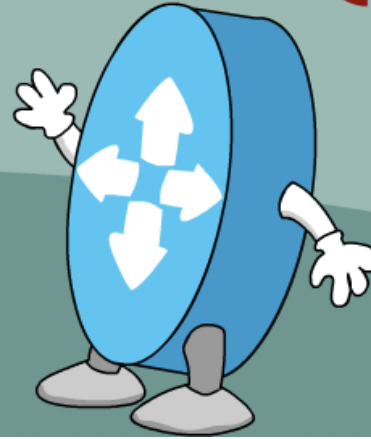




# NETDEVOPS {LIVE!}



DEVNET

What's the big deal about Source Control? Why your network configurations should be in "git"

Matthew DeNapoli

Developer Advocate, DevNet

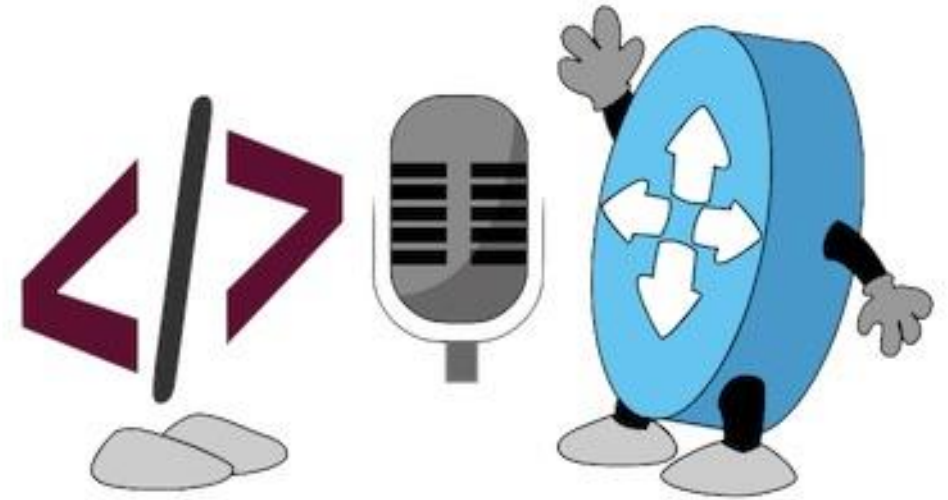
Twitter: @theDeNap

Season 1, Talk 5

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

# What are we going to talk about?

- Where in the NetDevOps Toolbag are we?
- Where in the CI/CD Pipeline are we?
- Need for Version Control
- What do we “git”?
- Git, by the book
- Demo Git flow using Gitlab



# The NetDevOps Engineers Tool Bag

(Example tools, not comprehensive)

Distributed Source Control  
(git, Subversion, Mercurial, GitHub, BitBucket, GitLab)

Build Server  
(GitLab, Jenkins, Team City, Drone)

Configuration Management  
(Ansible, Puppet, NSO, NAPALM, DIY)

Network Test Tooling  
(pyATS, TRex, Robot, Behave)

Telemetry & Monitoring  
(ELK, Grafana, Pipeline, UTM)

