



Building a CI/CD Pipeline

Byron Sommardahl

This Segment

01

This vs That

CI vs CD vs
Continuous Delivery

02

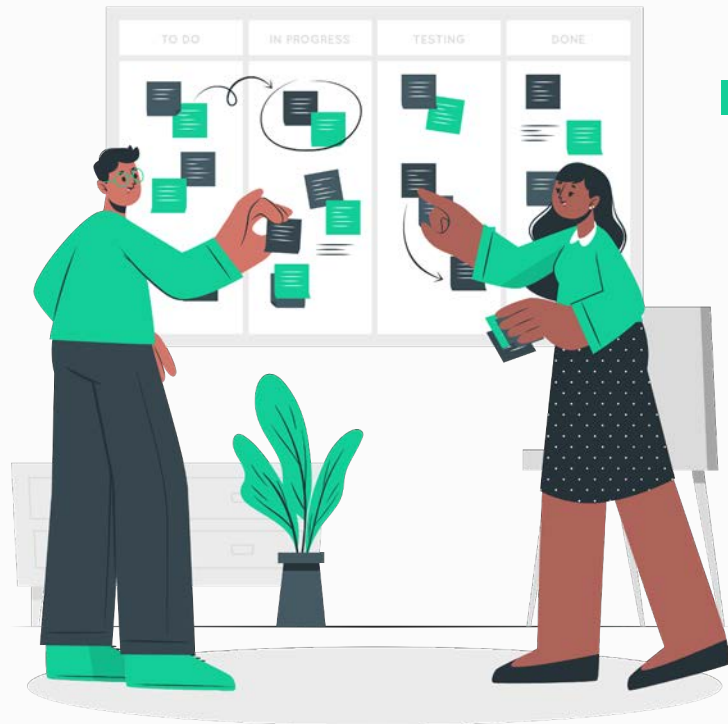
A New Direction

Continuous delivery as a
direction

03

Influential Metrics

How to influence culture with
metrics



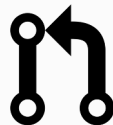
Continuous Delivery



Coding



Code Review



Merge/
Integrate



Compile/Lint



Run Unit
Tests



Run
Integration
Tests

Continuous Delivery



Verify
Dependency
Security



Deploy to
Test
Environment



Team Test



Deploy to
Client Test
Environment

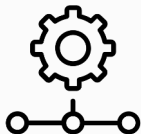


Client Test



Create
Infrastructure

Continuous Delivery



Configure
Infrastructure



Deploy to
Production



Smoke Test
in Production



Rollbacks

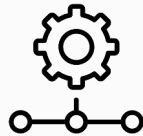
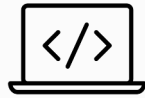


Promoting to
Production



Celebration

Continuous Delivery



Continuous Everything!



Continuous
Delivery



Continuous
Integration



Continuous
Deployment

Continuous Everything!



Continuous
Delivery



Continuous
Integration

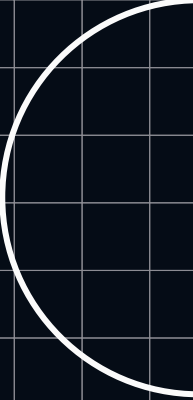


Continuous
Deployment



Continuous Delivery

< < < < < <



> > > > > >



Continuous Delivery



Continuous Integration



Continuous Delivery



Continuous
Integration



Continuous
Deployment



Continuous Delivery



Continuous
Integration



Continuous
Deployment



Continuous Delivery



Continuous
Integration



Development
Process



Continuous
Deployment



Toyota Manufacturing

Continuous improvement is a process, continuously and intentionally moving in the direction of “True North”.



Toyota Manufacturing

Continuous improvement is a process,
continuously and intentionally moving
in the direction of “True North”.

