



# 7 Skills To Jumpstart Your Technical Career

**Dr. Sally Wahba**  
**Principal Software Engineer**  
**Splunk, a Cisco Company**  
**@sallyky**

# Agenda

---

## **1. Contributing to existing code**

# Agenda

---

1. Contributing to existing code
2. **Software debugging**

# Agenda

---

1. Contributing to existing code
2. Software debugging
3. **Scripting and software testing**

# Agenda

---

1. Contributing to existing code
2. Software debugging
3. Scripting and software testing
4. **Version control**

# Agenda

---

1. Contributing to existing code
2. Software debugging
3. Scripting and software testing
4. Version control
5. **Building code**



# Agenda

---

1. Contributing to existing code
2. Software debugging
3. Scripting and software testing
4. Version control
5. Building code
6. **Continuous Integration/Continuous Delivery/Deployment**



# Agenda

---

1. Contributing to existing code
2. Software debugging
3. Scripting and software testing
4. Version control
5. Building code
6. Continuous Integration/Continuous delivery
- 7. Linux and Containers**


# Story Time



# Story Time



# First Ticket



Teams in Space  
Scrum: Teams in Space

- Backlog
- Agile board
- Releases
- Reports
- All issues
- Components
- Add-ons

PROJECT SHORTCUTS


- Mars Team HipChat Room
- Space Station Dev Roadmap
- Teams in Space Org Chart
- Orbital Spotify Playlist
- Hyperspeed Bitbucket Repo
- + Add shortcut

## TIS-70 Scrum Board

QUICK FILTERS: Critical partners Only my partners Recently updated

### 12 To do

▼ TIS Developer Love 3 issues


 **TIS-37**

↑ Service should return prior trip details and info

SeeSpaceEZ plus


2

▼ Everything Else 21 issues

 **TIS-68**

↑ Homepage footer uses an inline style-should use class


Large Team Support

 **TIS-20**

↑ Engage Saturn Shuttle lines for group tours


Space Travel Partners

3

 **TIS-12**


⊘ Create 90 day plans for all departments in Mars office

### 2 In progress

 **TIS-10**

↑ Bad JSON data coming back from hotel API


SeeSpaceEZ plus

 **TIS-17**

↑ Engage Saturn's Rings Resort as preferred


Space Travel Partners

### 3 Done

 **TIS-8**


↑ Requesting flights is now taking > 5 seconds

SeeSpaceEZ plus

 **TIS-56**

↑ Add pointer to main css file to create child themes

Large Team Support

 **TIS-45**

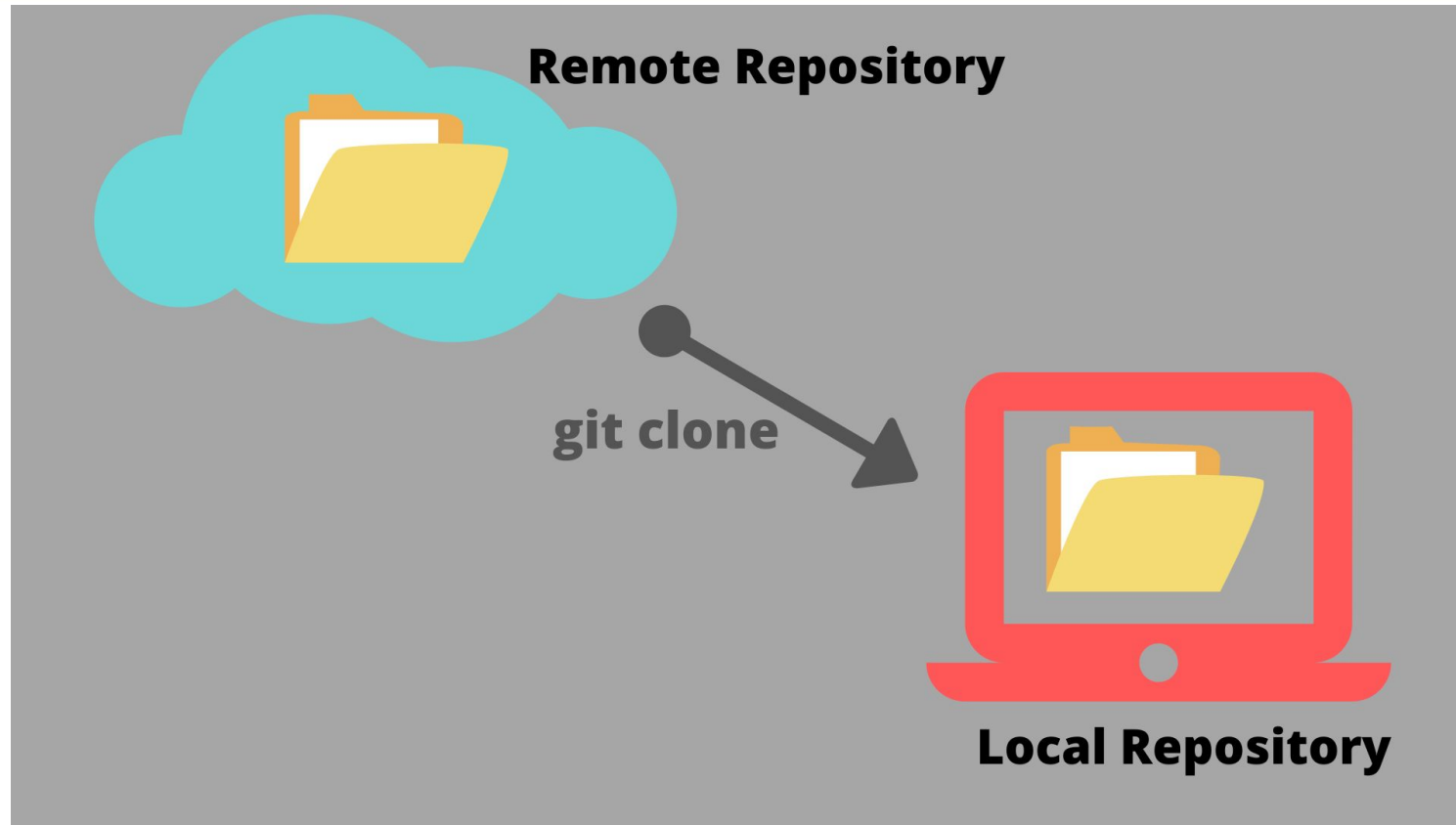
↑ Email non registered users to sign up with TIS