CLI

SNMP

NETCONF/  
RESTCONF

gRPC

REST APIs

YANG/Native Data Model

Configuration Data

Operational Data

Network Device

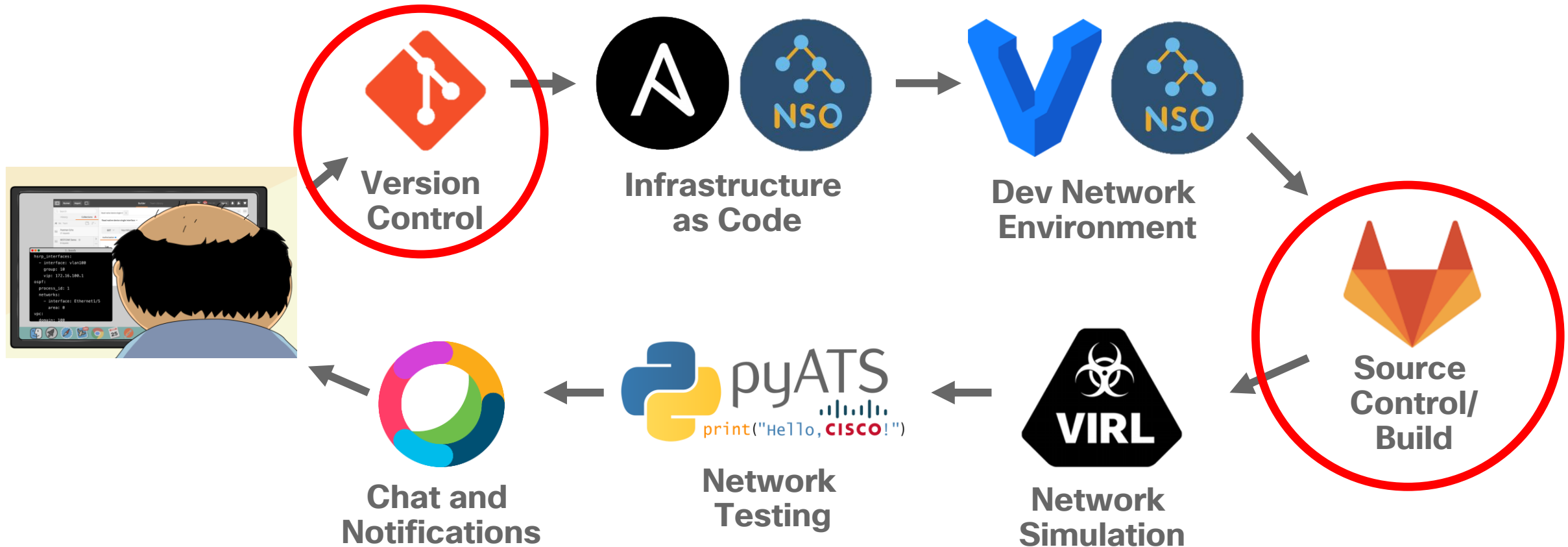
Network  
Virtualization  
Platforms  
(VIRL/CML, NFVIS,  
Vagrant)

Development Environment  
(Vagrant, NSO, VIRL/CML)

Test Environment  
(VIRL/CML)

