# Integrating Ansible with GitHub Actions

## Introducing GitHub Actions

**Henry Been**

Independent DevOps & Azure Architect

@henry_been www.henrybeen.nl

# Limitations of Running Ansible Locally

Sharing playbooks becomes hard

Versions are difficult to track

You need access to all servers or vaults

Running playbooks on a schedule is not possible

# Benefits of Deployment Pipelines

Source control capabilities

Executing playbooks from a dedicated runner

Running playbooks on change

Automated linting and testing

# Overview

**Introducing the Use Case**

**Working with GitHub source control**

- Repositories
- Committing and pushing

**Building pipelines with GitHub Actions**

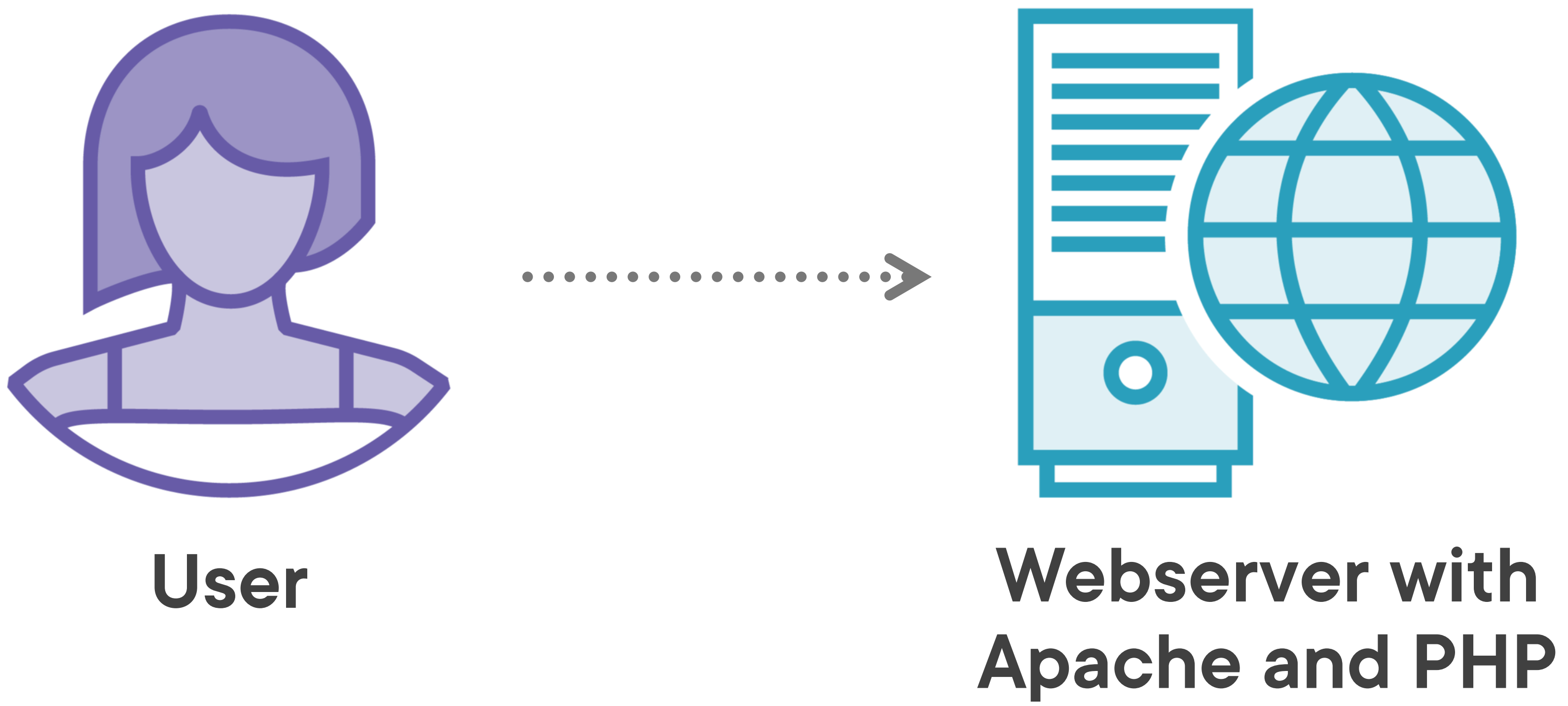
- Actions
- Triggers
- Workflows

**Implement playbook validation**

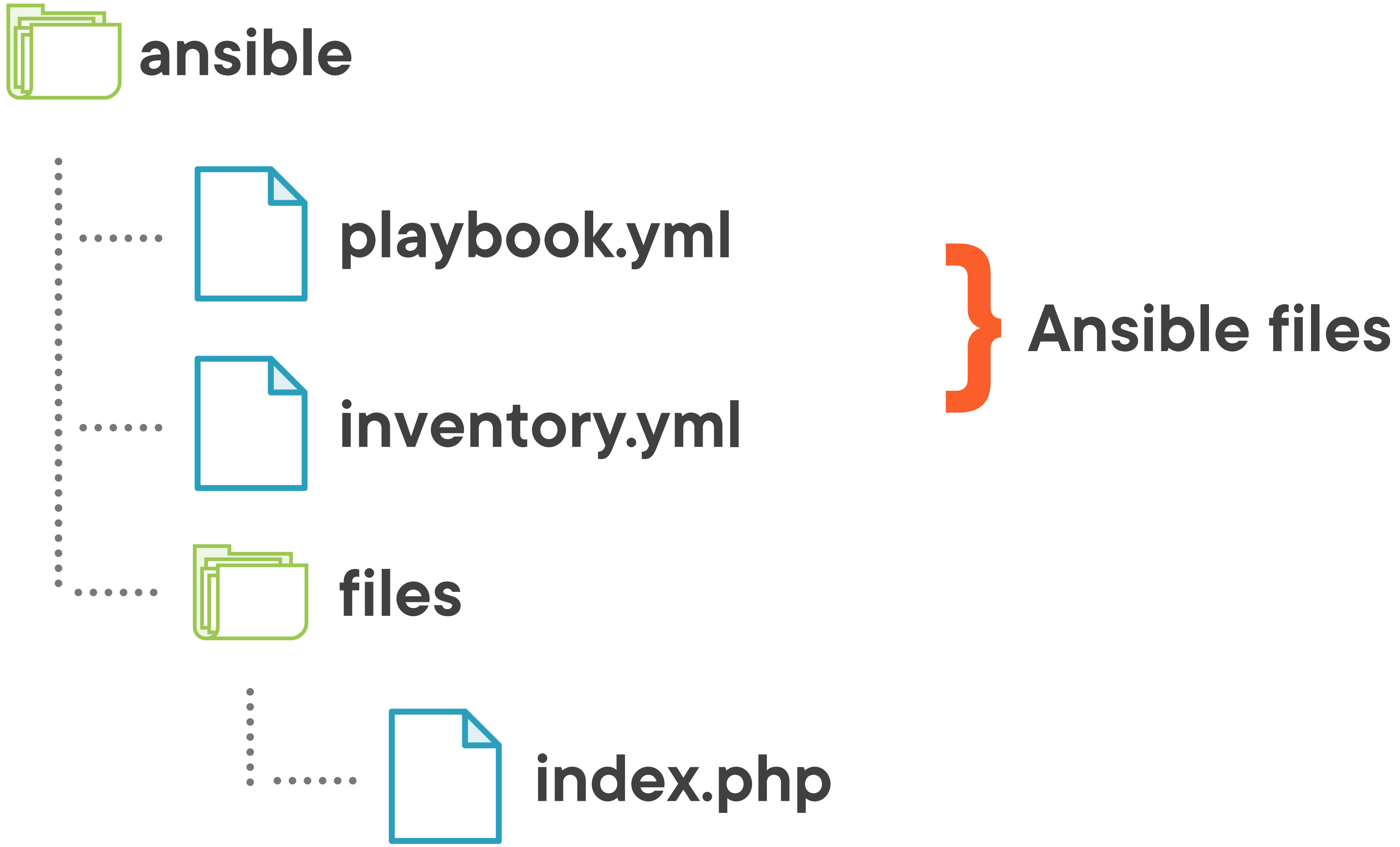
- Automatically on every change
- Mandatory before merging a pull request