SeeSpaceEZ plus

# Version Control

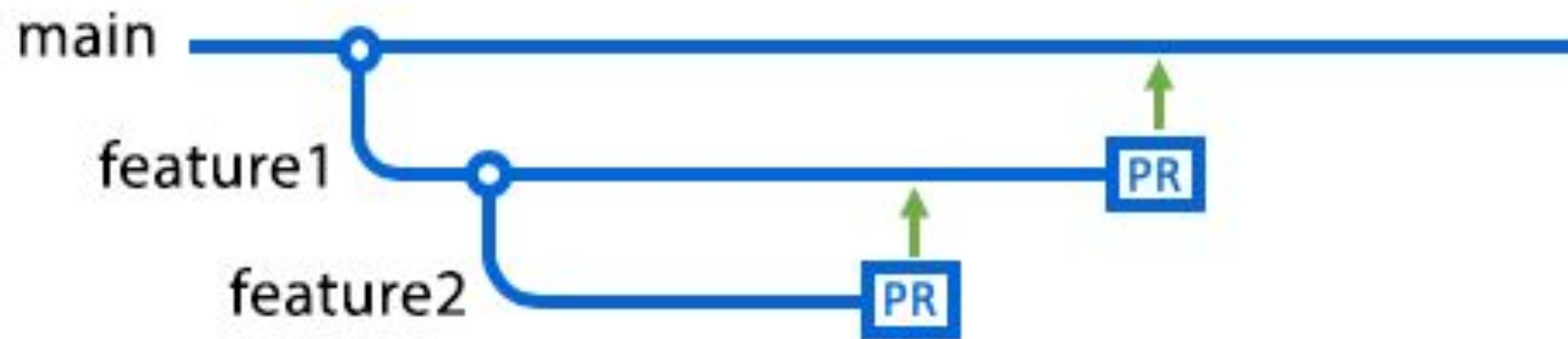


# Version Control



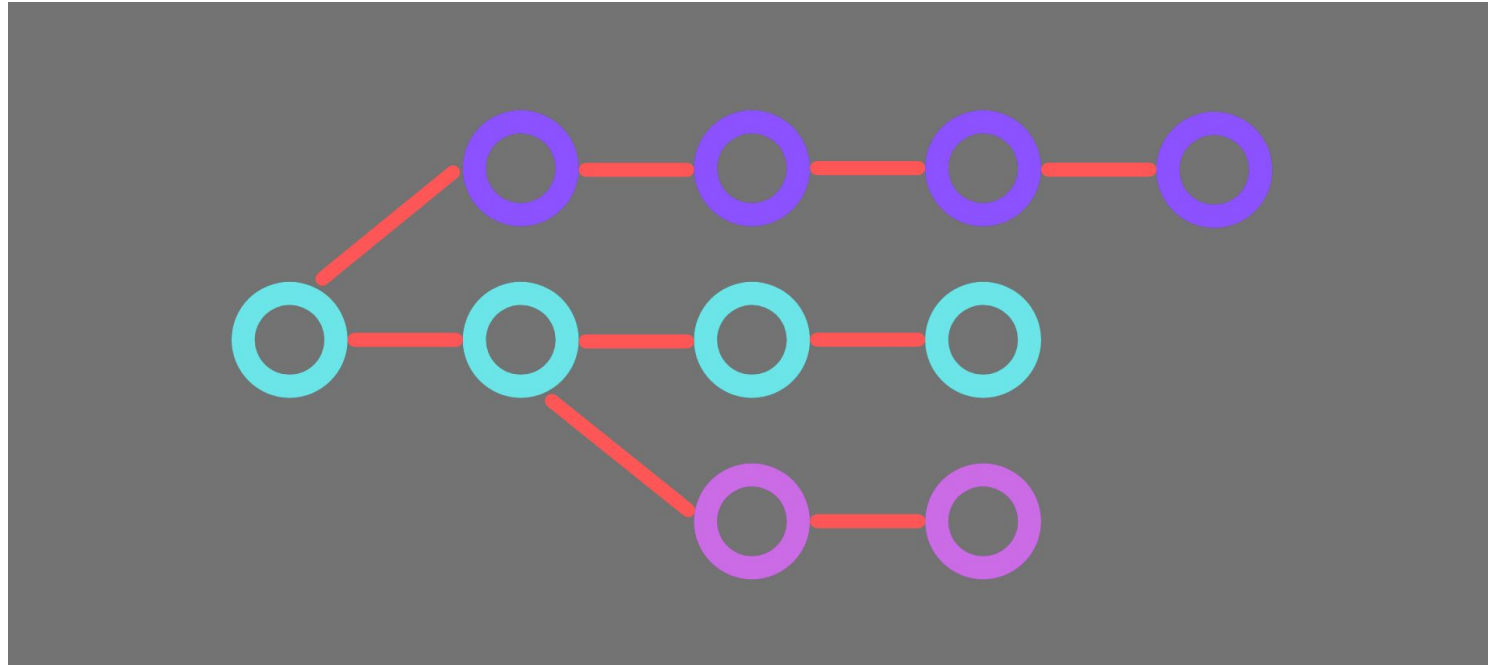
```
$ git clone git@github.com:myawesomecompany/awesomesource
```