About the

Project

Influencing Culture

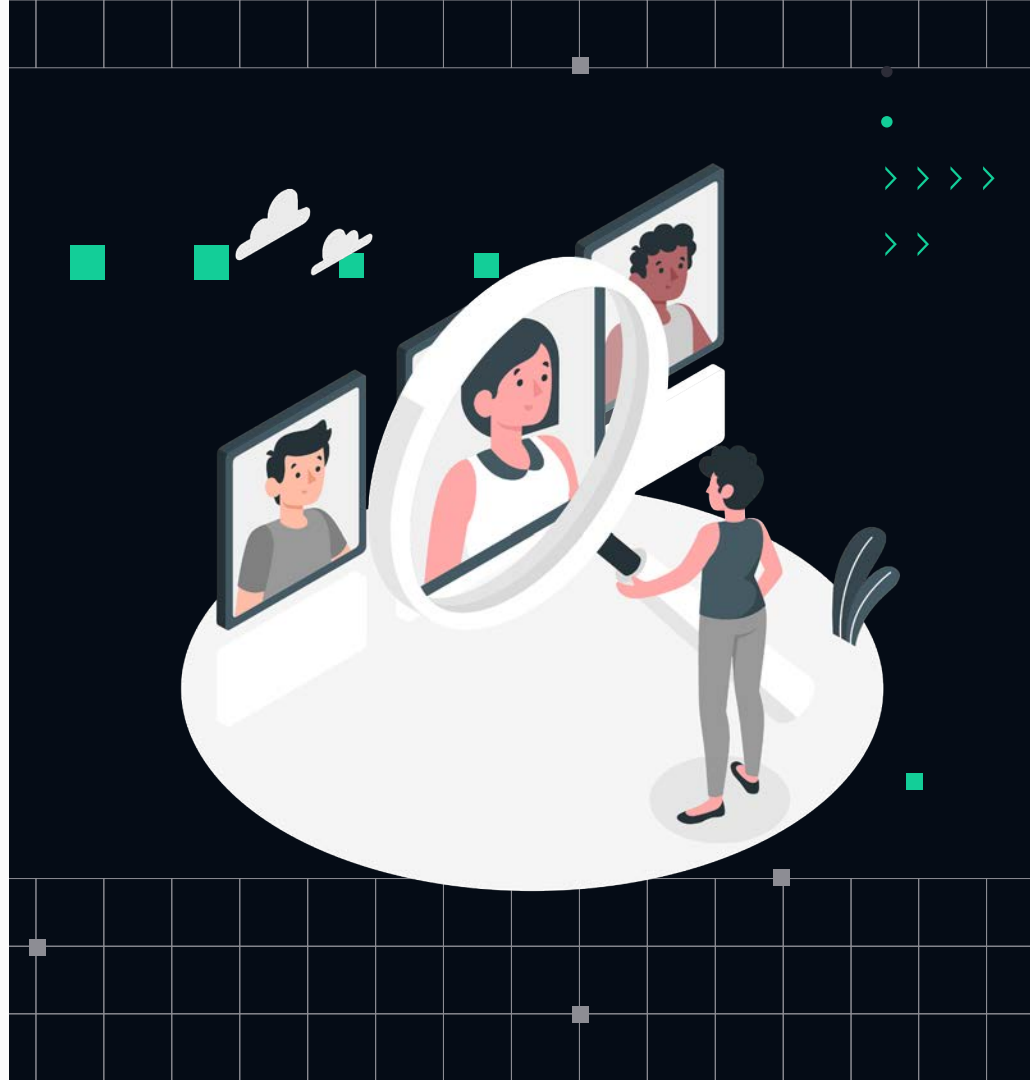
10

1 million tiny, consistent, intentional pushes.



“Tell me how you
measure me and I will
tell you how I will
behave.”

- Eli Goldratt





Common Metrics: > > > >

- Mean Time to Recovery (MTTR)
- Lead time
- Delivery rate
- Cycle Time
- Test Coverage
- Bug Escape Rate
- Etc

So what?? Who cares?



Tech People



Business People

So what?? Who cares?



Tech People

- Quality of life
- Pride in work
- Desire to help organization



Business People

So what?? Who cares?



Tech People

- Quality of life
- Pride in work
- Desire to help organization



Business People

- Better bottom-line
- Competitive in market
- Less turn-over

Business Value

Increase Revenue

Increased profits, more sales, new features.



Reduce Costs

Lower server costs, less manpower.



Protect Revenue

Keep revenue flowing, preserve customer engagement.



Avoid Costs

Reduce costs, cut expenses.



