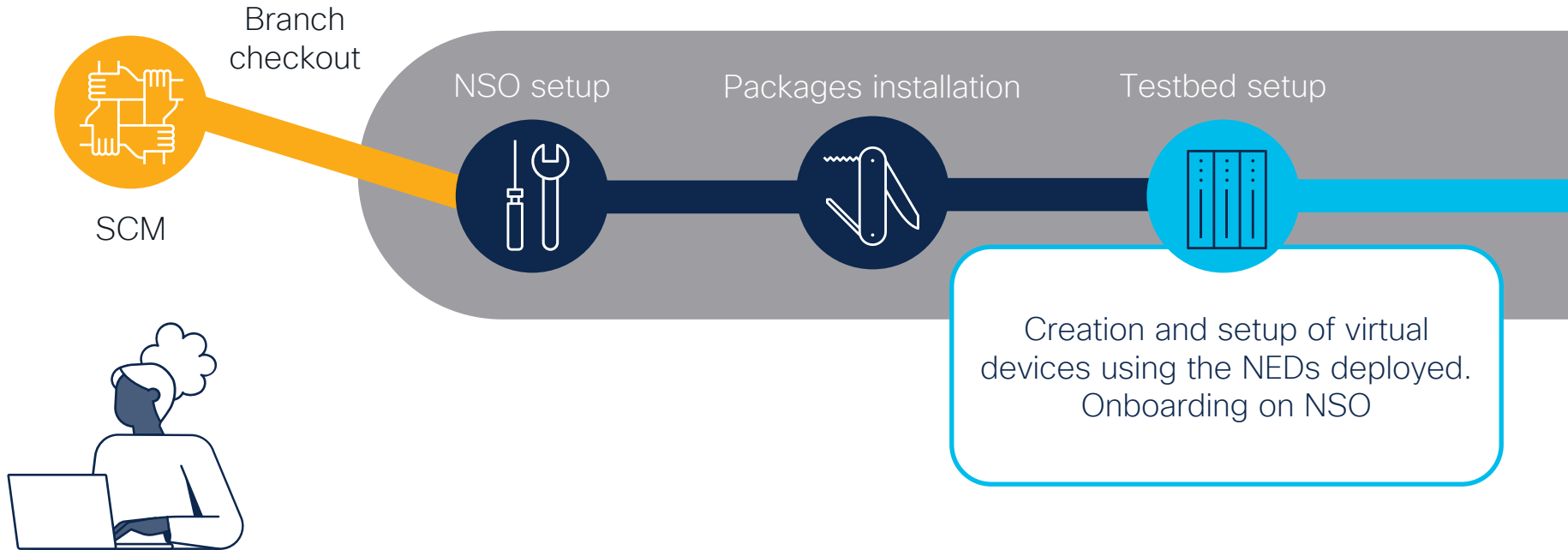
# Pipeline stages

Build



# Our pipeline stages

Test



# Our pipeline stages

Test

Test execution



Unit tests are executed in a sequential order

# Our pipeline stages

Deploy

Test execution

Artifact publishing



If all the tests pass, the compiled packages are bundled into a distributable format

# Our pipeline stages

Deploy

Test execution



Artifact publishing



Build results  
publishing



Publishing of the passed vs failed tests and general errors (UI, email, Webex Teams, etc)

# Our pipeline stages

Deploy

Test execution



Artifact publishing



Build results  
publishing



Cleanup



Deletion of ephemeral  
resources (NSO instance,  
virtual devices, etc)

# Our pipeline stages

Deploy

Test execution



Artifact publishing



Build results  
publishing



Cleanup



Deployment of the new package version  
in production NSO server



# Platforms and tooling



# Source Code Management (SCM)



“Octocat”  
GitHub mascot



“GitLab Tanooki”  
GitHub mascot



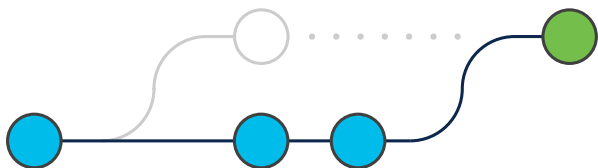
Track modifications to a source code repository



Collaborative code reviews,  
pushing and merging



Adaptive to Software  
Development workflows



# CI/CD platform: Jenkins



Leading open-source CI/CD platform



Code-based stage definition

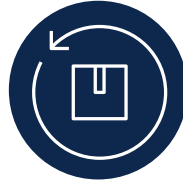


Plugin marketplace for the most popular platforms

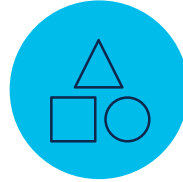
# NSO in Docker



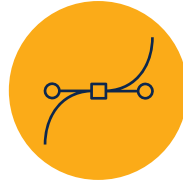
[github.com/NSO-developer/nso-docker](https://github.com/NSO-developer/nso-docker)