# Version Control





# Version Control

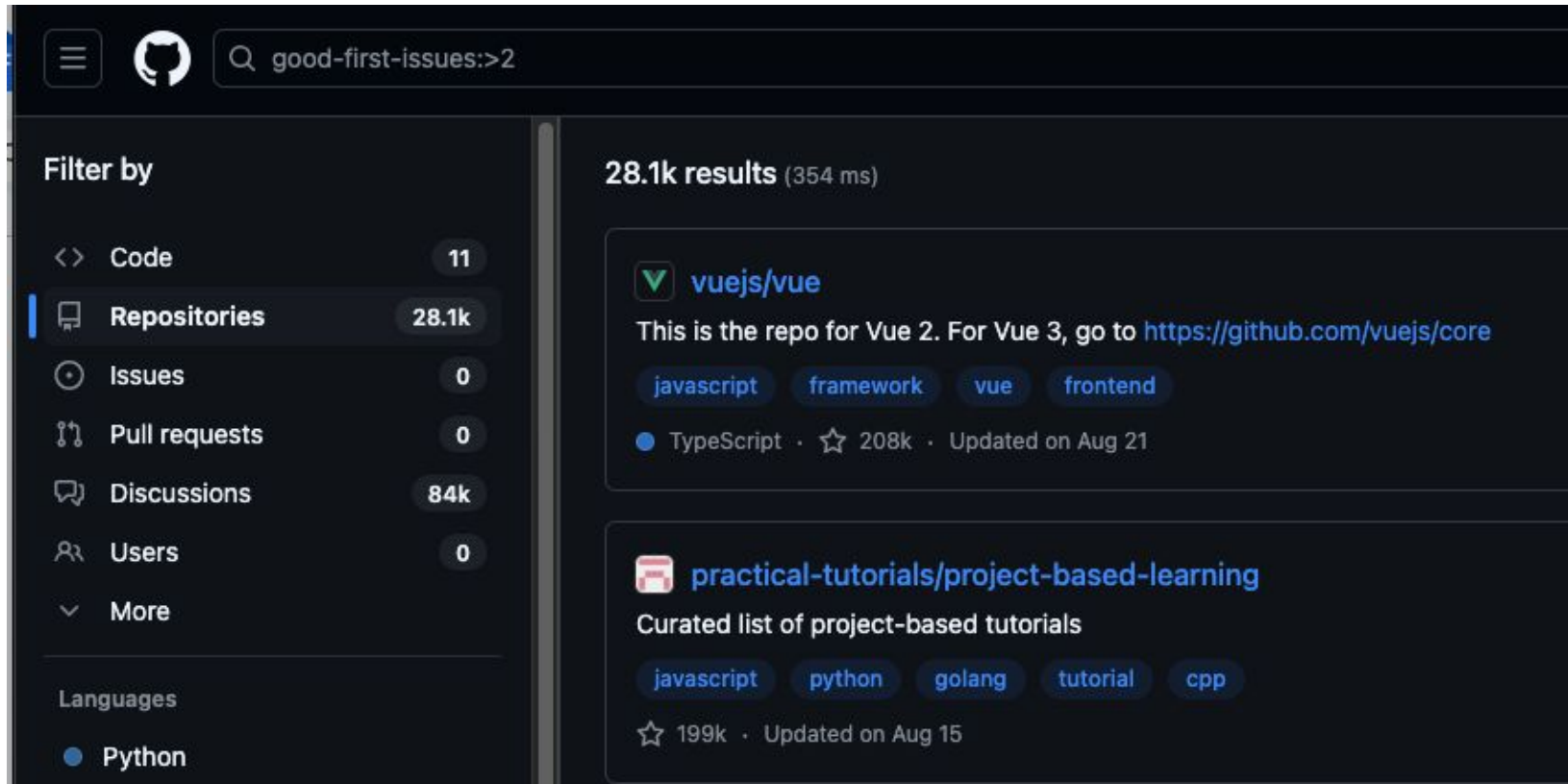


**\$ git checkout -b myFirstTicket**

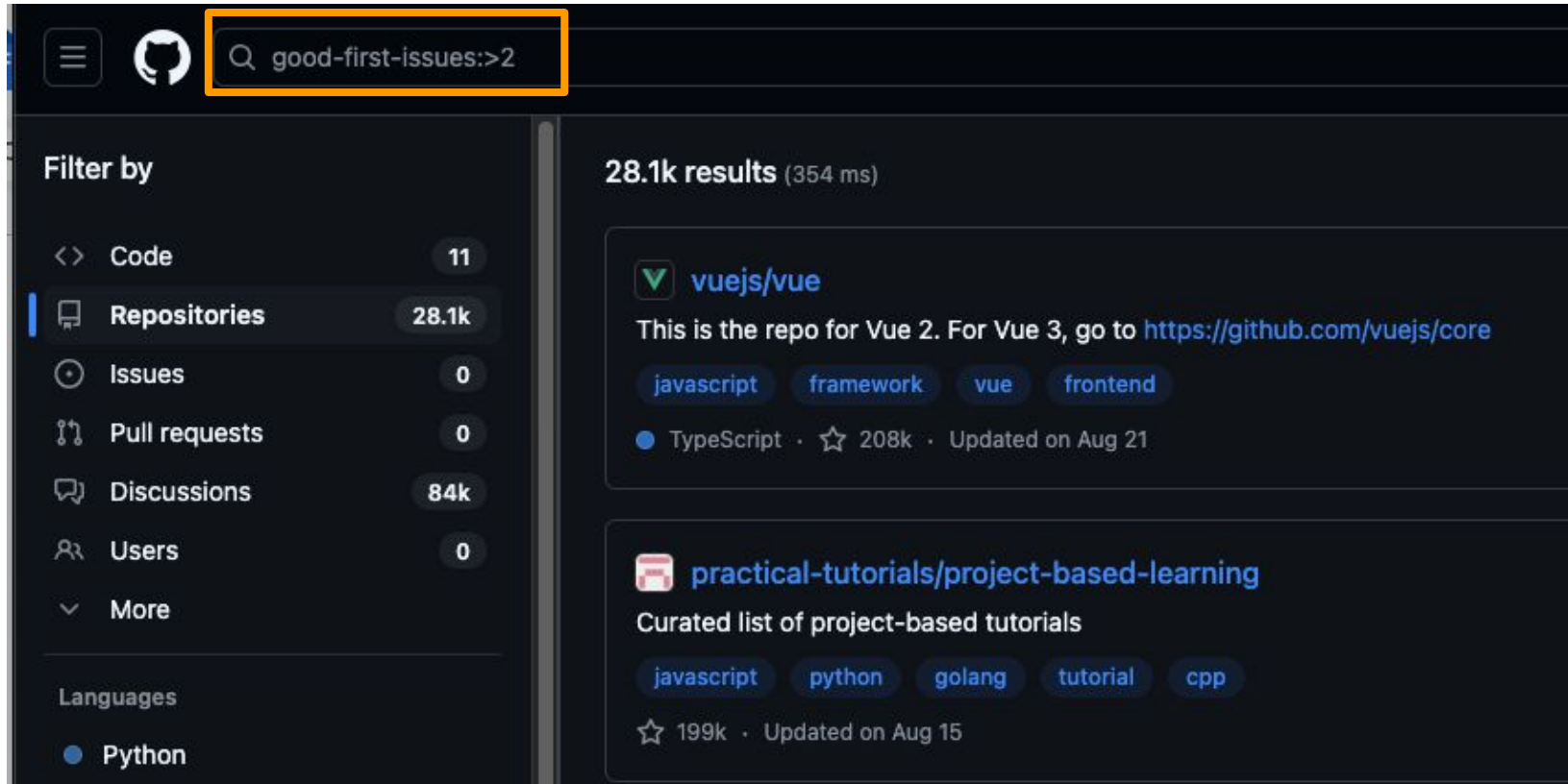
# Contributing to Existing Code



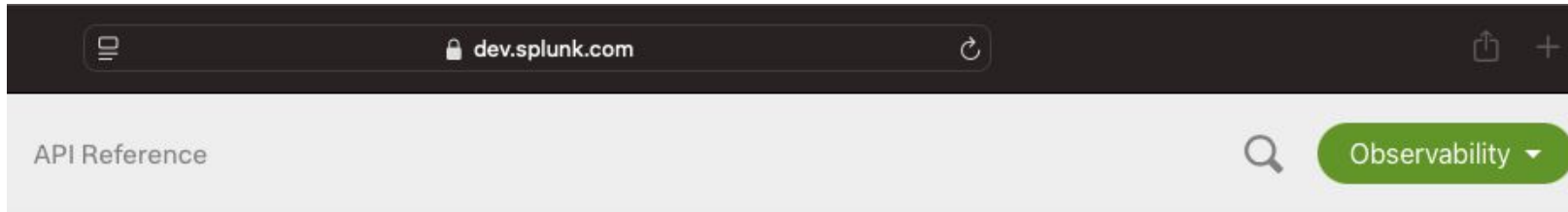
# Contributing to Existing Code



# Contributing to Existing Code



# Contributing to Existing Code



## Example: Send data to Splunk Observability Cloud

To do anything with the user interface or the API, you need to send data to Splunk Observability Cloud. Although the most common mechanism for doing this is to use an integration or SDK library, you can also use the API. Splunk Observability Cloud persists incoming data for further use.

To experiment with Splunk Observability Cloud, start by sending data, as shown in the following `curl` command:

```
$ curl --request POST \
--header "Content-Type: application/json" \
--header "X-SF-TOKEN: <ORG_TOKEN>" \
--data \
'{
  "gauge": [
    {
      "metric": "memory.free",
      "dimensions": { "host": "server1" },
      "value": 42
    }
  ]
}' \
https://ingest.<REALM>.signalfx.com/v2/datapoint
```

# Contributing to Existing Code

## Create issue

Issue type \*

Story

[Learn about issue types](#)