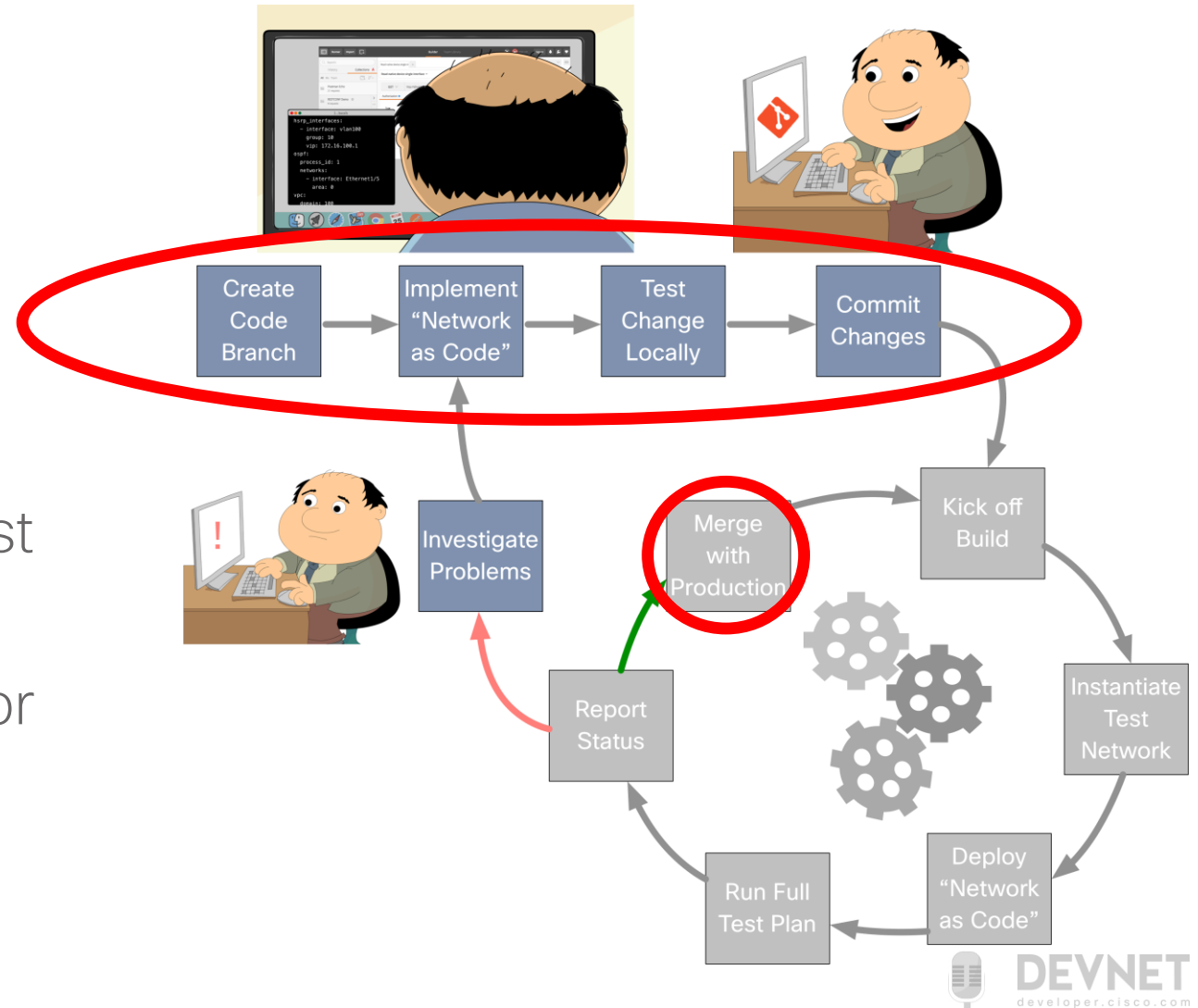
Production Environment

# NetDevOps CICD Demo Environment



# NetDevOps Configuration Pipeline

- “Treating the Network as Code”
- Network Configuration stored in Source Control
- Changes are proposed in code “branches”
- CI/CD Build Servers deploy and test proposed configurations
- Successful configurations ready for deployment to “Production”



# The Need for Version Control

How do I make incremental changes and share my work with others?

How do I go back to the version of this file from (yesterday, last week, last year, ...)?

What changed between version X and version Y of a file?

People have been making changes to the same file (or set of files)...  
How do I reconcile and merge all these changes?