Project maintained by Cisco for easily running NSO in Docker



Deploy different NSO versions on-demand

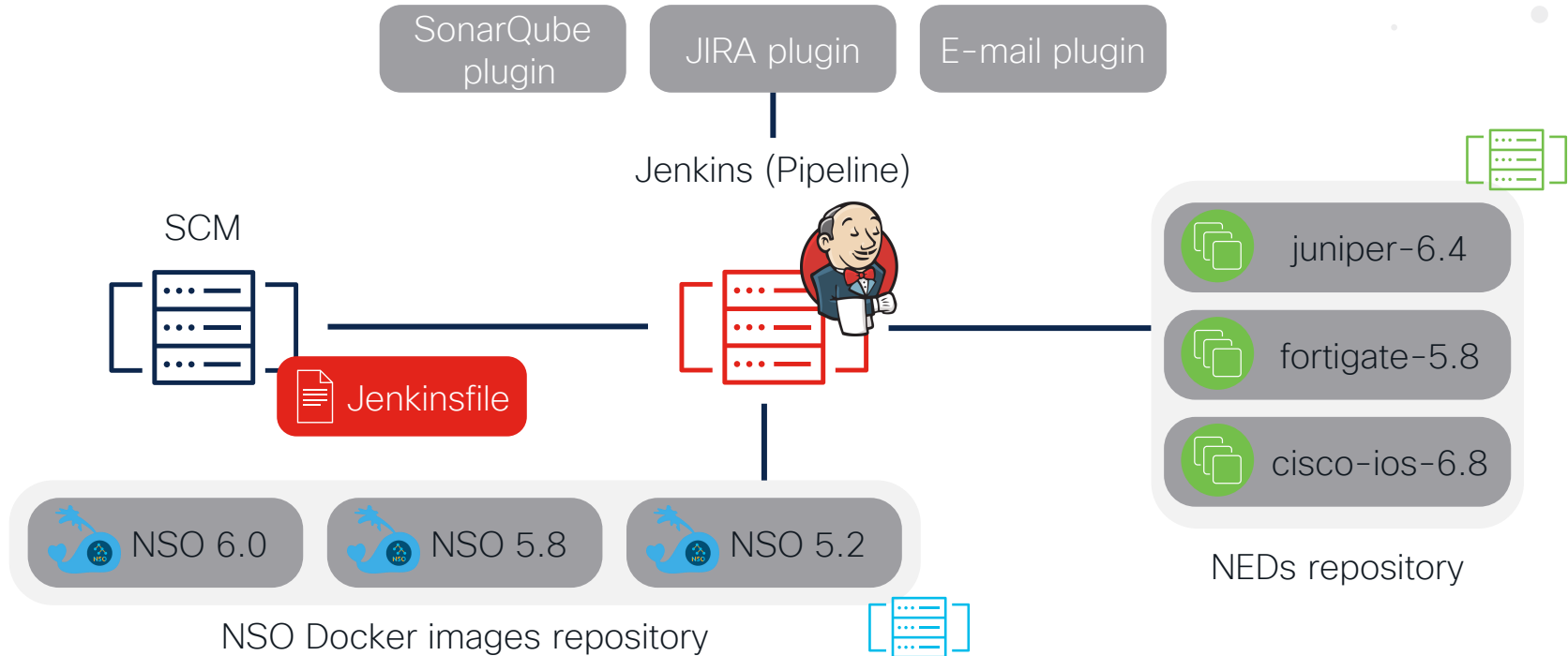


dev image (Development)  
base image (Production)  
nid skeleton (NEDs)