Status

Todo

This is the issue's initial status upon creation

General

Security SLA Extension

SFDC

Summary \*

Update documentation for XYZ module



# Contributing to Existing Code

```
ErrCount int64
}

// GzipHandler transparently
func (z *ReadZipper) GzipHan
return http.HandlerFunc(
    var err error
    if r.Header.Get( key:
        gzi := z.zippers
        if gzi != nil {
            gz := gzi.(*gzip.Reader)
            // put it back
```

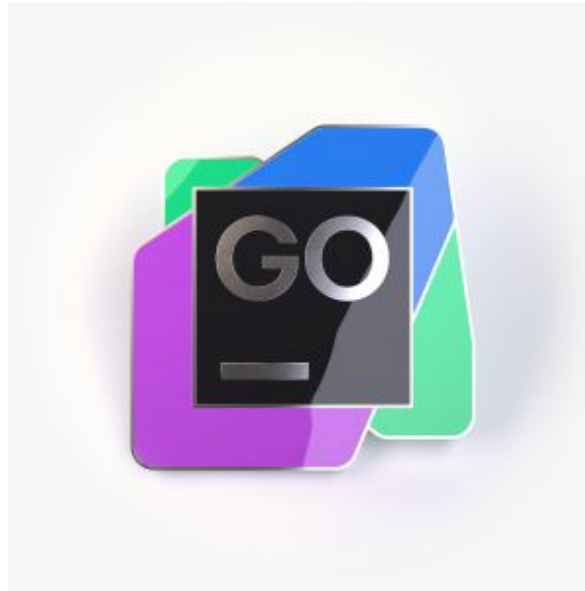
Method GzipHandler in github.com/signalfx/ing

Project Files

- collectdlistener.go protocol/collectd 214
- signalfxlistener.go protocol/signalfx 323
- zipper.go protocol/zipper 26
- zipper\_test.go protocol/zipper 61



# Building Code



# Building Code

```
M Makefile x
1 SHELL = /bin/bash
2 BASE = $(CURDIR)
3 CP_RF = cp -rf
4 GOLANGCI_LINT_VERSION = 1.49.0
5
6 # enable module support across all go commands.
7 export GOMOD = on
8
9 .SILENT: ; # no need for @
10 .ONESHELL: ; # recipes execute in same shell
11 .NOTPARALLEL: ; # wait for this target to finish
12
13 # default is verification of sfxinternalgo which includes all services
14 .PHONY: verify
15 ► verify: install-tools lint test
```