> >

> > > >

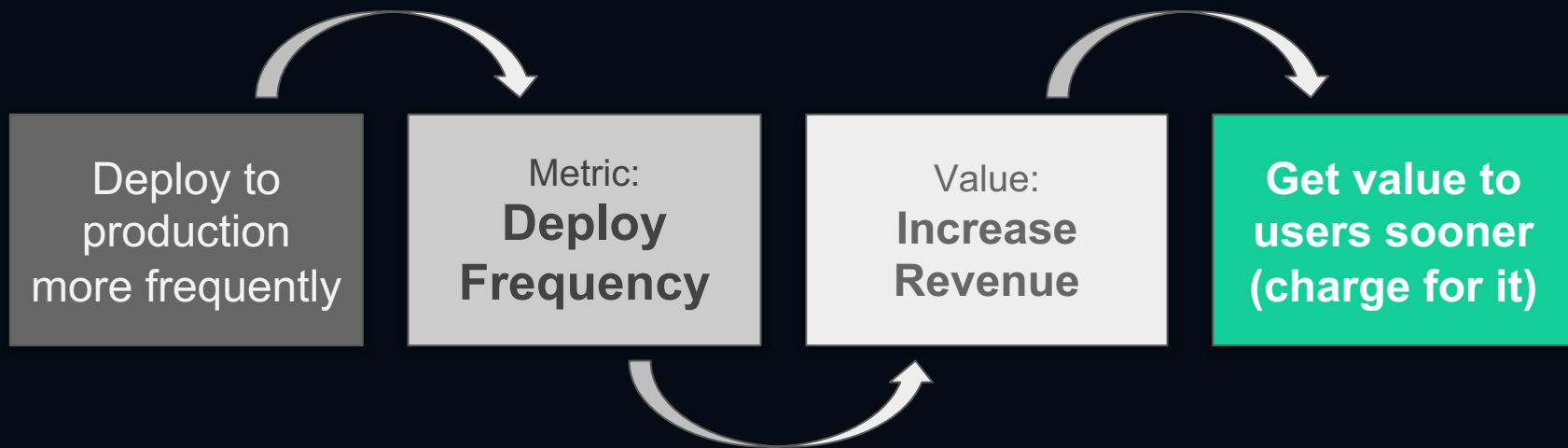
Influential Metrics



Influential Metrics



Influential Metrics



Influential Metrics



Influential Metrics





bit.ly/devops-quiz

Take the quiz. See how your organization measures up!



Student Poll

According to the “DORA DevOps Quick Check” (bit.ly/devops-quiz), how does your organization rank?

- A. Low
- B. Medium
- C. High
- D. Elite

10 Minute Break



Building a CI/CD Pipeline

Byron Sommardahl

This Segment

01

Benign Scripts

Maintainable scripts that “do no harm”.

02

YAML Notation

How to write value YAML.

03

Vendor Syntax

Navigating the differences.



Benign Scripts

A necessary evil, but don't necessarily
need to be evil.



```
{
  "name": "codex-academy-website",
  "version": "0.1.0",
  "private": true,
  "scripts": {
    "serve": "cross-env NODE_ENV=development vue-cli-service serve",
    "build": "vue-cli-service build",
    "test:unit": "vue-cli-service test:unit",
    "lint": "vue-cli-service lint",
    "deploy": "gh-pages -d dist",
    "test": "jest --coverage",
    "predeploy": "cross-env NODE_ENV=production npm run build",
    "storybook:build": "vue-cli-service storybook:build -c config/storybook",
    "storybook": "vue-cli-service storybook:serve -p 6006 -c config/storybook",
    "smoke": "ts-node e2e/smoke.e2e.ts"
  }
}
```



Benign Scripting

Script Files

Slice scripts into logical groups using files.



> >

> > > >

Benign Scripting

Script Files

Slice scripts into logical groups using files.



One Language

Choose a language and stick with it to keep your team together.

> >

> > > >

Benign Scripting

Script Files

Slice scripts into logical groups using files.



One Language

Choose a language and stick with it to keep your team together.

SRP

Do one thing and do it really well!



> >

> > > >

Benign Scripting

Script Files

Slice scripts into logical groups using files.



One Language

Choose a language and stick with it to keep your team together.

SRP

Do one thing and do it really well!



Fail Fast, Fail Hard

Embrace failure and bring it to the forefront.

> >

> > > >




YAML?

“Human-readable data-serialization language”



YAML Spacing



```
image: "node:lts-alpine"
stages:
  - build
  - test
  - deploy
# These folders are cached between builds
cache:
  key: some value
  paths:
    - node_modules/
    - .cache/
    - public/
```


YAML Keys

```
image: "node:lts-alpine"
```

```
stages:
```

```
  - build
```

```
  - test
```

```
  - deploy
```

```
# These folders are cached between builds
```

```
cache:
```

```
  key: some value
```


```
  paths:
```

```
    - node_modules/
```

```
    - .cache/
```

```
    - public/
```

YAML Comments




```
image: "node:lts-alpine"
-
stages:
  - build
  - test
  - deploy
-
# These folders are cached between builds
cache:
  key: some value
  paths:
    - node_modules/
    - .cache/
    - public/
```

```
employees:
  - martin:
      name: Martin D'vloper
      job: Developer and Master of the Universe
      skills:
        - python
        - perl
        - pascal
  - tabitha:
      name: Tabitha Bitumen Jones
      job: Developer
      skills:
        - lisp
        - fortran
        - erlang
```