# What do I “git” from Github/Gitlab/others?

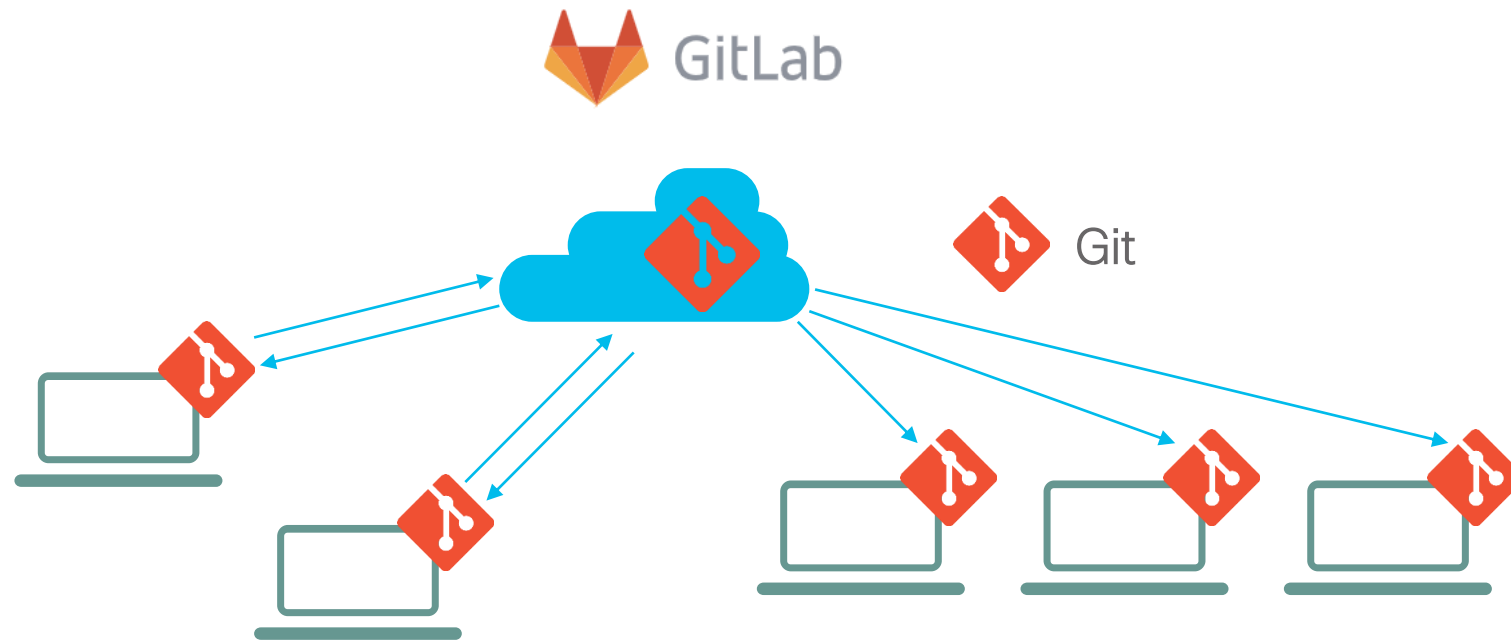
- **Project Management**

- Issues
- Tasks
- Comments/Notes
- CI/CD integrations

- **Code Review**

- Propose Changes – Pull/Merge request
- Request Review/Manage Feedback
- See Diff
- Branch management

# Git vs. GitLab (or GitHub or BitBucket, etc.)



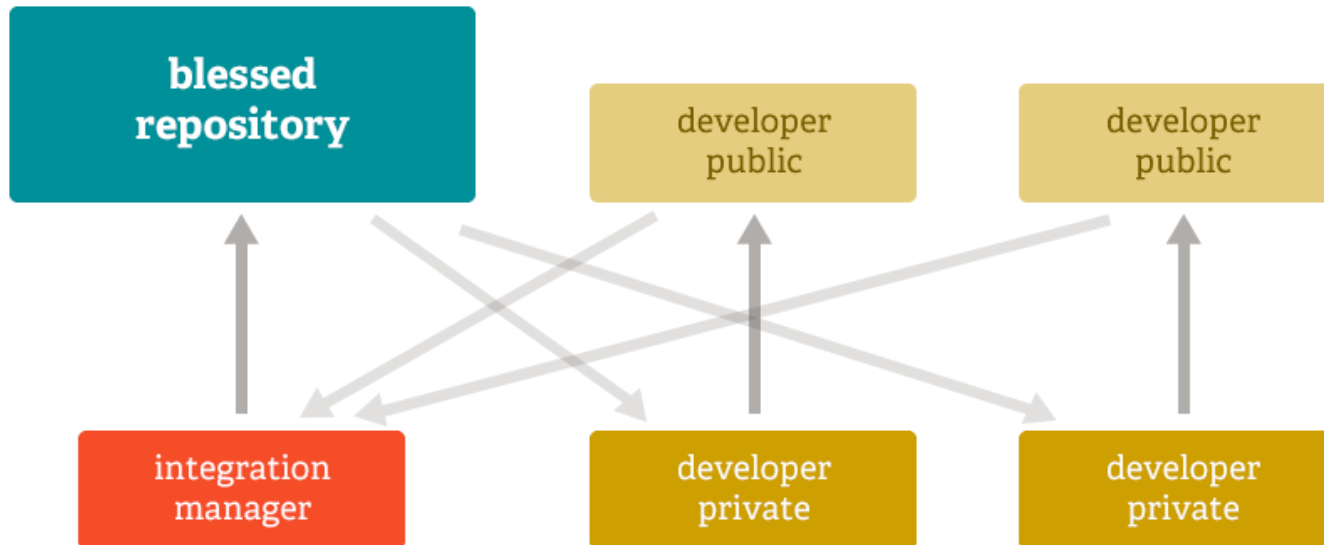


# Basic Git Terminology

- **Repository (Repo)** – A vault for storing version controlled files
- **Working Directory** – The visible directory and its contents
- **Versioned Files** – Files you have asked Git to track
- **Un-Versioned Files** – Files in your working directory not tracked by Git
- **Commit** – Snapshot in time (of your version controlled files)
- **Branches** – A safe place for you to work

# Distributed Version Control

- Opens up to new workflows: git flow
- Each system has an exact replica of the repo as other collaborators.