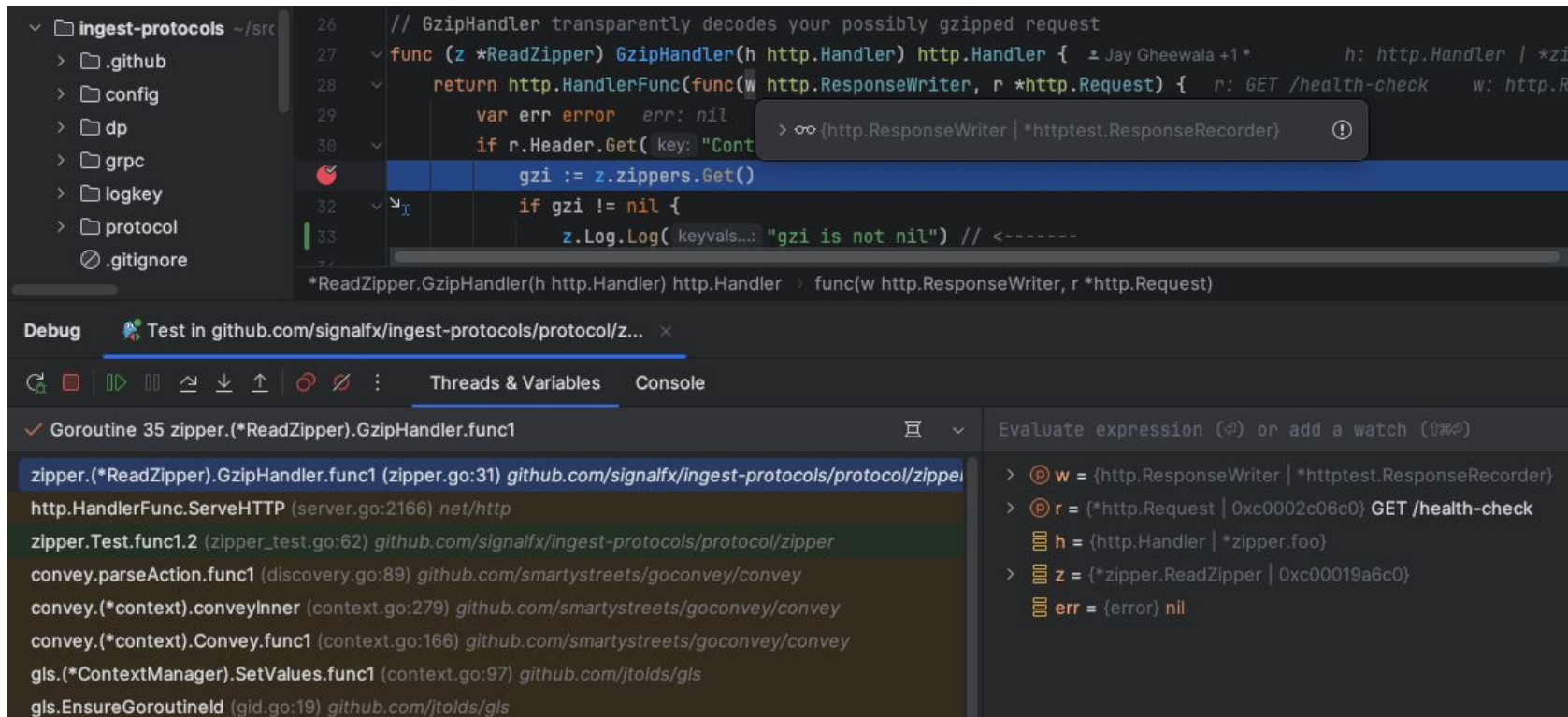
# Software Debugging



# Software Debugging

```
if gzi != nil {  
    z.Log.Log( keyvals...: "gzi is not nil") // <-----  
    gz := gzi.(*gzip.Reader)  
    // put it back  
    defer z.zippers.Put(gz)  
    err = gz.Reset(r.Body)  
    if err == nil {  
        defer log.IfErr(z.Log, gz.Close())  
        // nasty? could construct another object but seems expensive  
        r.Body = gz  
        z.Log.Log( keyvals...: "Incrementing hit count") // <-----  
        atomic.AddInt64(&z.HitCount, delta: 1)  
    }  
}
```

# Software Debugging



The screenshot displays an IDE with a Go project named 'ingest-protocols'. The code in the editor defines a `GzipHandler` function that wraps an `http.Handler`. It uses `zippers` to manage gzip compression. A breakpoint is set at line 31, where `gzi := z.zippers.Get()` is executed. The debug console shows the current goroutine stack, with the top frame being `zipper.(*ReadZipper).GzipHandler.func1`. The 'Threads & Variables' pane on the right shows the state of variables: `w` is an `http.ResponseWriter`, `r` is an `http.Request` for `GET /health-check`, `h` is an `http.Handler`, `z` is a `ReadZipper`, and `err` is `nil`.

```
26 // GzipHandler transparently decodes your possibly gzipped request
27 func (z *ReadZipper) GzipHandler(h http.Handler) http.Handler {
28     return http.HandlerFunc(func(w http.ResponseWriter, r *http.Request) {
29         var err error
30         if r.Header.Get("Content-Encoding") == "gzip" {
31             gzi := z.zippers.Get()
32             if gzi != nil {
33                 z.Log.Log(keyvals...: "gzi is not nil") // <-----
34             }
35         }
36         w.WriteHeader(r.StatusCode)
37         h.ServeHTTP(w, r)
38     })
39 }
```

Debug Test in github.com/signalfx/ingest-protocols/protocol/z...

Threads & Variables Console

✓ Goroutine 35 zipper.(\*ReadZipper).GzipHandler.func1

zipper.(\*ReadZipper).GzipHandler.func1 (zipper.go:31) github.com/signalfx/ingest-protocols/protocol/zipper

http.HandlerFunc.ServeHTTP (server.go:2166) net/http

zipper.Test.func1.2 (zipper\_test.go:62) github.com/signalfx/ingest-protocols/protocol/zipper

convey.parseAction.func1 (discovery.go:89) github.com/smartystreets/goconvey/convey

convey.(\*context).conveyInner (context.go:279) github.com/smartystreets/goconvey/convey

convey.(\*context).Convey.func1 (context.go:166) github.com/smartystreets/goconvey/convey

gls.(\*ContextManager).SetValues.func1 (context.go:97) github.com/jtolds/gls

gls.EnsureGoroutineId (gid.go:19) github.com/jtolds/gls

Evaluate expression (⌘) or add a watch (⇧⌘)