# Use Case: Configuring a Linux, Apache, and PHP Server

# Example Architecture



# Example Ansible Files



ansible

playbook.yml

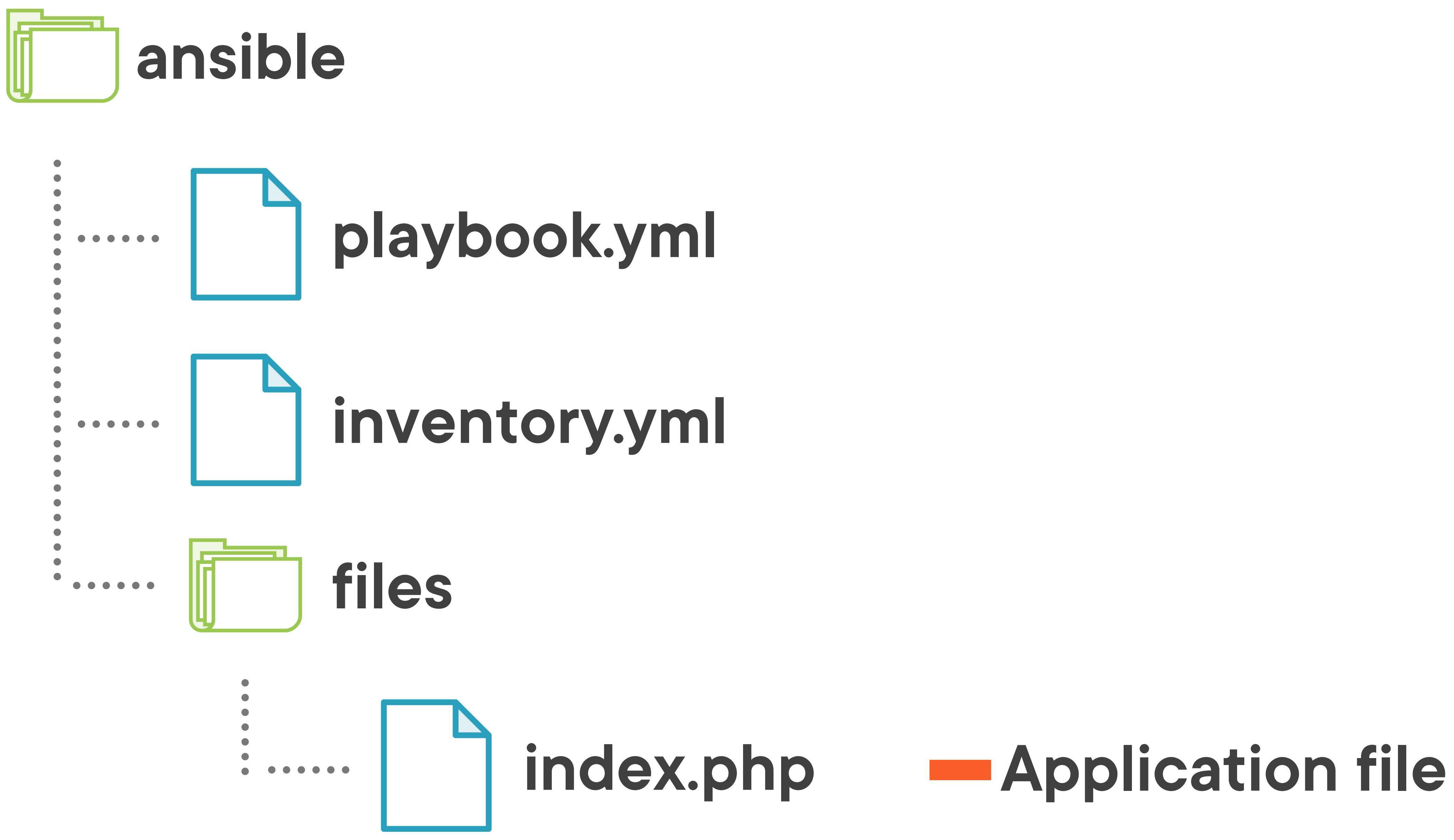
inventory.yml

files

index.php

Ansible files

# Example Ansible Files



ansible

playbook.yml

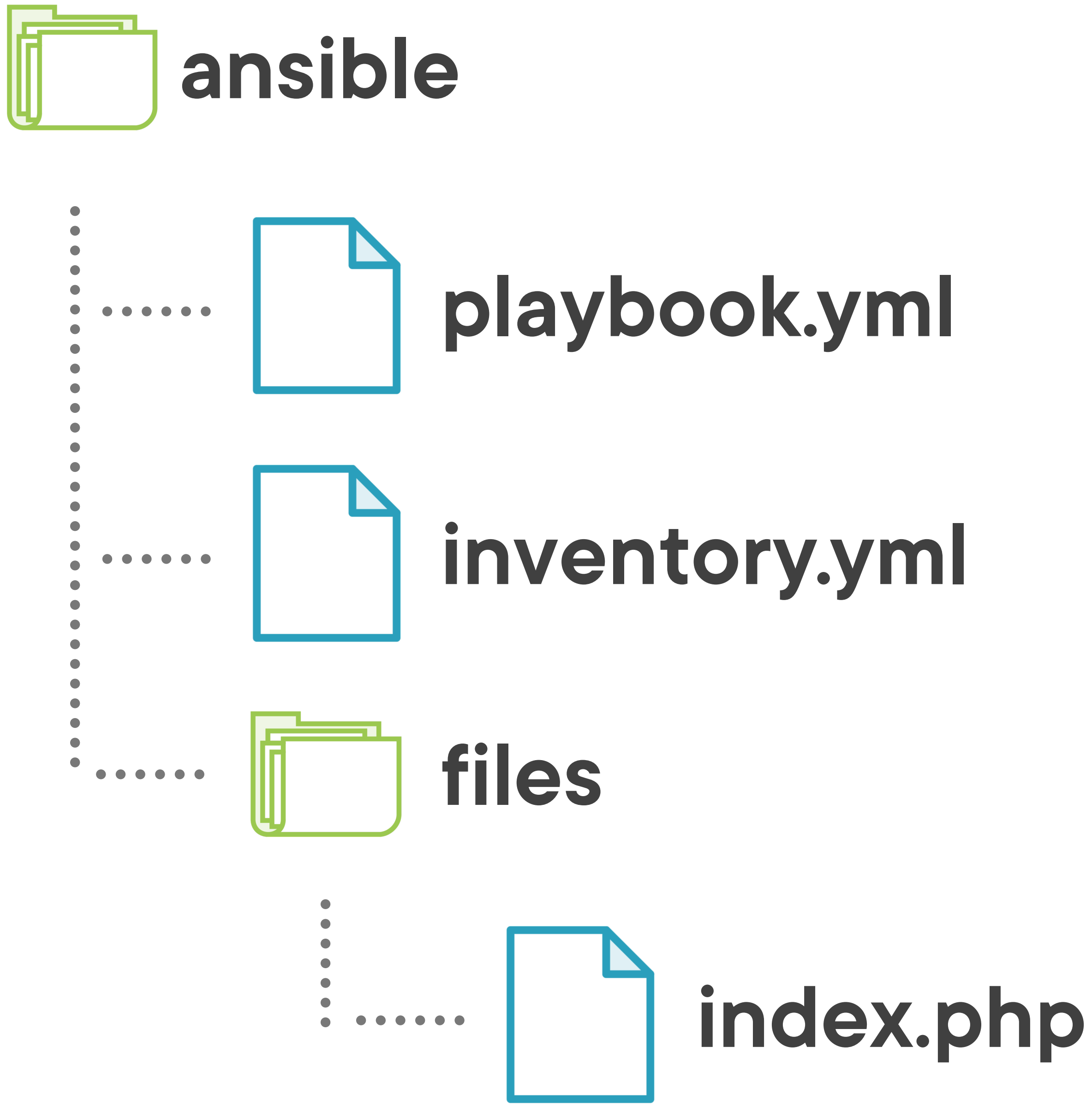
inventory.yml

files

index.php

Application file

# Example Ansible Files



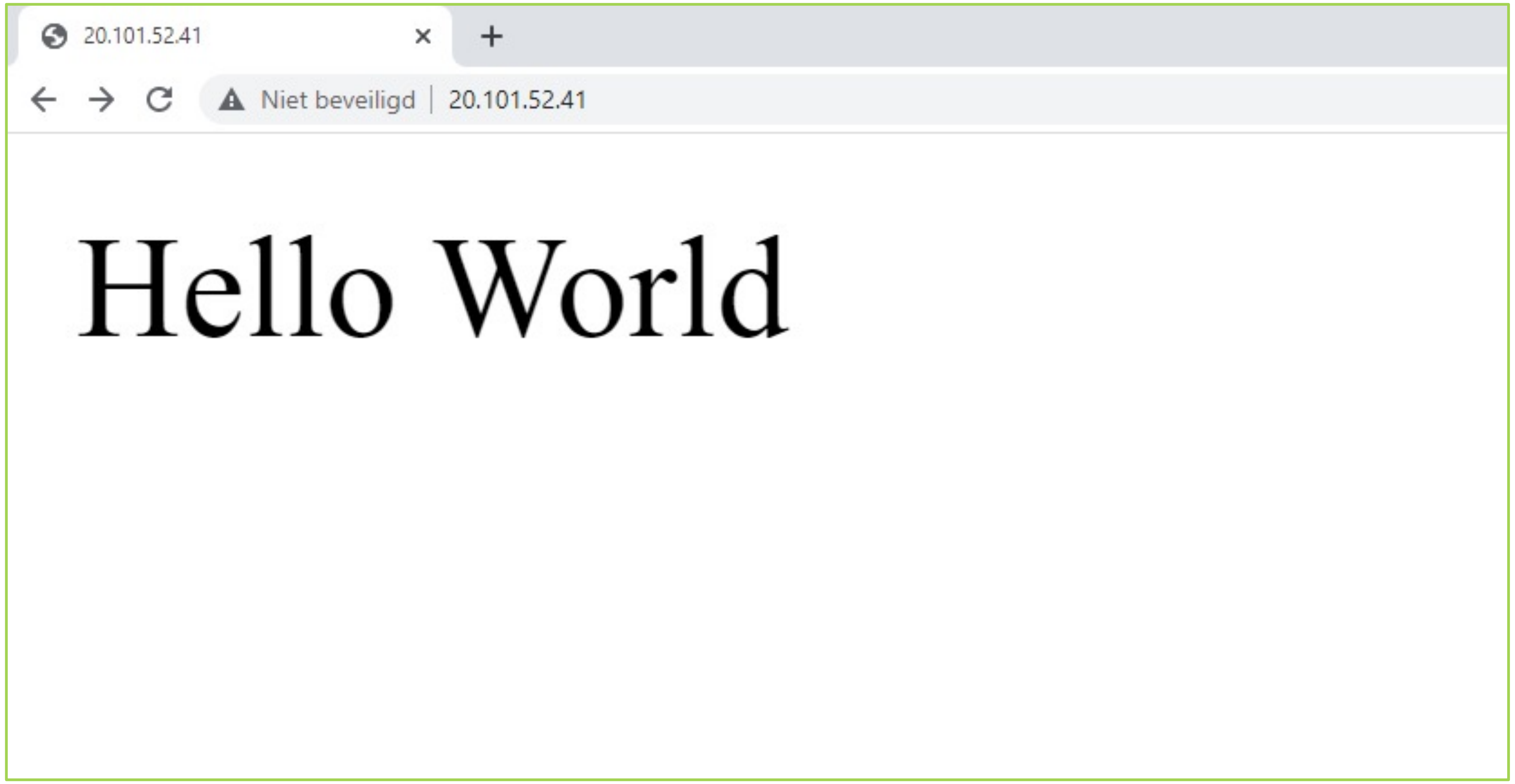
ansible

playbook.yml

inventory.yml

files

index.php



Hello World

# The Reasons for Source Control

# The Problems of Local Source Code

**No versioning**

- Rolling back to a previous version is hard or impossible

**Overwritten changes**

- Developers making changes in the same files, overwriting each others changes

# Benefits of Source Control

Full backup and history of all changes