<https://git-scm.com/images/about/workflow-b@2x.png>

# A Peak Under the Hood

- Commits contain Trees
- Trees contain links to Files
- Git stores *full copies of all Changed Files*

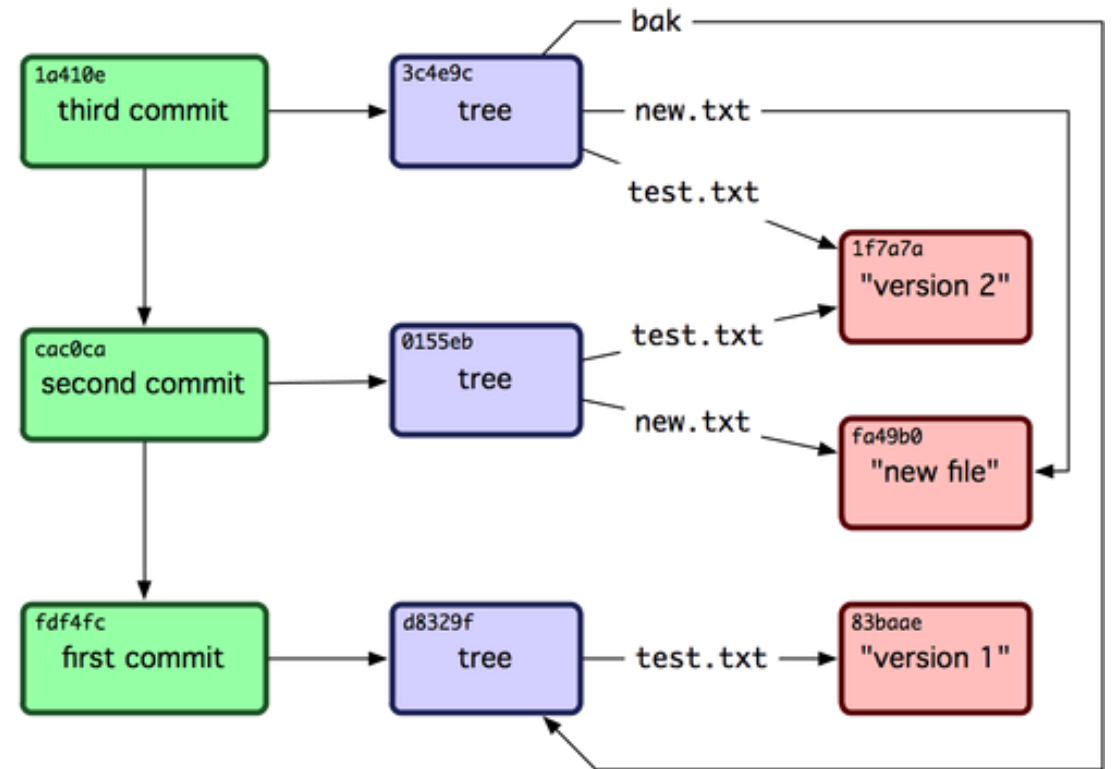
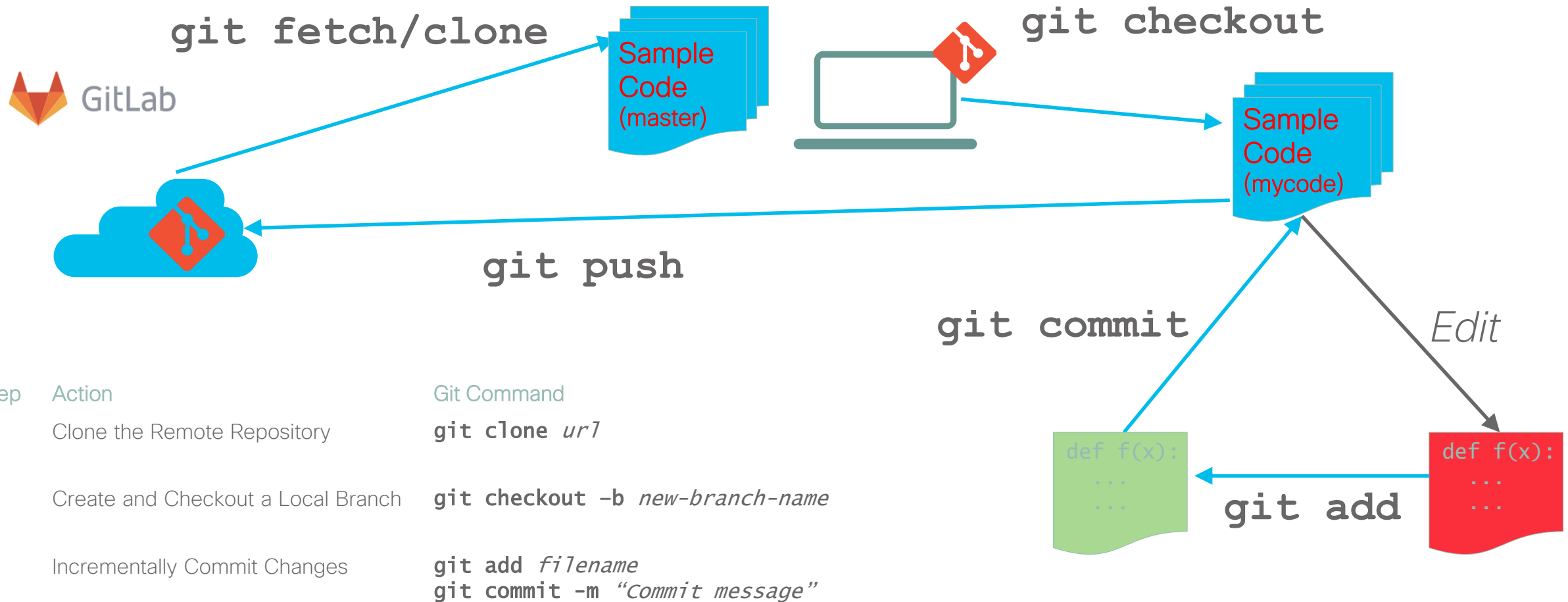