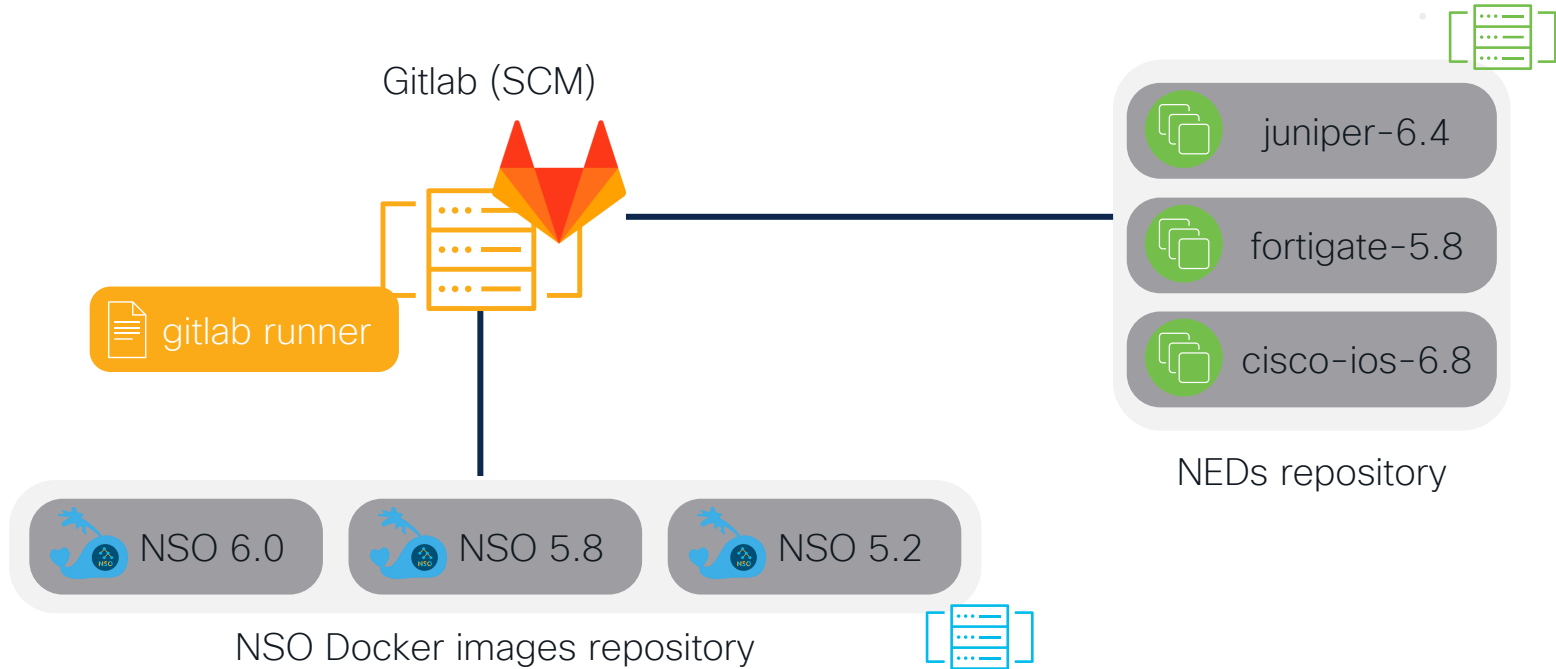


You must provide the NSO and NED files

# CI/CD architectures

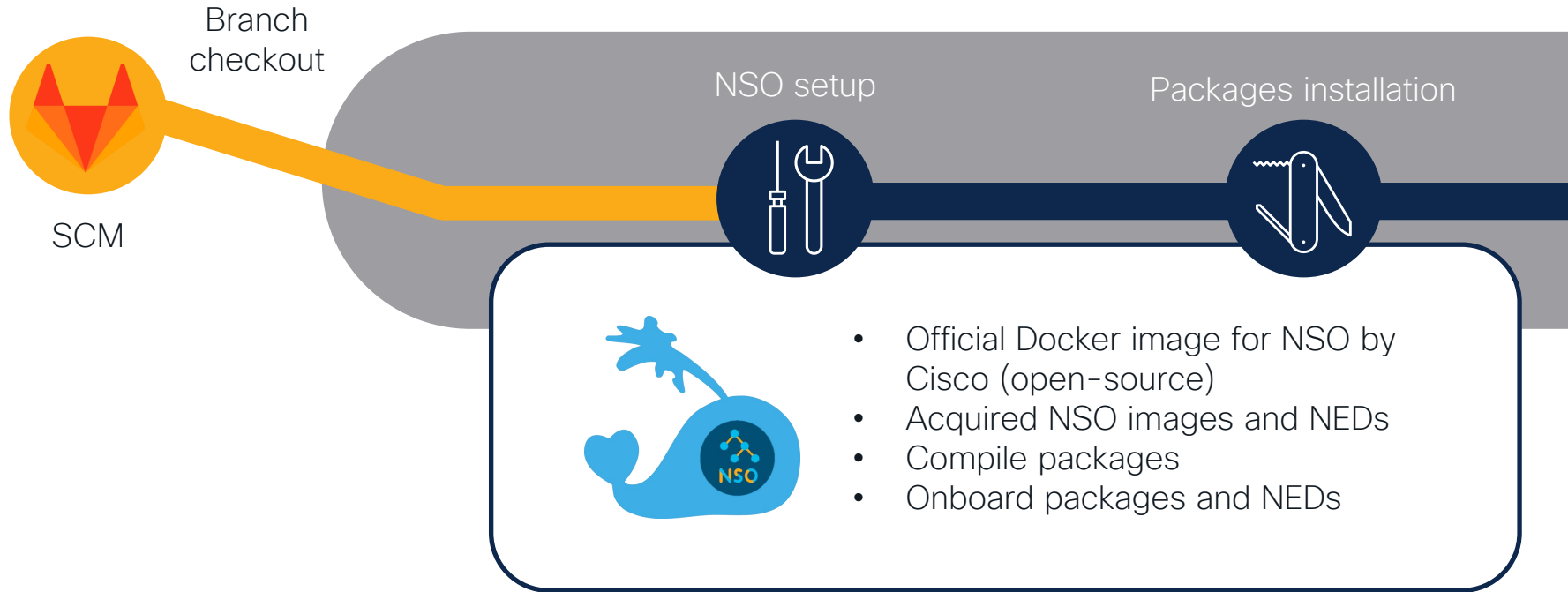


# CI/CD architectures



# Platforms and tooling

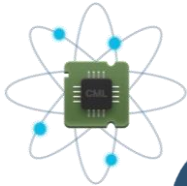
Build



# Platforms and tooling

Test

Testbed setup



- Create a Network Digital Twin
- Cisco Modelling Labs (Cisco)
- ncs-netsim (already included in NSO)