Safe experimentation and change preparation

Traceability and accountability

Easy collaboration

# But most importantly…

A single source of truth

A centralized location for enforcing standards

A single starting point for deployment

# Demo

Storing Playbooks in GitHub

Download and install Git

https://git-scm.com/downloads

# Demo

**Creating a GitHub repository**

- Creating an account
- Creating the repository
- Initializing the repository

**Working with Git**

- Cloning the repository
- Making and committing changes
- Pushing the changes

# More information

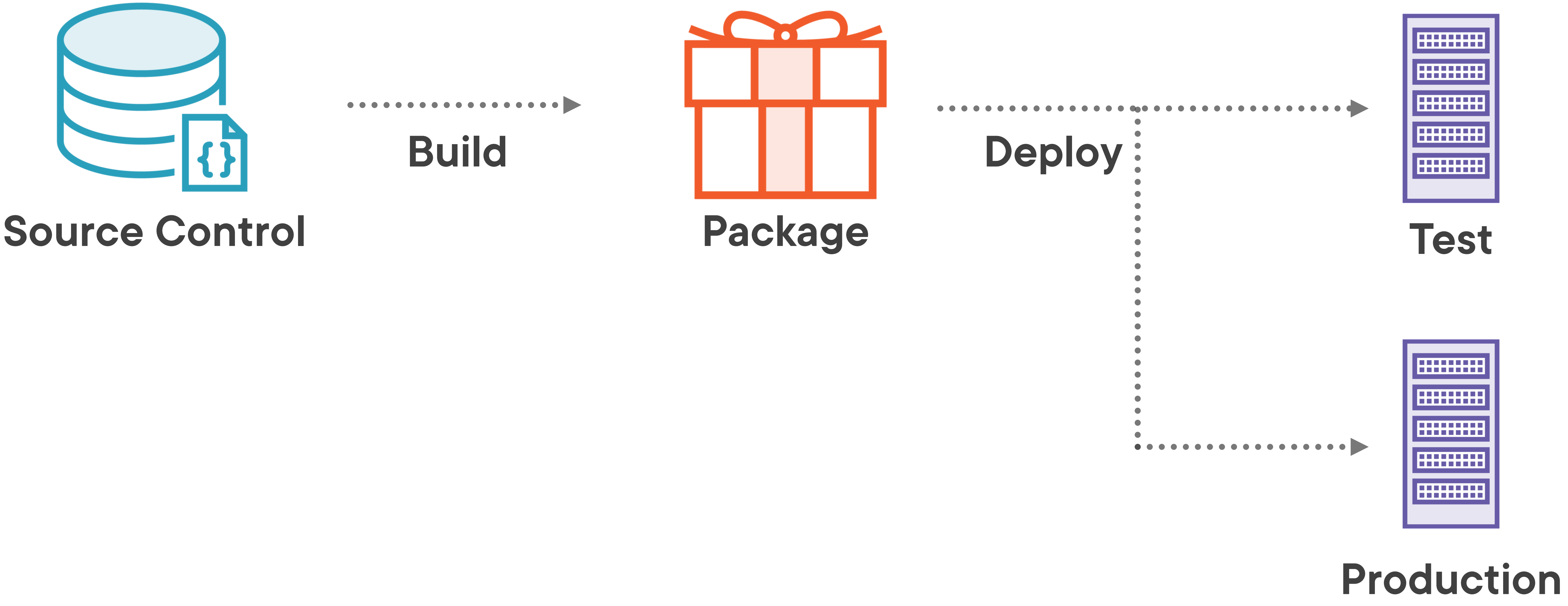
## GitHub: Getting Started

Gill Cleeren

# GitHub Actions: Workflows, Actions, and Triggers

A single source of truth as the starting point for validating and deploying Ansible Playbooks.