Image Source: <http://git-scm.com>

# Useful Git Commands

<b>Setup</b>	Tell git who you are <i>one-time setup</i>	<code>git config --global user.name "your name"</code> <code>git config --global user.email your@email.com</code>
<b>Clone</b>	Clone ("download") a git repository	<code>git clone url</code>
<b>Fetch</b>	Fetch branch(es)/tag(s)	<code>git fetch repository</code>
<b>Status</b>	Check the Status of your local repository	<code>git status</code>
<b>Checkout</b> A Branch	Create and Checkout a local <b>Branch</b> Creates a "safe place" for your changes	<code>git checkout -b new-branch-name</code>
<b>Add</b>	Add a file to your next commit.	<code>git add filename</code>
<b>Commit</b>	Commit your changes.	<code>git commit -m "Your commit message."</code>
<b>Push</b>	Push changes to remote (central server)	<code>git push</code>
<b>Merge</b>	Combine changes from branch(es)	<code>git merge</code>

# NetDevOps Sample-Code Workflow



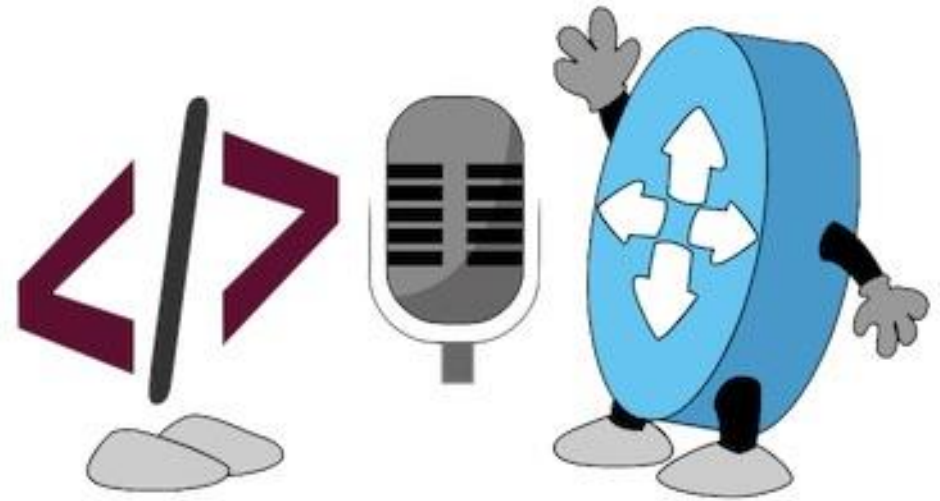
Step	Action	Git Command
1.	Clone the Remote Repository	<code>git clone url</code>
2.	Create and Checkout a Local Branch	<code>git checkout -b new-branch-name</code>
3.	Incrementally Commit Changes	<code>git add filename</code> <code>git commit -m "Commit message"</code>

Let's See This in Action!

Summing up

# What did we talk about?

- Where in the NetDevOps Toolbag are we?
- Where in the CI/CD Pipeline are we?
- Need for Version Control
- Git, by the book
- Demo Git flow using Gitlab





# Webinar Resource List

- Docs and Links
  - <https://developer.cisco.com/netdevops>
  - <https://git-scm.com>
  - <https://gitlab.com>
- Learning Labs
  - Introduction to Git - <http://cs.co/intro-to-git>
- DevNet Sandboxes
  - Multi-IOS Sandbox with CICD Demo <http://cs.co/sbx-multi>
- Code Samples
  - <http://cs.co/code-sbx-multi>