# Platforms and tooling

Test

Testbed setup

Test execution



- Can be as simple as Python unittest Test Suites
- pyATS Framework
- Robot Framework

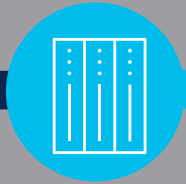
# Platforms and tooling

Deploy

Testbed setup

Test execution

Artifact publishing

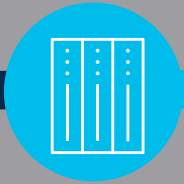


- Repository manager to organize all binary resources
- Jfrog Artifactory
- Also GitHub and GitLab have options for managing artifacts

# Platforms and tooling

Deploy

Testbed setup



Test execution



Artifact publishing



Build results publishing



Cleanup



- Bash scripting to publish results and teardown resources
- Notifications sending to 3<sup>rd</sup> party resources

# Demo

# Our User Story



Existing service  
package for  
access lists in  
Cisco IOS  
devices



User Story  
US00012:  
“Adding  
provisioning of  
interfaces”



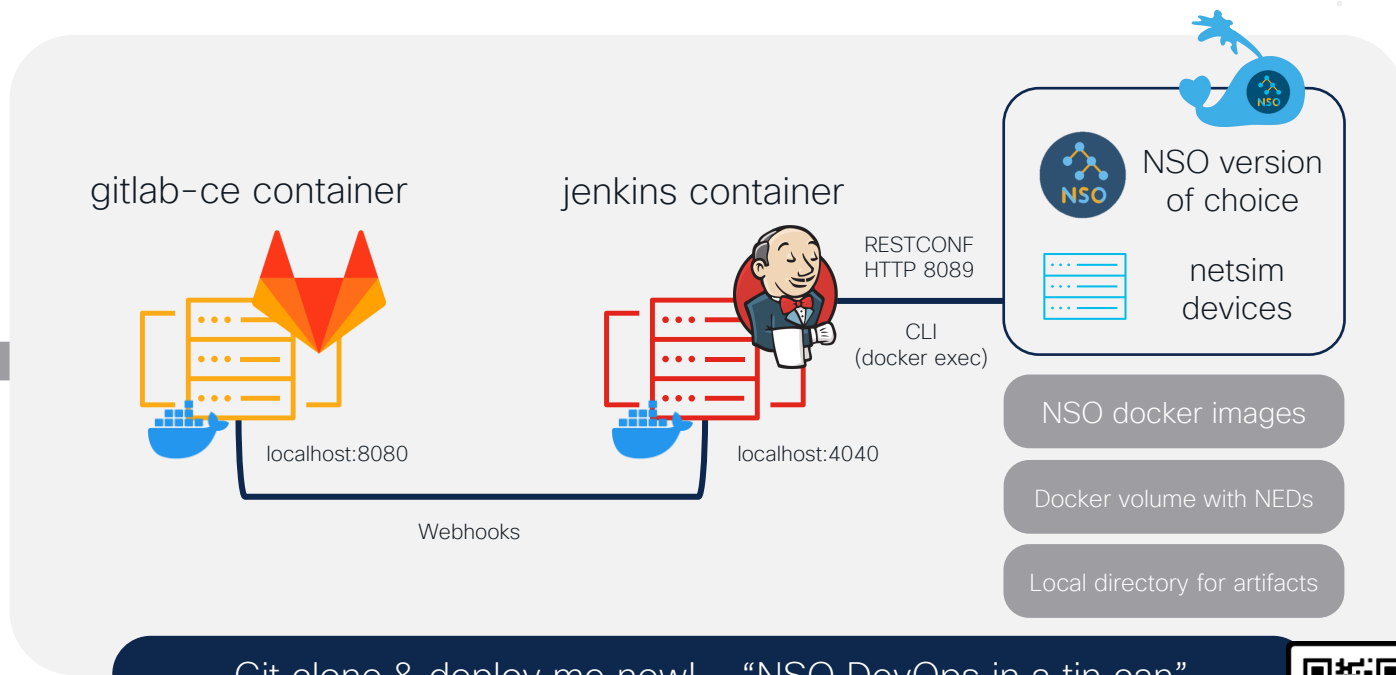
Test Case:  
Interface  
provisioning  
params match the  
expected payload