# GitHub Actions



Source Control

Build

Package

Deploy

Test

Production

# Continuous …

## Integration

Fast and frequent integration of changes to verify they work correctly together

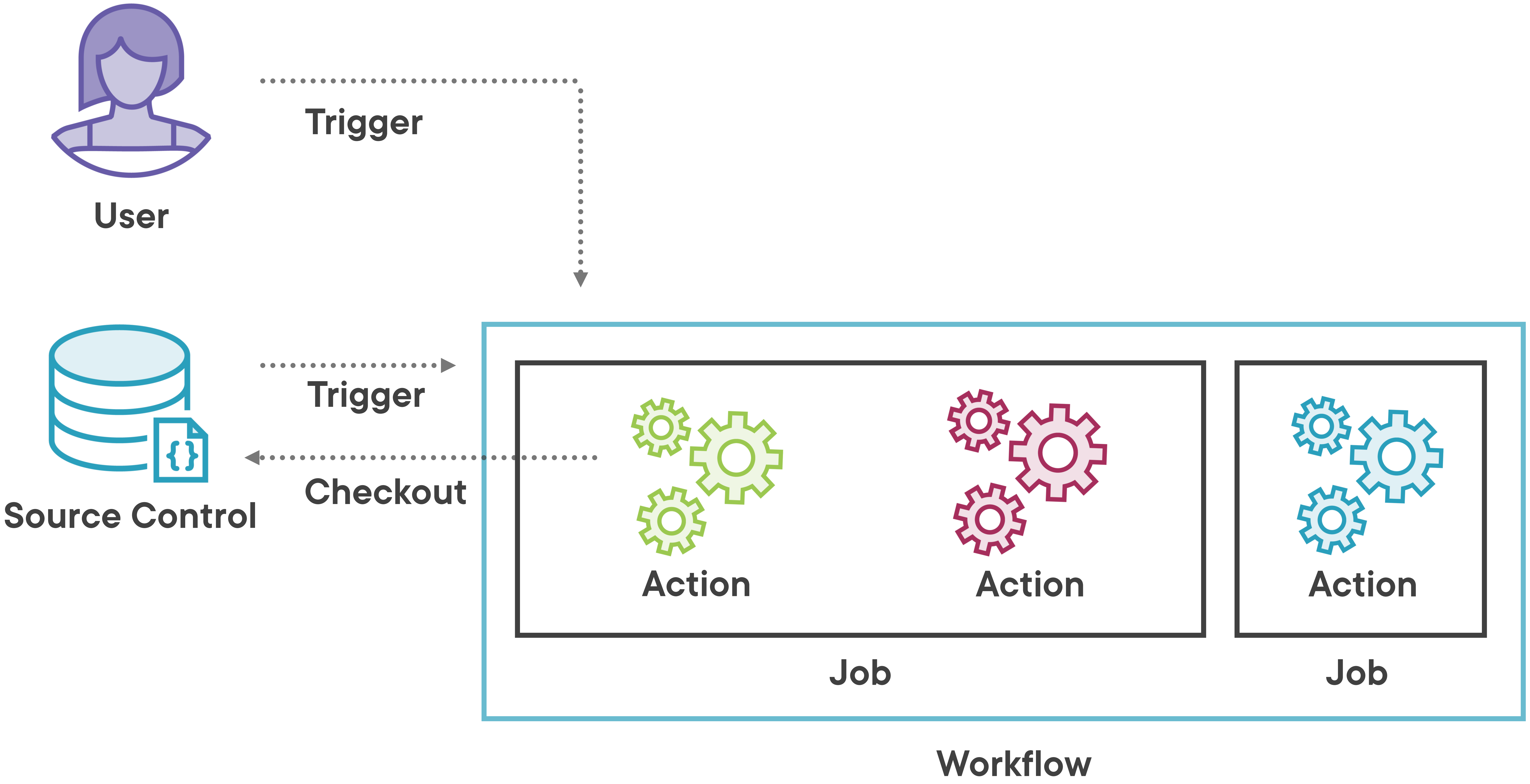
## Delivery

Every accepted change leads to the automated build of a deployable package

## Deployment

Every accepted change is automatically deployed

# GitHub Actions



# Available Triggers

**Manually**

**On code events**

- On push or pull request
- Around 30 other event types
- Optionally with branch filters