YAML Scalar

YAML Dictionaries

```
employees:
  - martin:
      name: Martin D'vloper
      job: Developer and Master of the Universe
      skills:
        - python
        - perl
        - pascal
  - tabitha:
      name: Tabitha Bitumen Jones
      job: Developer
      skills:
        - lisp
        - fortran
        - erlang
```



```
image: "node:lts-alpine"
```

```
stages:
```

```
  - build
```

```
  - test
```

```
  - deploy
```

```
# These folders are cached between builds
```

```
cache:
```

```
  - key: some value
```

```
  - paths:
```

```
    - node_modules/
```

```
    - .cache/
```

```
    - public/
```



YAML Collections

YAML

Multi-Line

```
formatted_text: | (pipe)
.....These three lines of text
.....form the value of this key,
.....including the new lines.
-
unformatted_text: > (greater-than)
.....These five lines of text
.....form the value of this key
.....but does not include the new
.....lines. Instead it's really
.....one long line of text.
-
```



YAML?

“Human-readable data-
serialization language”



CI/CD Solutions Abound



GitHub Actions

Leading cloud-based git repository provider with added CI/CD functionality with “Actions”.



GitLab CI/CD

Open-source, community-driven git repository provider with built-in CI/CD facilities.



CircleCI

Popular cloud-based CI/CD provider. Requires 3rd party git provider.



CI/CD Solutions Solve the Same Problems

- Triggering
- Importing Code
- Running Scripts
- Raising Alerts

> >
> > >
>
>
>
> > >
>
>
>



CI/CD Solutions Abound



GitHub Actions

Leading cloud-based git repository provider with added CI/CD functionality with “Actions”.



GitLab CI/CD

Open-source, community-driven git repository provider with built-in CI/CD facilities.



CircleCI

Popular cloud-based CI/CD provider. Requires 3rd party git provider.



bit.ly/github-actions-quickstart

Workflows Folder



```
byron:~/my-git-project$ cd .github/workflows  
byron:~/my-git-project/.github/workflows$ ls  
github-actions-demo.yml
```

```
name: GitHub Actions Demo
on: [push]
jobs:
  Explore-GitHub-Actions:
    runs-on: ubuntu-latest
    steps:
      - run: echo "🎉 The job was automatically triggered by a ${github.event_name} event."
      - run: echo "👋 This job is now running on a ${runner.os} server hosted by GitHub!"
      - run: echo "🔍 The name of your branch is ${github.ref} and your repository is ${github.repository}."
      - name: Check out repository code
        uses: actions/checkout@v2
      - run: echo "💡 The ${github.repository} repository has been cloned to the runner."
      - run: echo "🖥️ The workflow is now ready to test your code on the runner."
      - name: List files in the repository
        run: |
          ls ${github.workspace}
      - run: echo "🍏 This job's status is ${job.status}."
```



```
name: GitHub Actions Demo
```

```
on: [push]
```

```
jobs:
```

```
  Explore-GitHub-Actions:
```

```
    runs-on: ubuntu-latest
```

```
    steps:
```

- run: echo "🎉 The job was automatically triggered by you. It looks like you're working on `github.repository`.""
- run: echo "🐧 This job is now running on a `{{ runner.os }}` operating system, powered by GitHub Actions."
- run: echo "🔍 The name of your branch is `{{ github.ref_name }}`.""
- name: Check out repository code
- uses: actions/checkout@v2
- run: echo "👉 The `{{ github.repository }}` repository was cloned with this action."



`name: GitHub Actions Demo`

`on: [push]`

`jobs:`

`Explore-GitHub-Actions:`

`runs-on: ubuntu-latest`

`steps:`

- `- run: echo "🎉 The job was automatically triggered by you. It looks like you're working on GitHub Actions and that's awesome! :tada: More to it later!"`
- `- run: echo "🐧 This job is now running on a ${{ runner.os }} machine hosted by GitHub Actions, in ${{ github.repository }} repository!"`
- `- run: echo "🔍 The name of your branch is ${{ github.ref }} and your workflow is ${{ github.workflow }}."`
- `- name: Check out repository code`
 `uses: actions/checkout@v2`
 `run: echo "📄 The ${{ github.repository }} repository has been cloned to the runner."`

on: [push]

jobs:

Explore-GitHub-Actions:

```
runs-on: ubuntu-latest
```

step:

```
on: [push, pull_request]
```

```
github.repository_id: [REDACTED]
```

- **name**: Check out repository code

```
uses: actions/checkout@v2
```

```
run: echo "👉 The $[[ github_repository ]] repository"
```




name: GitHub Actions Demo

on: [push]

jobs:

Explore-GitHub-Actions:

runs-on: ubuntu-latest

steps:

- run: echo "🎉 The job was automatically triggered by you. It looks like you're working on GitHub Actions and that is awesome! It looks like you're really trying to learn about GitHub Actions, so here are some of the things you can do with GitHub Actions:
- run: echo "🐧 This job is now running on a \${{ runner.os }} Ubuntu runner. Here is some information about the environment you're running on:
- run: echo "🔍 The name of your branch is \${{ github.ref }} and your repository is \${{ github.repository }}."
- name: Check out repository code
- uses: actions/checkout@v2
- run: echo "📄 The \${{ github.repository }} repository has \${{ github.ref }} files and \${{ github.ref }} commits. We just checked out the files for this branch. Here is the output of ls -la:



name: GitHub Actions Demo

on: [push]

jobs:

Explore-GitHub-Actions:

runs-on: ubuntu-latest

steps:

- run: echo "🎉 The job was automatically triggered by you. It looks like you're working on GitHub Actions and that's awesome! :tada:"
- run: echo "🐧 This job is now running on a \${{ runner.os }} machine hosted by GitHub Actions, in 10 minutes of the time that you were waiting. Hit :octocat: on the top right of your terminal to see the whole config that was run, and see the terminal output for this job in the GitHub Actions log." :octocat:
- run: echo "🔍 The name of your branch is \${{ github.ref }} and your repository is \${{ github.repository }}."
- name: Check out repository code
- uses: actions/checkout@v2
- run: echo "👀 The \${{ github.repository }} repository has \${{ github.ref }} checked out at \${{ github.sha }}."



jobs:

Explore-GitHub-Actions:

```
runs-on: ubuntu-latest
```

steps:

- ```
- run: echo "🎉 The job was automatically triggered by you"
- run: echo "🐧 This job is now running on a ${{ runner.os }} machine"
- run: echo "🔍 The name of your branch is ${{ github.ref }}"
github.repository }}"
- name: Check out repository code
 uses: actions/checkout@v2
- run: echo "💡 The ${{ github.repository }} repository has the
- run: echo "💻 The workflow is now ready to test your code and build files in the repository"
```

name: GitHub Actions Demo

on: [push]

jobs:

Explore-GitHub-Actions:

runs-on: ubuntu-latest

steps:

- run: echo "🎉 The job was automatically triggered by you. It looks like you're working on GitHub Actions and that is awesome! It looks like you've just started learning GitHub Actions so I created a workflow just for you: [Explore-GitHub-Actions](#). You can learn more about GitHub Actions at: [https://docs.github.com/actions](#)."
- run: echo "🐧 This job is now running on a \${{ runner.os }} operating system with IP: \${{ runner.ip }}."
- run: echo "🔍 The name of your branch is \${{ github.ref }} and your repository is \${{ github.repository }}."
- name: Check out repository code
- uses: actions/checkout@v2
- run: echo "💡 The \${{ github.repository }} repository has been cloned to the workflow's working directory."
- run: echo "🖨️ The workflow is now ready to test your code! Please wait for your workflow to complete."



name: GitHub Actions Demo

on: [push]

jobs:

Explore-GitHub-Actions:

runs-on: ubuntu-latest

steps:

- run: echo "🎉 The job was automatically triggered by you. It looks like you're working on GitHub Actions and that's awesome! :tada:"
- run: echo "🐧 This job is now running on a \${{ runner.os }} machine hosted by GitHub Actions, in what you can see as a virtual machine with access to a network of GitHub Actions hosted runners!"
- run: echo "🔍 The name of your branch is \${{ github.ref }} and your repository is \${{ github.repository }}"
- name: Check out repository code
- uses: actions/checkout@v2
- run: echo "💡 The \${{ github.repository }} repository has been cloned to the runner.">
- run: echo "🗂 The workflow is now ready to test your code on the runner.">
- run: echo "🔖 The workflow is now ready to test your code on the runner.">

name: GitHub Actions Demo

on: [push]

jobs:

Explore-GitHub-Actions:

runs-on: ubuntu-latest

steps:

- run: echo "🎉 The job was automatically triggered by a [new push](#) to the repository. The name of your branch is \${{ github.ref\_name }} and your workflow is \${{ github.workflow }}."
- run: echo "🐧 This job is now running on a \${{ runner.os }} Ubuntu runner. Feel free to experiment with our [runner images](#) and [github actions](#) to create and test your own workflows."
- run: echo "🔎 Test your workflow in your [GitHub Actions](#) page!"
- run: echo "📄 The workflow is now ready to test your code with the [next steps](#) below!"
- run: echo "💡 This workflow was automatically generated by GitHub Actions. To get a better understanding of how GitHub Actions work, see the [documentation](#) on running workflows on your repository automatically."

github:

runs-on: ubuntu-latest

container: node:10.16-jessie

steps:

- run: echo "📄 The workflow is now ready to test your code with the [next steps](#) below!"
- run: echo "💡 This workflow was automatically generated by GitHub Actions. To get a better understanding of how GitHub Actions work, see the [documentation](#) on running workflows on your repository automatically."

name: GitHub Actions Demo

on: [push]

jobs:

Explore-GitHub-Actions:

runs-on: ubuntu-latest

steps:

- run: echo "🎉 The job was automatically triggered by you. It looks like you're working on GitHub Actions and that's awesome! :tada:"
- run: echo "🐧 This job is now running on a \${{ runner.os }} machine hosted by GitHub Actions, in a virtual environment with no access to the host's network or disk. This is the 'default' GitHub Actions runner image."
- run: echo "🔍 The name of your branch is \${{ github.ref }} and your repository is \${{ github.repository }}."
- name: Check out repository code
- uses: actions/checkout@v2
- run: echo "💡 The \${{ github.repository }} repository has been cloned to the runner."
- run: echo "🖨 The workflow is now ready to test your code in this runner."



```
name: GitHub Actions Demo
```

```
on: [push]
```

```
jobs:
```

```
 Explore-GitHub-Actions:
```

```
 runs-on: ubuntu-latest
```

```
 steps:
```

```
 - run: echo 🎉 The job was automatically triggered by you
```

```
 - run: echo 🐧 This job is now running on a ${{ runner.os }} machine
```

```
 - run: echo 🔍 The name of your branch is ${{ github.ref }}
```

```
github.repository }}, "The repository is ${{ github.repository }}"
```

```
 - name: Check out repository code
```

```
 uses: actions/checkout@v2
```

```
 - run: echo 💡 The ${{ github.repository }} repository is ready to be tested
```

```
 - run: echo 💻 The workflow is now ready to test your code on this branch
```

```
 - name: List files in the repository
```

```
 run: |
```

```
 ls ${{ github.workspace }}
```

```
on: [push]
jobs:
 Explore-GitHub-Actions:
 runs-on: ubuntu-latest
 steps:
 - run: echo "🎉 The job was automatically triggered by you"
 - run: echo "🐧 This job is now running on a ${{ runner.os }}"
 - run: echo "🔍 The name of your branch is ${{ github.ref }}"
 - run: echo "📁 The name of your repository is ${{ github.repository }}"
 - name: Check out repository code
 uses: actions/checkout@v2
 - run: echo "💡 The ${{ github.repository }} repository has been cloned to the runner"
 - run: echo "🖥️ The workflow is now ready to test your code in the runner"
 - name: List files in the repository
 run: |
 ls ${{ github.workspace }}
 - run: echo "🍏 This job's status is ${{ job.status }}"
```

```
on: [push]
```

```
jobs:
```

```
 Explore-GitHub-Actions:
```

```
 runs-on: ubuntu-latest
```

```
 steps:
```

- run: echo 🎉 The job was automatically triggered by you
- run: echo 🐧 This job is now running on a \${{ runner.os }} machine
- run: echo 🔍 The name of your branch is \${{ github.ref\_name }}

```
github.repository }}."
```

- name: Check out repository code

```
 uses: actions/checkout@v2
```

- run: echo 💡 The \${{ github.repository }} repository has been cloned to the runner

- run: echo 🖥️ The workflow is now ready to test your code on the runner

- name: List files in the repository

```
 run: |
```

```
 ls ${{ github.workspace }}
```

- run: echo 🍏 This job's status is \${{ job.status }}

```
on: [push]
```

```
jobs:
```

```
 Explore-GitHub-Actions:
```

```
 runs-on: ubuntu-latest
```

```
 steps:
```

- run: echo "🎉 The job was automatically triggered by you"
- run: echo "🐧 This job is now running on a \${runner} runner!"
- run: echo "🔍 The name of your branch is \${github.ref}

```
github.repository }}, "The repository name is ${github.repository}."
```

- name: Check out repository code  
 uses: actions/checkout@v2
- run: echo "💡 The \${github.repository} repository has been cloned to the workflow workspace"
- run: echo "💻 The workflow is now ready to test your code on the runner!"
- name: List files in the repository  
 run: |  
 ls \${github.workspace}
- run: echo "🍏 This job's status is \${job.status}. Check the status page for more details."



```
on: [push]
```

```
jobs:
```

```
 Explore-GitHub-Actions:
```

```
 runs-on: ubuntu-latest
```

```
 steps:
```