> w = {http.ResponseWriter | \*httptest.ResponseRecorder}

> r = {http.Request | 0xc0002c06c0} GET /health-check

> h = {http.Handler | \*zipper.foo}

> z = {\*zipper.ReadZipper | 0xc00019a6c0}

> err = {error} nil

# Scripting and Software Testing

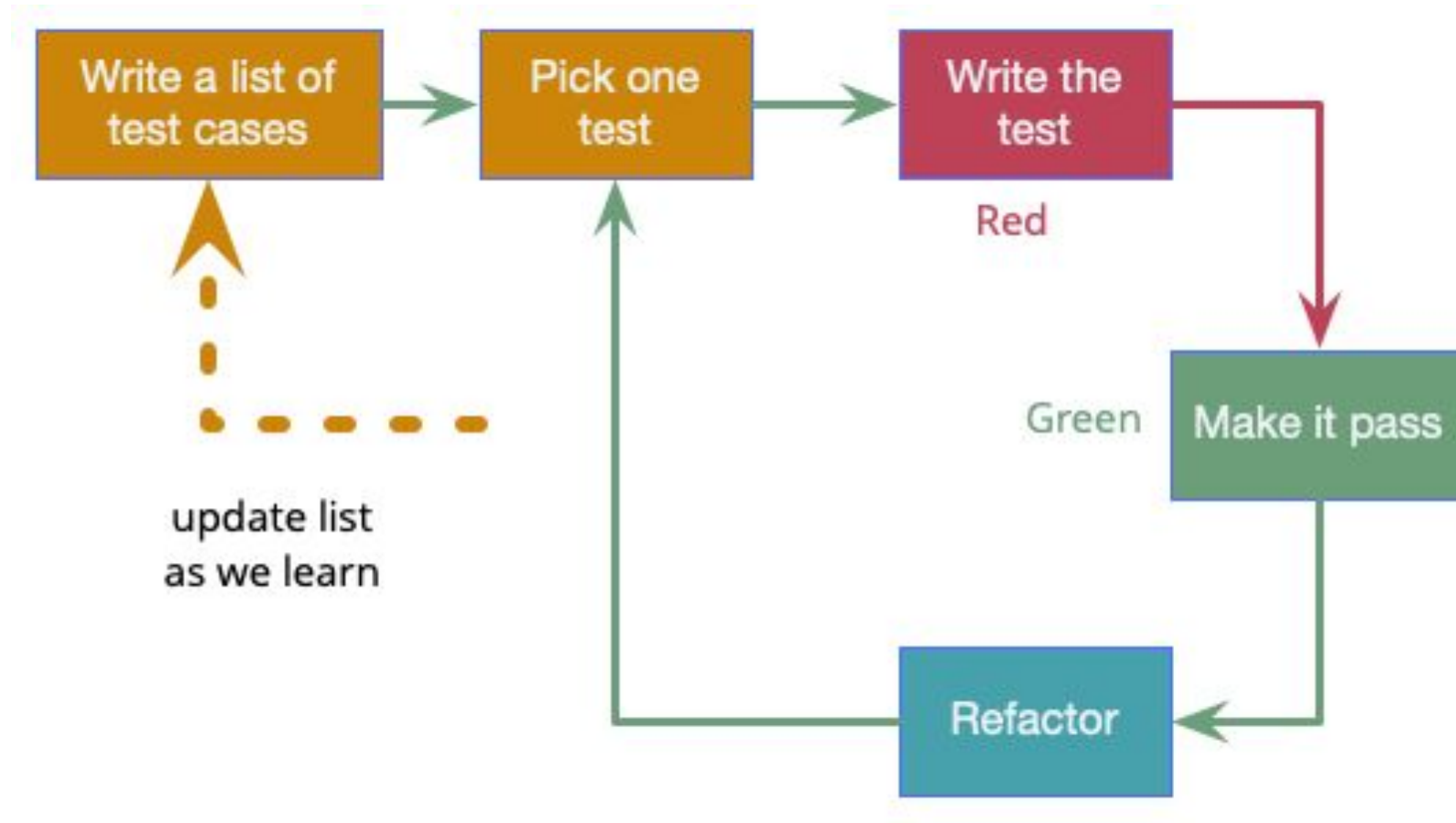
**PASSED**

# Scripting and Software Testing

```
func Test(t *testing.T) {  
    Convey( items..., "Test zipper", t, func() {  
        zippers := NewZipper()  
        badZippers := newZipper(func(r io.Reader) (*gzip.Reader, error) {  
            return new(gzip.Reader), errors.New( msg: "nope")  
        })  
        f := new(foo)  
        zipped := new(bytes.Buffer)  
        w := gzip.NewWriter(zipped)  
        _, err := w.Write([]byte("OK"))  
        So(err, ShouldBeNil)  
        So(w.Close(), ShouldBeNil)  
        tests := []struct {...}{  
            { zipper: zippers, name: "test non gzipped", data: []byte("OK"),
```