All previous  
tests from  
former US  
must pass as  
well



# Our CI/CD layout for NSO



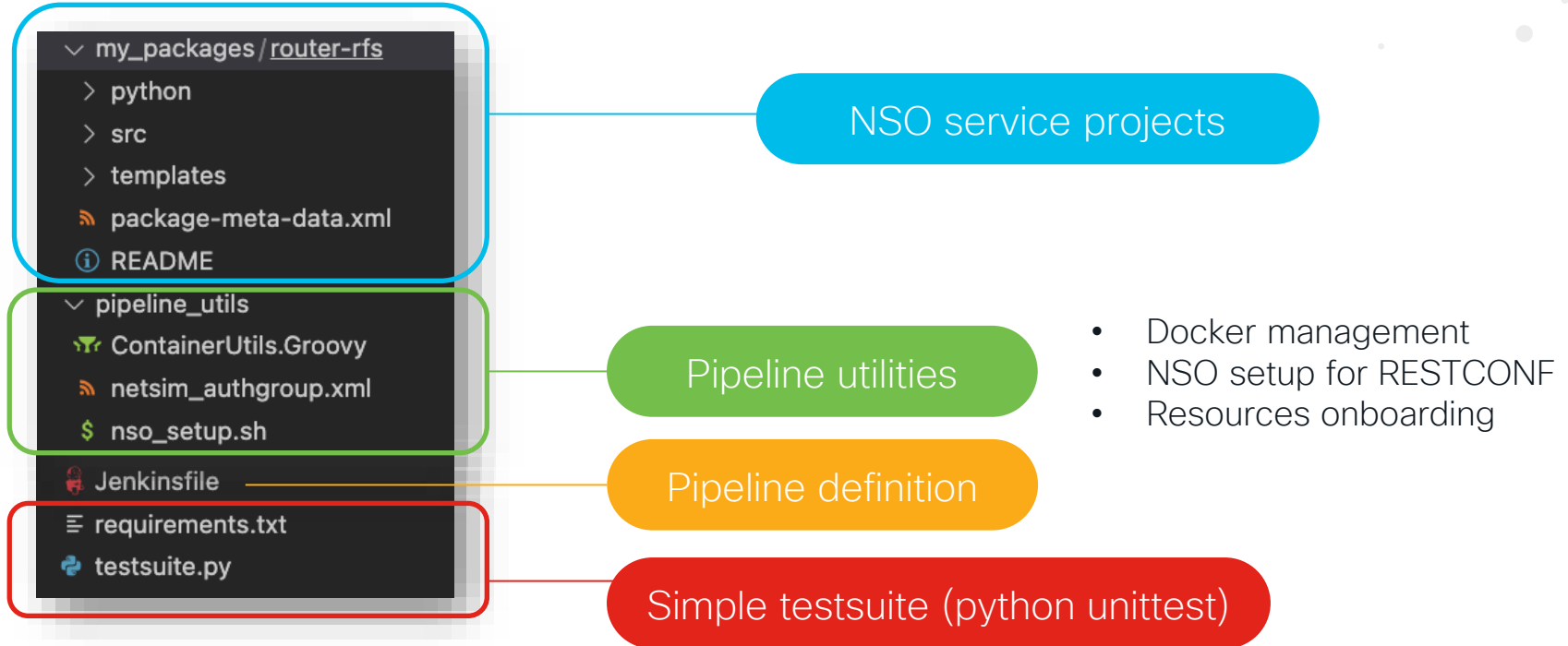
Git clone & deploy me now! – “NSO DevOps in a tin can”

[http://cs.co/nso\\_devops\\_tincan](http://cs.co/nso_devops_tincan)



SCAN ME

# Demo project structure



# Jenkinsfile: Pipeline as Code

## Stage View



```
pipeline {  
  agent any  
  stages {  
    stage('NSO setup') {...}  
    stage('NEDs setup') {...}  
    stage('Packages setup') {...}  
    ...  
  }  
}
```

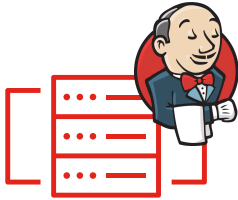
Definition of the actions to be performed on each stage with a file in the project directory



# NSO Services testing

testsuite.py

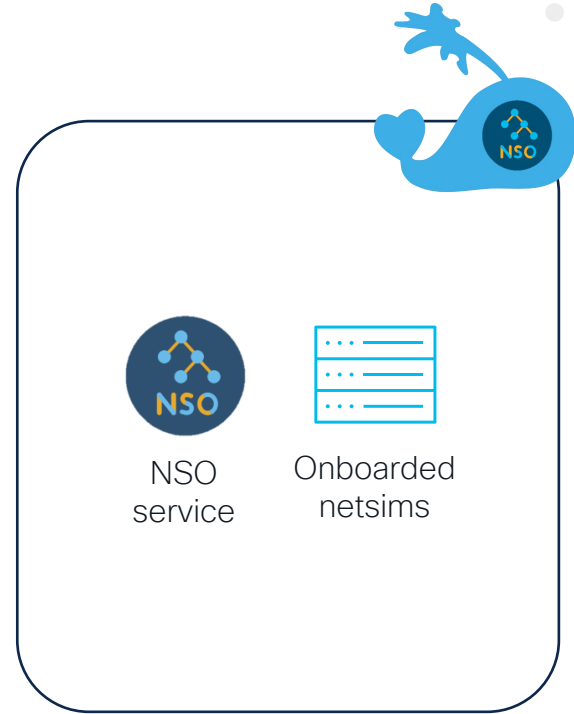
Jenkins (Pipeline)