- run: echo 🎉 The job was automatically triggered by you
- run: echo 🐧 This job is now running on a \${{ runner.os }} machine
- run: echo 🔍 The name of your branch is \${{ github.ref }}
- run: echo 📁 The name of your repository is \${{ github.repository }}
- name: Check out repository code
  - uses: actions/checkout@v2
- run: echo 💡 The \${{ github.repository }} repository has \${{ github.ref }} files
- run: echo 🖥️ The workflow is now ready to test your code on the runner
- name: List files in the repository
  - run: |
    - ls \${{ github.workspace }}
- run: echo 🍏 This job's status is \${{ job.status }}

on: [push]

jobs:

Explore GitHub Actions

runs

steps:

-

-

-

github.repository } }

- name: Check out repository code

uses: actions/checkout@v2

- run: echo "💡 The \${{ github.repository }} repository

- run: echo "💻 The workflow is now ready to test you

- name: List files in the repository

run: |

ls \${{ github.workspace }}

- run: echo "🍏 This job's status is \${{ job.status }}

> ✓ Run echo "💻 The workflow is now ready to test your code on the

> ✓ List files in the repository

> ✓ Run echo "🍏 This job's status is success."

```
on: [push]
jobs:
 Explore-GitHub-Actions:
 runs-on: ubuntu-latest
 steps:
 - run: echo "🎉 The job was automatically triggered by you"
 - run: echo "🐧 This job is now running on a ${{ runner.os }}"
 - run: echo "🔍 The name of your branch is ${{ github.ref }}"
 - run: echo "📁 The name of your repository is ${{ github.repository }}"
 - name: Check out repository code
 uses: actions/checkout@v2
 - run: echo "💡 The ${{ github.repository }} repository has files in the root"
 - run: echo "💡 The workflow is now ready to test your code"
 - name: List files in the repository
 run: |
 ls ${{ github.workspace }}
 - run: echo "🍏 This job's status is ${{ job.status }}"
```

```
name: GitHub Actions Demo
on: [push]
jobs:
 Explore-GitHub-Actions:
 runs-on: ubuntu-latest
 steps:
 - run: echo "🎉 The job was automatically triggered by a ${github.event_name} event."
 - run: echo "👋 This job is now running on a ${runner.os} server hosted by GitHub!"
 - run: echo "🔍 The name of your branch is ${github.ref} and your repository is ${github.repository}."
 - name: Check out repository code
 uses: actions/checkout@v2
 - run: echo "💡 The ${github.repository} repository has been cloned to the runner."
 - run: echo "🖥 The workflow is now ready to test your code on the runner."
 - name: List files in the repository
 run: |
 ls ${github.workspace}
 - run: echo "🍏 This job's status is ${job.status}."
```



# Exercise

Repair and test my YAML file for errors.

YAML: [bit.ly/bad-yaml](https://bit.ly/bad-yaml)

TEST: [yamllint.com](https://yamllint.com)

**10 Minute Break**



# Building a CI/CD Pipeline

Byron Sommardahl

# This Segment

01

## Feedback Loop

The shorter the better.

02

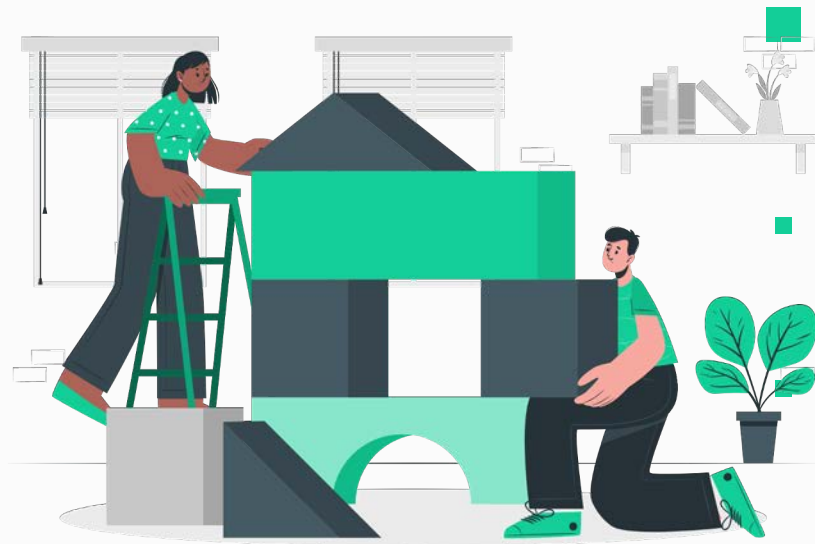
## CI Components

Recognizing the pieces that make continuous integration.

03

## The Artifact

The goal of any CI build!






[bit.ly/codex-site](https://bit.ly/codex-site)

# Continuous Integration



Checkout

Pull from source control.



Compile

Make source code usable for runtime.