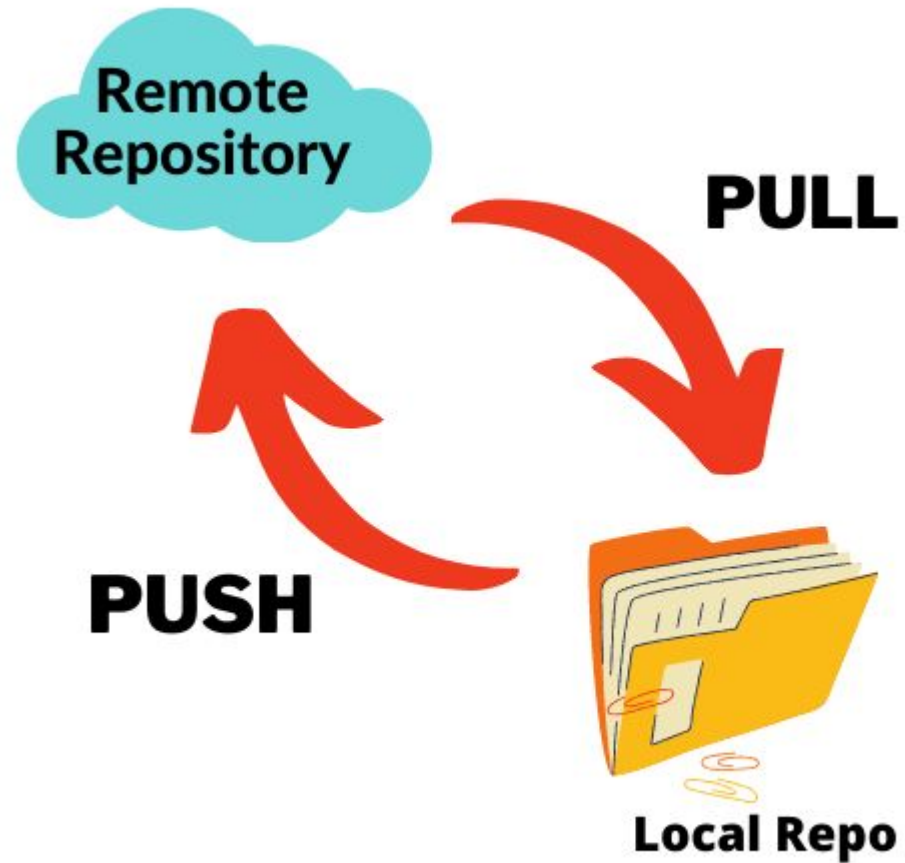
# Scripting and Software Testing



# Scripting and Software Testing

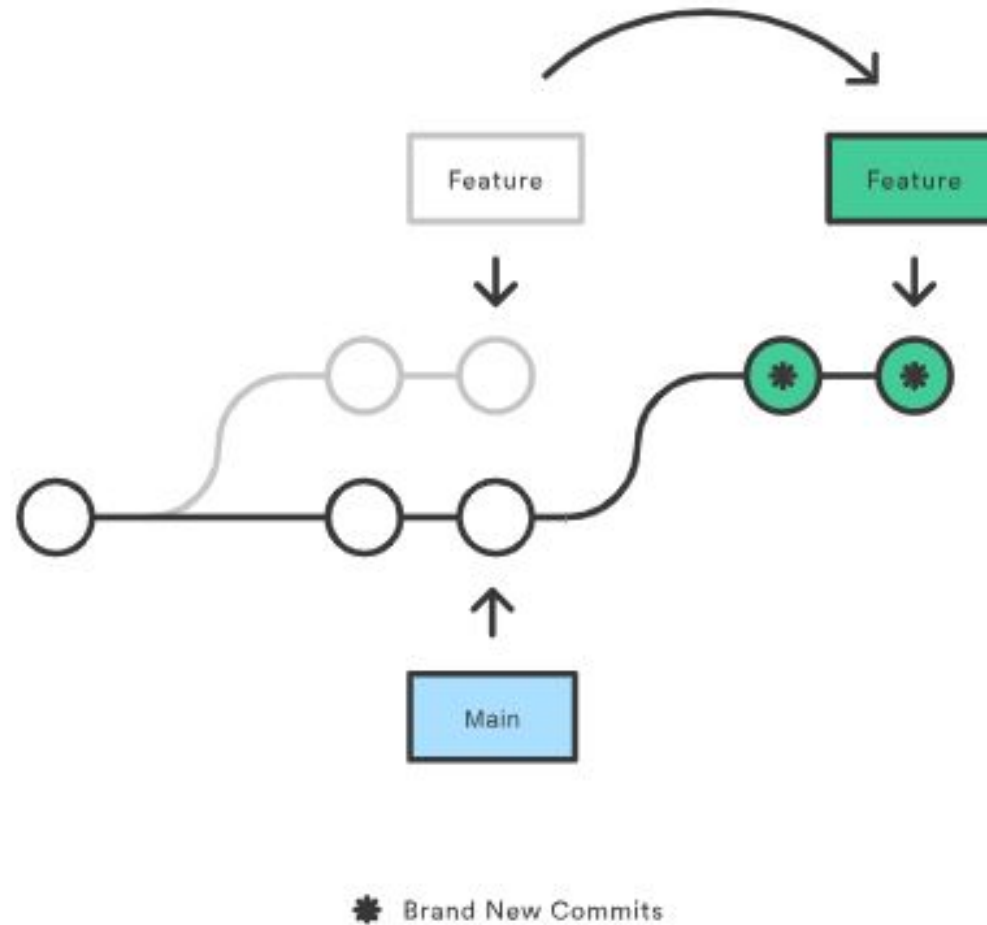