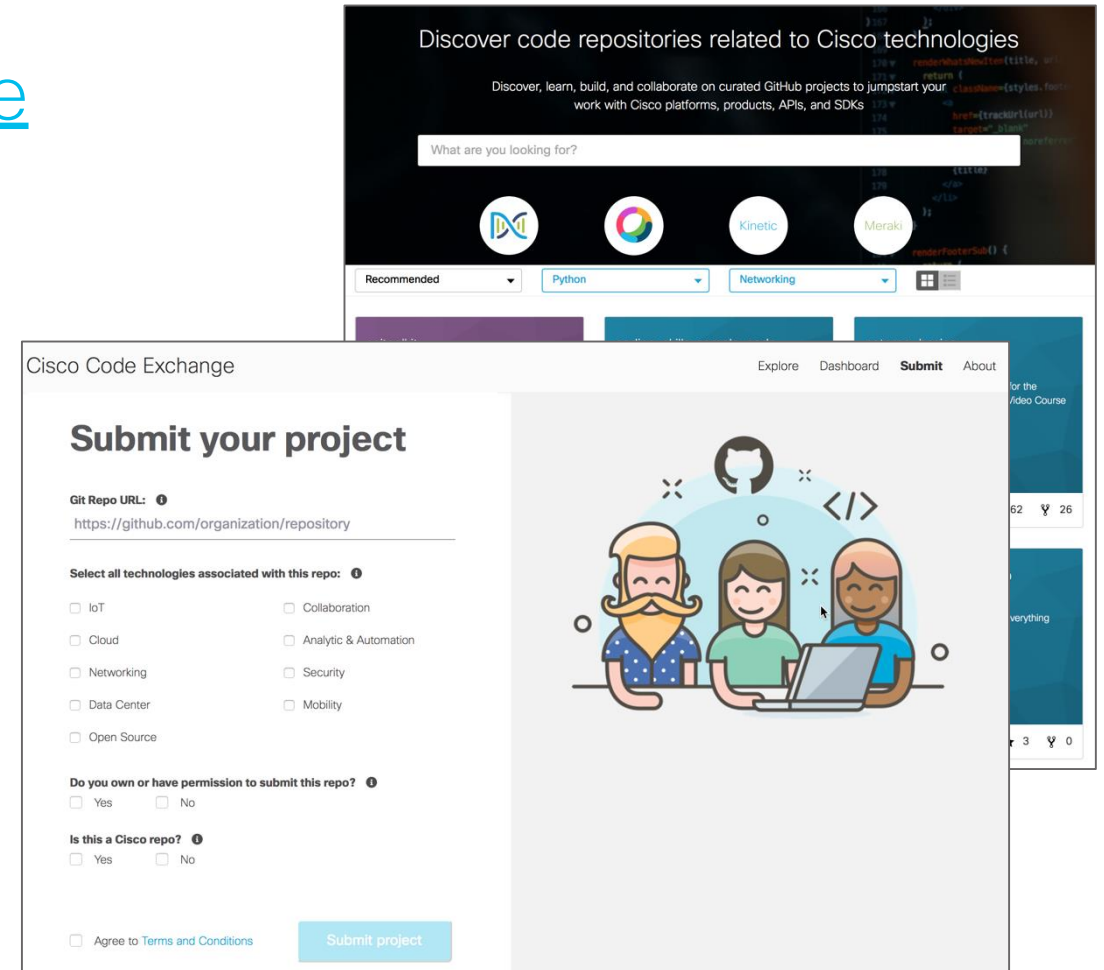


# NetDevOps Live! Code Exchange Challenge

[developer.cisco.com/codeexchange](https://developer.cisco.com/codeexchange)

***Open Issues, and Send PRs to sample repository to add some piece of configuration, or fix some error.***

*Example: Find a repository you're interested in, open a feature issue, work on the feature and submit a PR*



The image shows two overlapping screenshots of the Cisco Code Exchange website. The top screenshot displays a search interface with the heading "Discover code repositories related to Cisco technologies". It includes a search bar with the placeholder text "What are you looking for?", several technology icons (including Docker, Kubernetes, Kinetic, and Meraki), and filter dropdowns for "Recommended", "Python", and "Networking". The bottom screenshot shows the "Submit your project" form. The form includes a "Git Repo URL" field with the example "https://github.com/organization/repository", a section for selecting technologies (IoT, Cloud, Networking, Data Center, Open Source, Collaboration, Analytic & Automation, Security, Mobility), a "Do you own or have permission to submit this repo?" section, and an "Is this a Cisco repo?" section. At the bottom, there