**On a schedule**

# Available Actions

Checkout a repository

Invoke a script

Compile, test, or deploy code

Lint and run Ansible files

Over 4000 others

You can write your own

# More information

## Getting Started with GitHub Actions

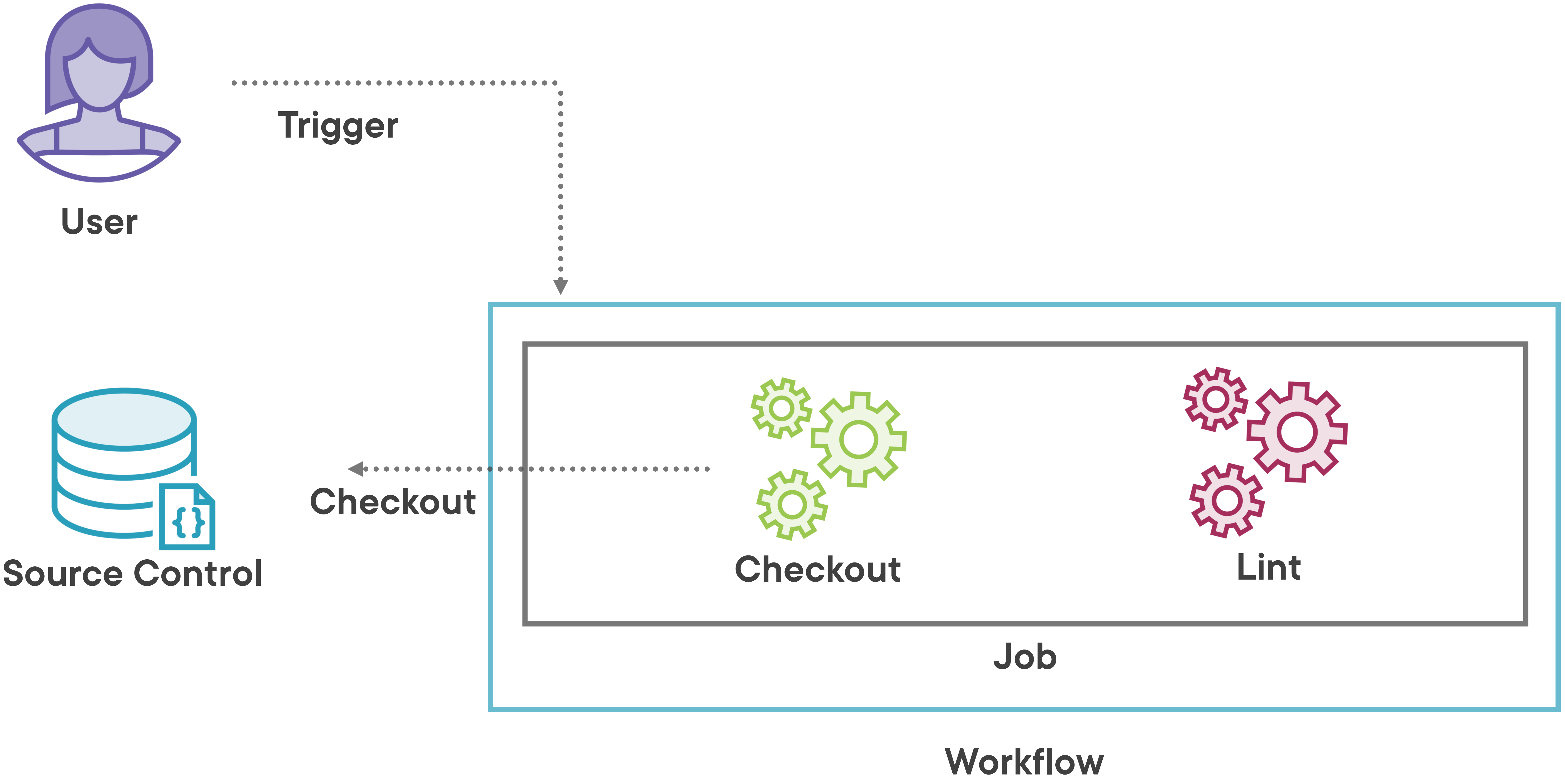
Aaron Stewart

Demo

## A Workflow for Linting Ansible Playbooks

Linting: Checking files to be well-formed, adhering to standards, and not using error-prone approaches