Unit Test

Make sure code does what it's supposed to do.



Audit

Make sure there aren't any obvious vulnerabilities.



Analyze

Verify source code adheres to team standards.



Artifact

Package the code for deployment.



Linus Torvalds

**“Talk is  
cheap.  
Show me the  
code.”**

> > >

•

>

>

>

>

> >

•

•

•

>

>





**10 Minute Break**



# Building a CI/CD Pipeline

Byron Sommardahl

# This Segment

01

## Human Error

Take humans out of the equation.

02

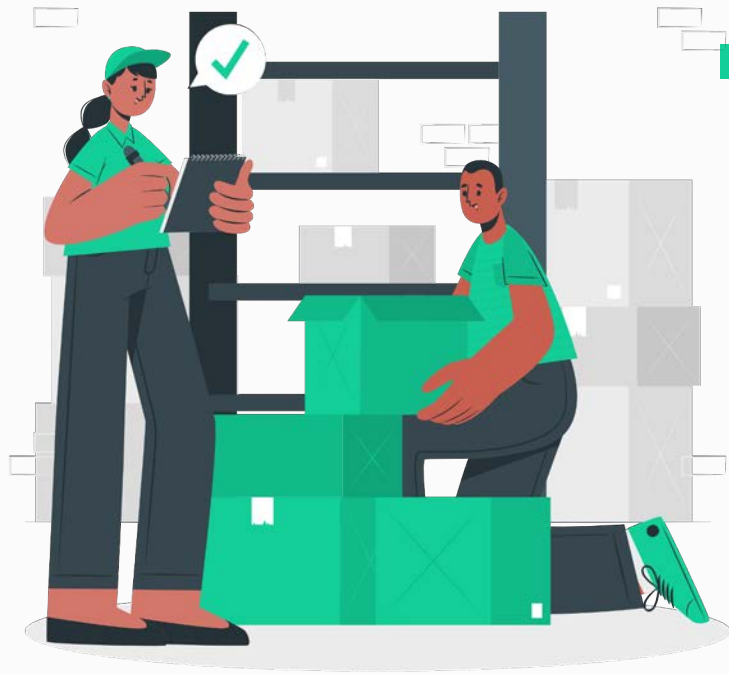
## Deploy Features

Get working software into users' hands.

03

## Sanity Check

Smoke tests for the win!



# Continuous Deployment



Create

Create any infrastructure needed for the app.



Configure

Configure infrastructure so that it works properly.



Deploy

Move executables into place so they can be used.



Verify

Check to make sure things work with smoke tests.



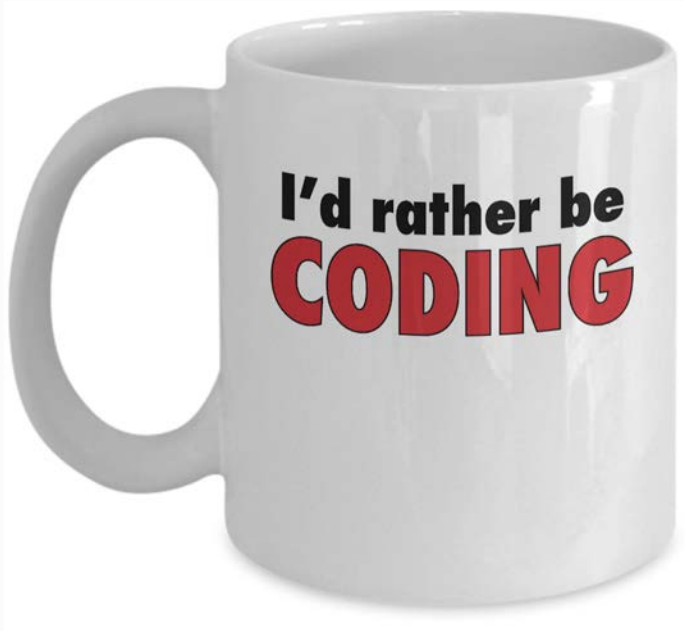
Rollback

Revert changes and clean up infrastructure on failure.



Alert

Let the team know of any failure so they can react.



**Back  
to the  
CODE!**

> > >

•

>

>

>

>

> >

•

•

•

>

>



# Stretch Goals

## Rollback

Save the world after attempting to kill it.



## Reusable Actions

Discover and utilize reusable actions.

## IaC

Give infrastructure and configuration citizenship.



## Strategies

Utilize deployment strategies for smarter outcomes.

> >

> > > >



# Thanks!

**Byron Sommardahl**  
[byron@sommardahl.com](mailto:byron@sommardahl.com)  
[calendly.com/sommardahl](https://calendly.com/sommardahl)