is a checkbox for "Agree to Terms and Conditions" and a "Submit project" button. To the right of the form is an illustration of three people (two men and one woman) working together at a laptop, with a GitHub logo and code symbols above them.

# Looking for more about NetDevOps?

- NetDevOps on DevNet  
[developer.cisco.com/netdevops](https://developer.cisco.com/netdevops)
- NetDevOps Live!  
[developer.cisco.com/netdevops/live](https://developer.cisco.com/netdevops/live)
- NetDevOps Blogs  
[blogs.cisco.com/tag/netdevops](https://blogs.cisco.com/tag/netdevops)
- Network Programmability Basics Video Course  
[developer.cisco.com/video/net-prog-basics/](https://developer.cisco.com/video/net-prog-basics/)



Got more questions? Stay in touch!



**Matt DeNapoli**

 [mdenapol@cisco.com](mailto:mdenapol@cisco.com)

 [@theDeNap](https://twitter.com/theDeNap)

 <http://github.com/denapom11>



**[developer.cisco.com](https://developer.cisco.com)**

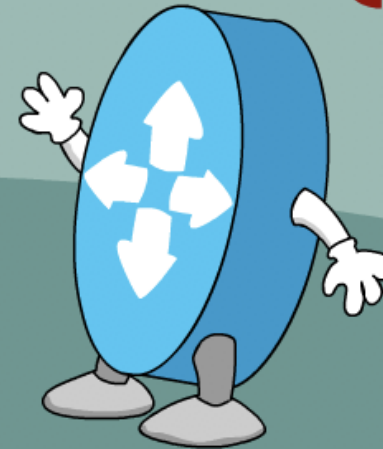
 [@CiscoDevNet](https://twitter.com/CiscoDevNet)

 [facebook.com/ciscocodevnet/](https://facebook.com/ciscocodevnet/)

 <http://github.com/CiscoDevNet>



# NETDEVOPS {LIVE!}



DEVNET

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

@netdevopslive 