# A Linting Workflow



# Demo

**Creating a linting workflow**

- Writing a minimal workflow
- Adding the linting action

**Fixing linting issues**

- Ignoring unwanted errors
- Fixing the version of PHP to install

# Running the Workflow on Every Push and Pull Request

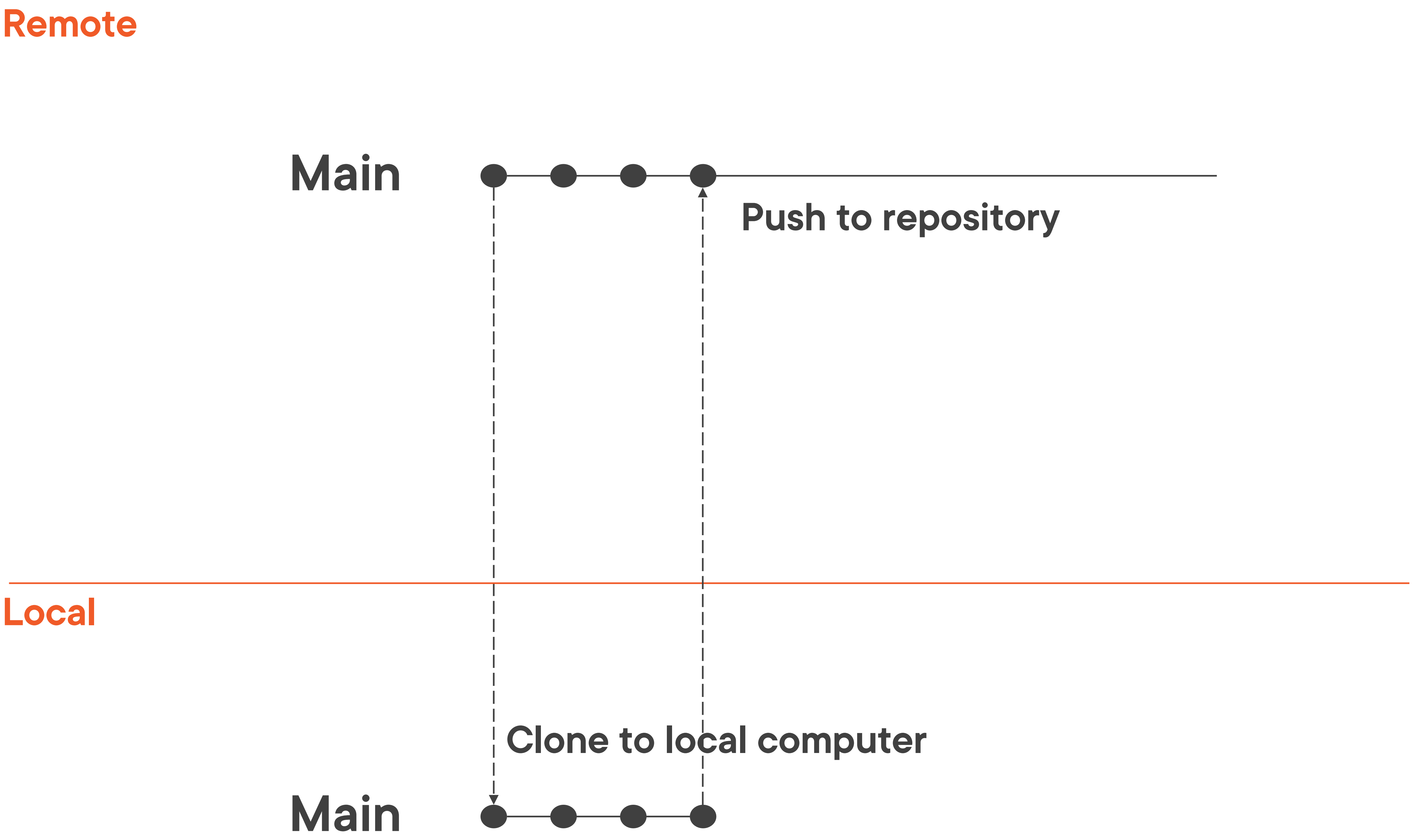
Demo

## Updating the workflow triggers

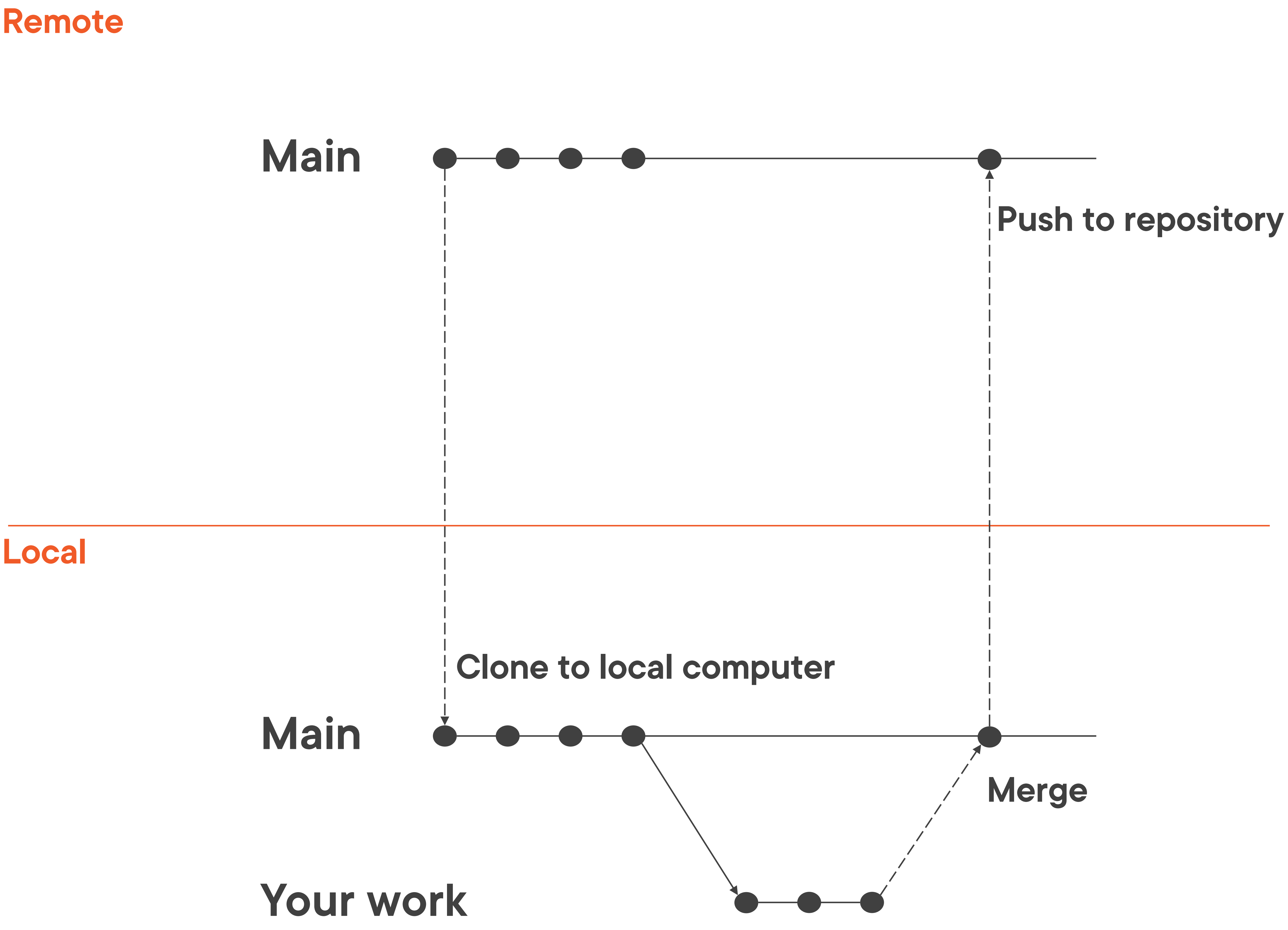