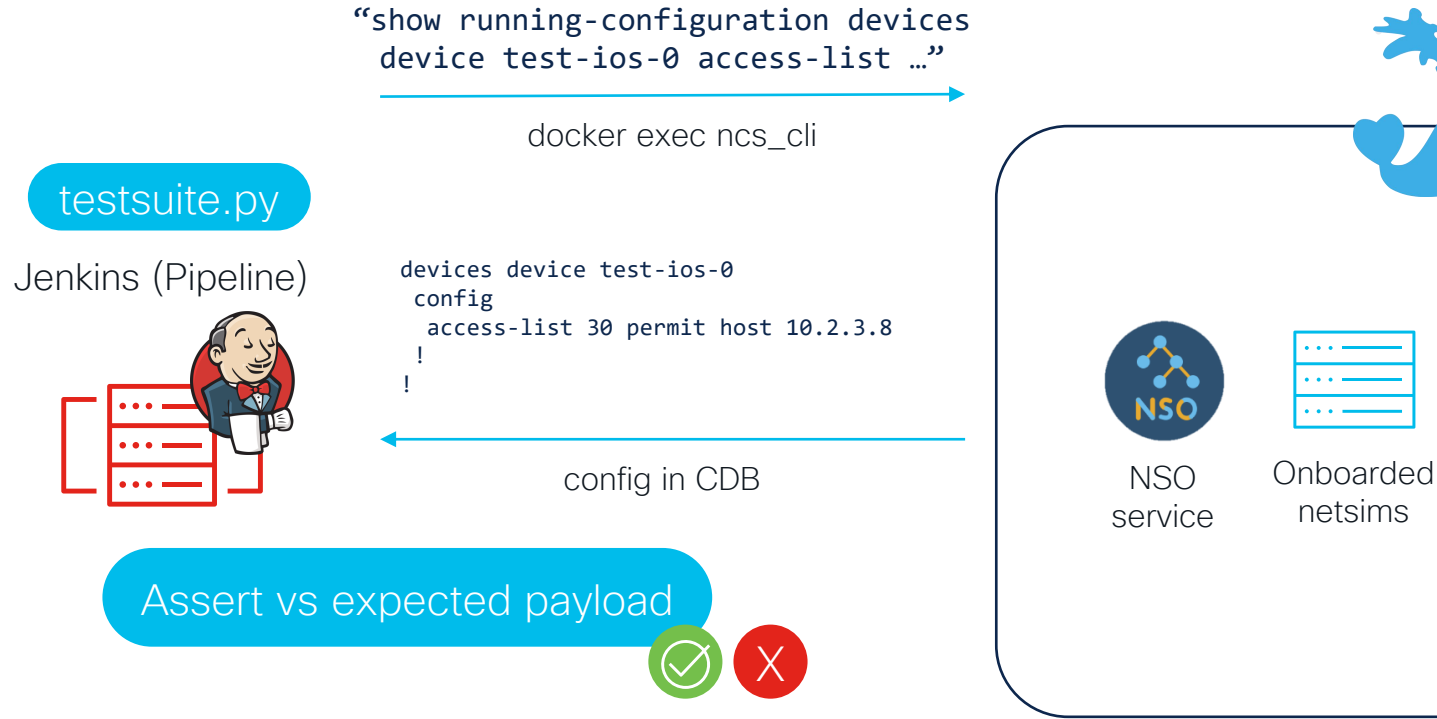
```
{
  "router-rfs:access-list-router-rfs":[
    {
      "device":"test-ios-0",
      "access_list":[
        {
          "id":"30",
          "action":"permit",
          "destination":"10.2.3.8"
        }
      ]
    }
  ]
}
```

RESTCONF PATCH

204 OK



# NSO Services testing



# Video



Branches (3) [nso-pipeline-de...]

GitLab Instance / test-project · X

+

localhost:8080/gitlab-instance-42118b3f/test-project

🔍 Search GitLab

🏠 📄 📧 📌 📁 📂 📅 📆 📇 📈 📉 📊 📋 📌 📍 📎 📏 📐 📑 📒 📓 📔 📕 📖 📗 📘 📙 📚 📛 📜 📝 📞 📟 📠 📡 📢 📣 📤 📥 📦 📧 📨 📩 📪 📫 📬 📭 📮 📯 📰 📱 📲 📳 📴 📵 📶 📷 📸 📹 📺 📻 📼 📽 📾 📿 📠 📡 📢 📣 📤 📥 📦 📧 📨 📩 📪 📫 📬 📭 📮 📯 📰 📱 📲 📳 📴 📵 📶 📷 📸 📹 📺 📻 📼 📽 📾 📿

👤

T test-project

Project information

Repository

Issues 0

Merge requests 0

CI/CD

Security & Compliance

Deployments

Packages and registries

Infrastructure

Monitor

Analytics

Wiki

Snippets

Settings

🚨 Critical security upgrade available

You are currently on version 15.7.2. We strongly recommend upgrading your GitLab installation. [Learn more about this critical security release.](#)

Upgrade now

Commit & NSO setup

GitLab Instance > test-project

T test-project

Project ID: 2

🔑

🔔

☆ Star 0

🍴 Fork 0

🔗 24 Commits

🌿 3 Branches

🏷️ 0 Tags

💾 573 KB Project Storage

🌿 Merge branch 'fixMultiB' into 'main'

Administrator authored 9 minutes ago

9ef79c73

main test-project +

Find file

Web IDE

📄

Clone

Add README

Add LICENSE

Add CHANGELOG

Add CONTRIBUTING

Enable Auto DevOps

Add Kubernetes cluster

Set up CI/CD

Configure Integrations

Name	Last commit	Last update
my_packages/router-rfs	Test for multibranch	5 hours ago
pipeline_utils	Issuing multibranch fix	21 minutes ago
Jenkinsfile	Testing multibranch support	1 hour ago
requirements.txt	Initial commit	1 day ago
testsuite.py	Initial commit	1 day ago

Dashboard > nso-pipeline-demo > US00012 > #1

```
+ docker exec -i nso6.0 bash -l -c /etc/init.d/
[Pipeline] }
[Pipeline] // script
[Pipeline] }
[Pipeline] // stage
[Pipeline] stage
[Pipeline] { (NEDs setup)
[Pipeline] script
[Pipeline] {
[Pipeline] load
[Pipeline] { (pipeline_utils/ContainerUtils.Groovy)
[Pipeline] }
[Pipeline] // load
[Pipeline] sh
+ docker cp /root/../my_neds/cisco-ios-cli-6.88/ nso6.0:/nso/run/packages/
[Pipeline] }
```

NEDs and packages setup

## Netsim devices setup

```
[Pipeline] echo
```

```
>>> System upgrade is starting.
```

```
>>> Sessions in configure mode must exit to operational mode.
```

```
>>> No configuration changes can be performed until upgrade has completed.
```

```
>>> System upgrade has completed successfully.
```

```
reload-result {
```

```
  package cisco-ios-cli-6.88
```

```
  result true
```

```
}
```

```
reload-result {
```

```
  package router-rfs
```

```
  result true
```

```
}
```

```
[Pipeline] }
```

Netsim devices onboarding

```
[Pipeline] script
[Pipeline] {
[Pipeline] load
[Pipeline] { (pipeline_utils/ContainerUtils.Groovy)
[Pipeline] }
[Pipeline] // load
[Pipeline] sh
+ docker exec -i nso6.0 bash -l -c ncs-netsim create-network /nso/run/packages/cisco-
ios-cli-6.88/ 1 test-ios-
[Pipeline] sh
+ docker exec -i nso6.0 bash -l -c ncs-netsim start
[Pipeline] echo
DEVICE test-ios-0 OK STARTED

[Pipeline] sh
+ docker exec -i nso6.0 bash -l -c ncs-netsim list
```

Testbed execution

```
[Pipeline] }
[Pipeline] // stage
[Pipeline] stage
[Pipeline] { (Test execution)
[Pipeline] script
[Pipeline] {
[Pipeline] load
[Pipeline] { (pipeline_utils/ContainerUtils.Groovy)
[Pipeline] }
[Pipeline] // load
[Pipeline] sh
+ cd /var/jenkins_home/workspace/nso-pipeline-demo_US00012/
+ pip install -r requirements.txt
[Pipeline] sh
+ cd /var/jenkins_home/workspace/nso-pipeline-demo_US00012/
+ python3 testsuite.py
```



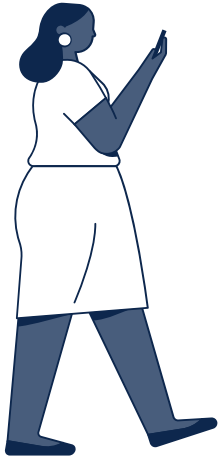
Artifact publishing & workspace cleanup

```
[Pipeline] script
[Pipeline] {
[Pipeline] load
[Pipeline] { (pipeline_utils/ContainerUtils.Groovy)
[Pipeline] }
[Pipeline] // load
[Pipeline] sh
+ cd /var/jenkins_home/workspace/nso-pipeline-demo_US00012/
+ mkdir releases/
[Pipeline] sh
+ docker exec -i nso6.0 bash -l -c cd /nso/run/packages/ && tar -czvf router-
rfs_US00012_03-02-2023_14-54-21.tar.gz router-rfs/
[Pipeline] sh
+ docker cp nso6.0:/nso/run/packages/router-rfs_US00012_03-02-2023_14-54-21.tar.gz
/var/jenkins_home/workspace/nso-pipeline-demo_US00012/releases/
[Pipeline] }
```

# Best practices



# Best practices



Manage individual branches for each User Story in your repository



Commit frequently, but never to the “main” branch (Issue a PR to it instead)

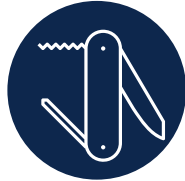
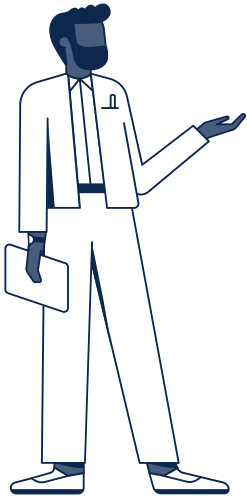