- To run on every push

## Branch Protection

To enforce the use of pull requests

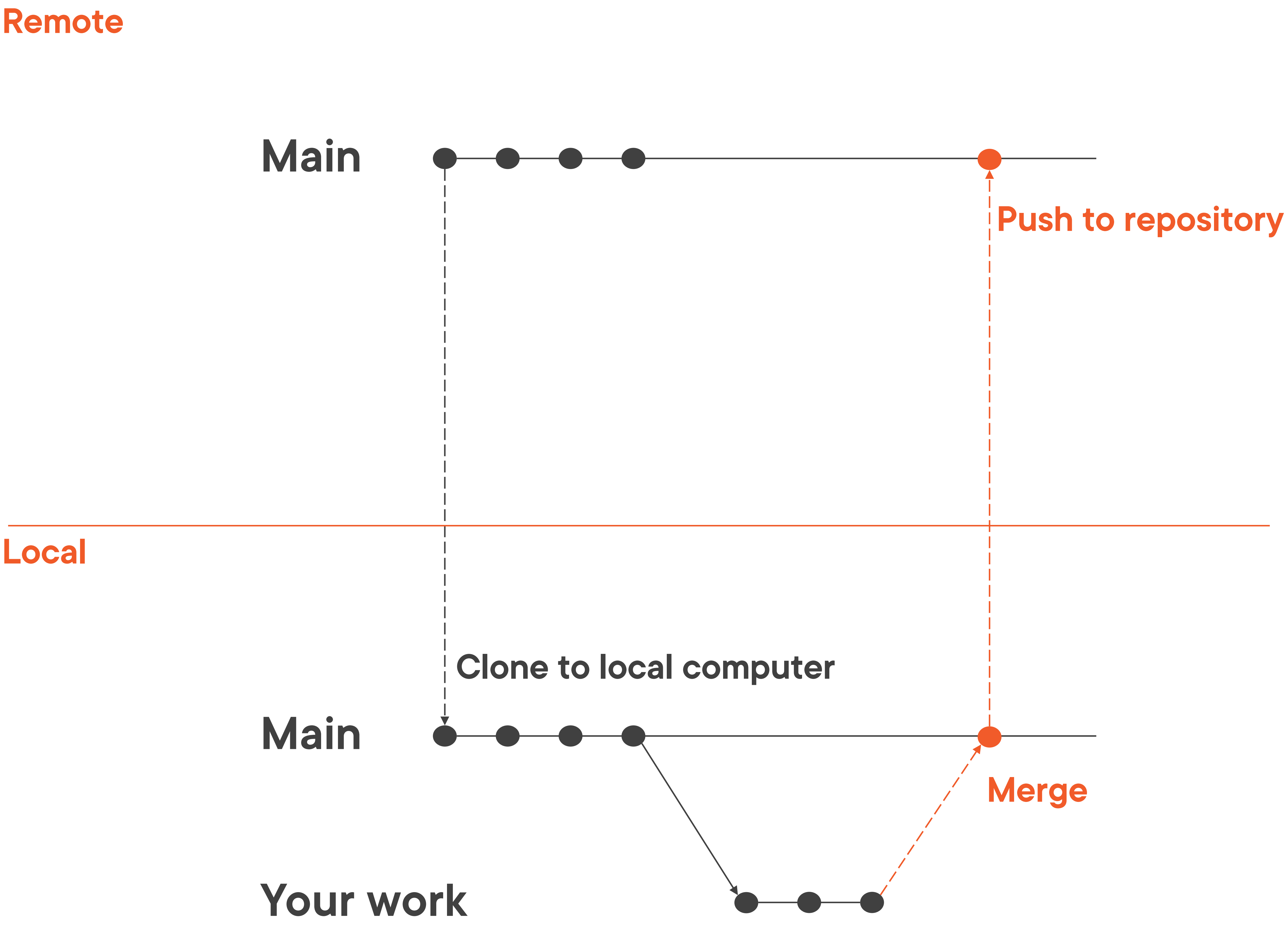


**Branch Protection**
To enforce the use of pull requests

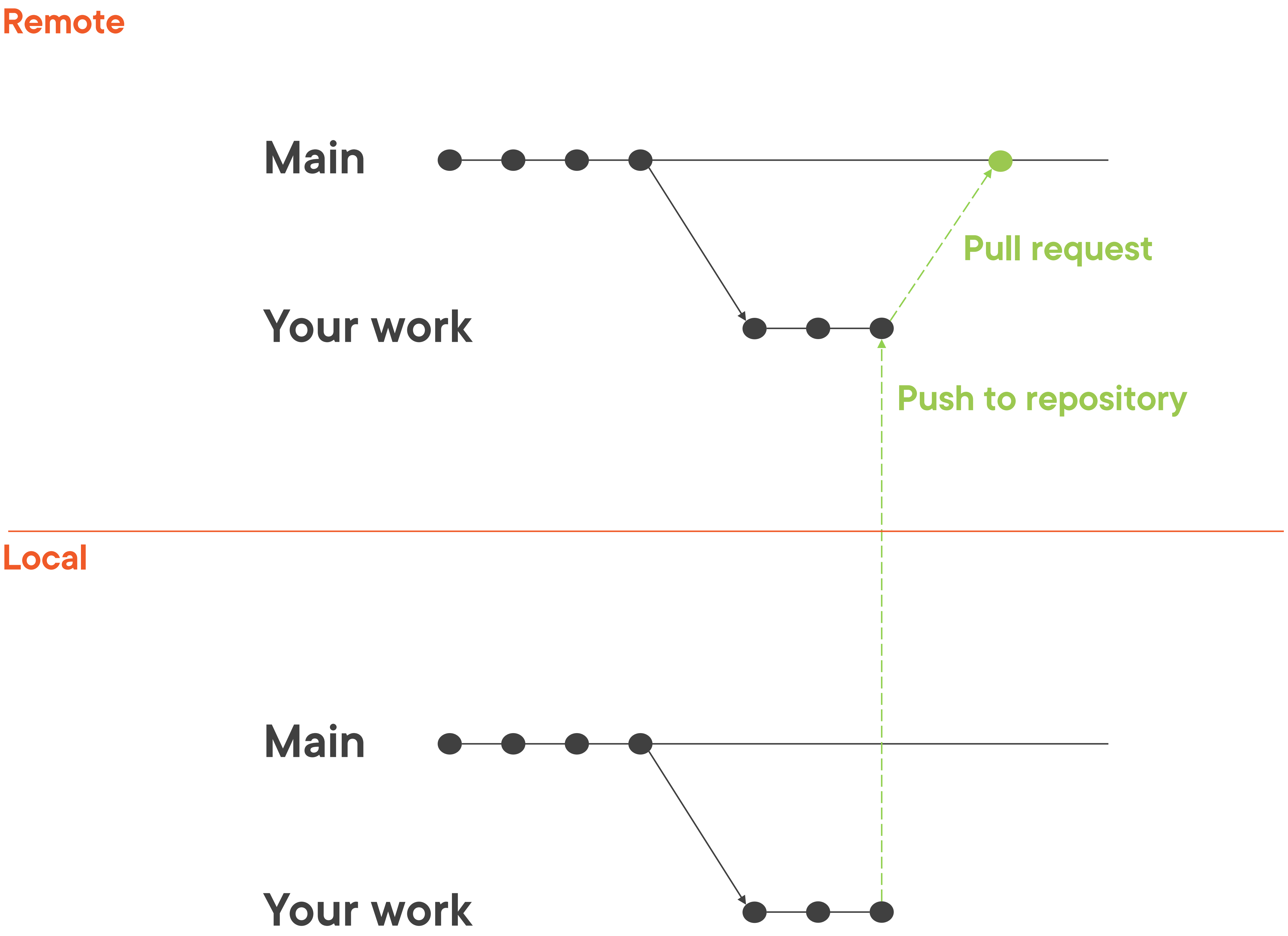


## Branch Protection

To enforce the use of pull requests



**Branch Protection**
To enforce the use
of pull requests



# Demo

**Exploring pull requests**

**Updating the workflow triggers**

- To run on every PR

**Blocking PRs when a workflow fails**

# Summary

# Summary

**GitHub Code**

- Source control
- Peer-reviews through pull requests

**GitHub Actions**

- Workflows, Triggers, Jobs, Actions
- To automatically lint Ansible files
- Block pull requests when linting fails

# Up Next:
# Running a Playbook Using GitHub Actions