Keep your tests as small and clear as possible. Errors must be very descriptive



Be merciless! All tests **MUST** pass

# Best practices



Use the NSO REST interface for provisioning and execution in your pipeline



Jenkins allows the addition of plenty of plugins (Code smells, vulnerabilities, etc)



Cisco can help with in-house frameworks for robust CI/CD (CXTA, CXTM, NSOArc)

# References



# References

“How automation is driving network engineer skills transformation”

<https://www.cisco.com/c/dam/en/us/solutions/collateral/executive-perspectives/technology-perspectives/automation-driving-network-eng-skills-trans.pdf>

DevNet NetDevOps resources

<https://developer.cisco.com/netdevops/>

NSO for Docker

[github.com/NSO-developer/nso-docker](https://github.com/NSO-developer/nso-docker)

NSO CI/CD DevNet Learning Lab

<https://developer.cisco.com/learning/labs/nso-cicd/introduction/>

Our demo repo: NSO DevOps in a tin can

<http://cs.co/nsodevopstincan>

Getting started with Jenkins pipelines

<https://www.jenkins.io/doc/pipeline/tour/hello-world/>



# Service Provider

## SP Automation and Orchestration

This learning map explains how Cisco's innovations and software solutions enable mass-scale programmable next generation networks. The sessions explore the latest in automation and orchestration to model-based operations and programmability while trusting cutting-edge security capabilities embedded within network elements and software assurance platforms monitoring them.

### START

- Feb 6 | 08:45  
**TECOPS-2003**  
Embracing SRE Practices in Infrastructure
- Feb 6 | 08:45  
**TECSPG-2014**  
Cisco Converged SDN Transport
- Feb 6 | 14:15  
**TECOPS-1201**  
From Zero to Hero: Cisco Network Service Orchestrator (NSO)
- Feb 7 | 08:30  
**BRKOPS-2136**  
Experience Telemetry - Driving Insights and Actions
- Feb 7 | 08:45  
**BRKOPS-2376**  
Expand your Automation Journey with new Cisco NSO Use Cases and Features

- Feb 7 | 10:00  
**BRKSP-2080**  
Crosswork Hierarchical Controller - Cross layer-vendor-domain Automation
- Feb 7 | 14:00  
**BRKOPS-2312**  
Do's and Don'ts in Network Test Automation
- Feb 7 | 15:30  
**BRKSP-2637**  
Network Automation with Routed Optical Networking (RON) Architecture
- Feb 7 | 17:00  
**BRKSPG-2263**  
Design, Deploy and Manage Transport Slices using SDN Controller and Assurance
- Feb 8 | 08:30  
↓  
**IBOOPS-2270**  
Get Ready for the Next Generation of Incident Response and Analysis

If you are unable to attend a live session, you can watch it [On Demand](#) after the event

CISCO *Live!*

- Feb 8 | 08:30  
**IBOOPS-2270**  
Get Ready for the Next Generation of Incident Response and Analysis
- Feb 8 | 12:00  
**BRKSPG-2028**  
Management of IP+Optical Networks Using an SDN Controller Architecture
- Feb 8 | 12:00  
**BRKSPG-2028**  
Management of IP+Optical Networks Using an SDN Controller Architecture
- Feb 8 | 13:30  
**BRKSPG-2664**  
Automate 5G Datacentre and Transport Components with NSO Cross-Domain Function Packs

- Feb 8 | 16:45  
**BRKSPG-2474**  
Reduce Resolution Time with a Service-Centric Approach to Troubleshooting
- Feb 9 | 08:30  
**LTROPS-1964**  
Test Automation for everyone using CXTA
- Feb 9 | 08:30  
**LTROPS-2417**  
Automate your Network Migration
- Feb 9 | 08:30  
**LTROPS-2711**  
Unified End 2 End Test Automation with CX Test Manager (CXTM)
- Feb 9 | 08:45  
**BRKMPL-2131**  
Deploying VPNs Over Segment Routed Networks Made Easy

- Feb 9 | 10:30  
**BRKOPS-2176**  
Leveraging Advanced Automation Capabilities in the Fault Management System
- Feb 9 | 15:45  
**BRKSPG-2250**  
Eliminate Congestion Surprises and Fire Drills Forever with Crosswork Cloud - Traffic Analysis as a Service
- Feb 10 | 11:00  
**BRKOPS-2766**  
How to Supercharge your Next-Gen Network with AIOps and Managed Services
- Feb 10 | 11:00  
**FINISH BRKSPG-2031**  
Deploying XR Programmability in Production Networks

If you are unable to attend a live session, you can watch it [On Demand](#) after the event



# Complete your Session Survey

- Please complete your session survey after each session. Your feedback is important.
- 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](https://ciscolive.com/on-demand).



The bridge to possible

# Thank you

CISCO *Live!*

CISCO *Live!*