# Version Control



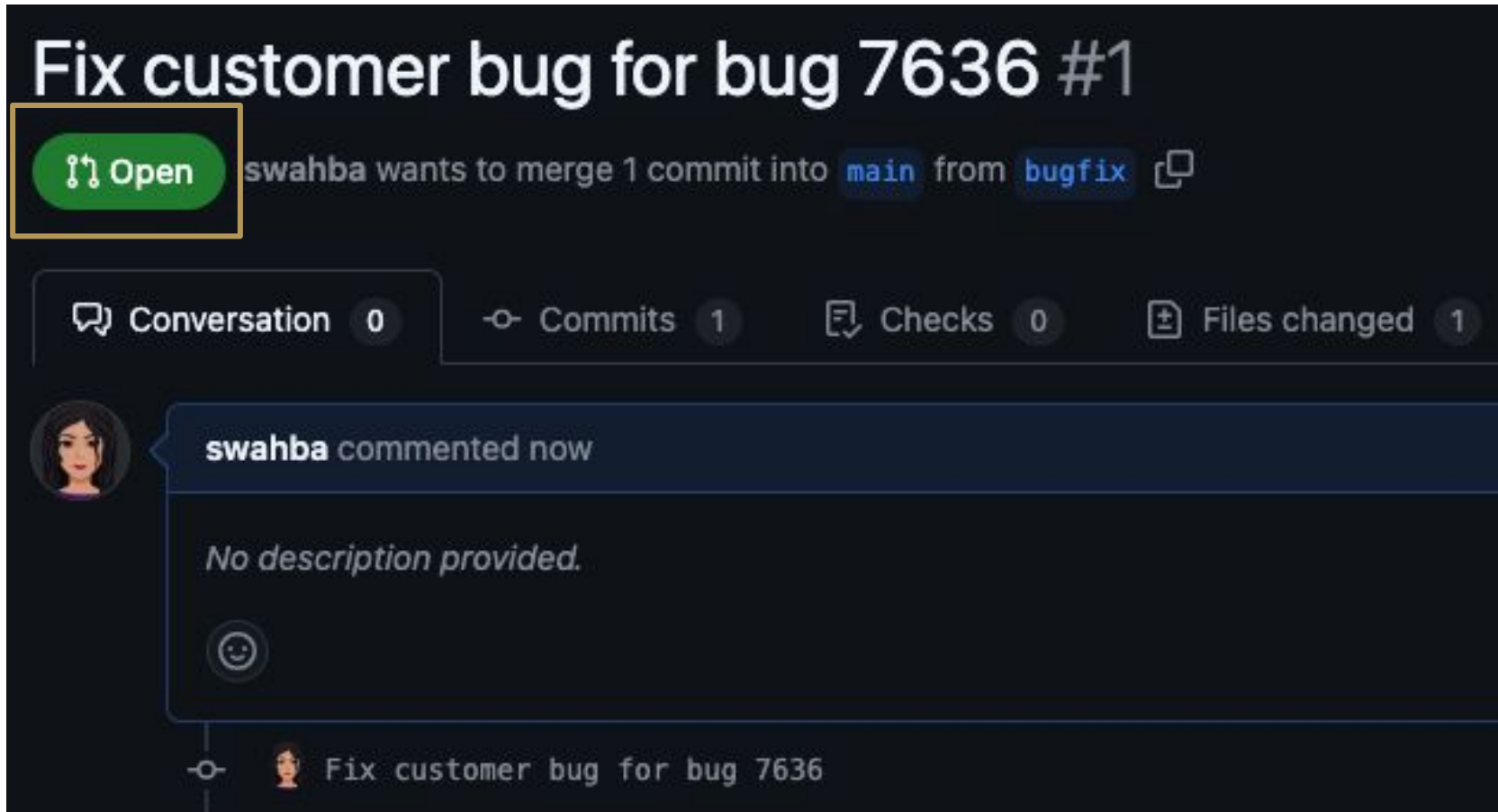
```
$ git push -set-upstream origin myFirstTicket
```

# Version Control

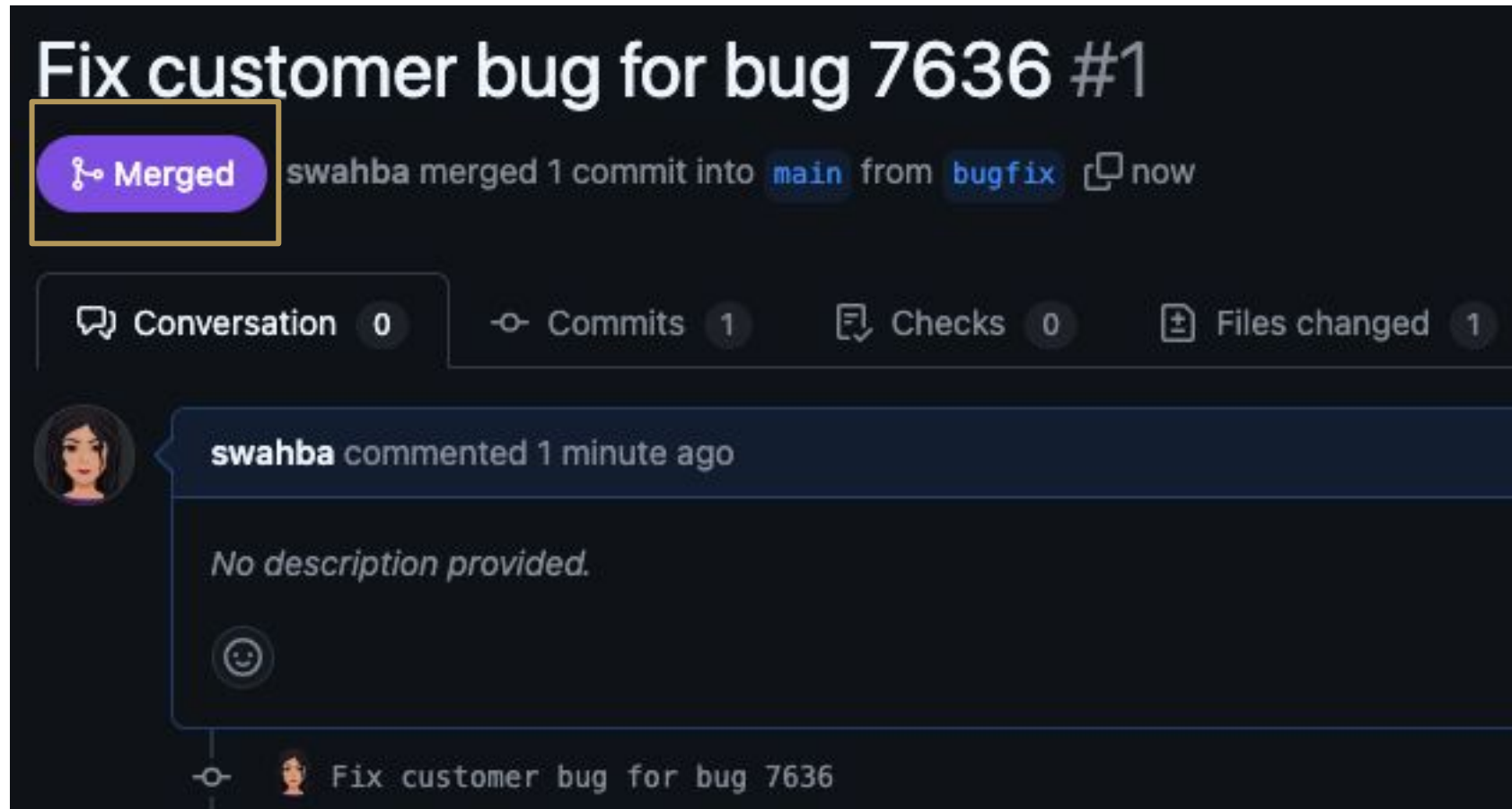


**\$ git rebase main**

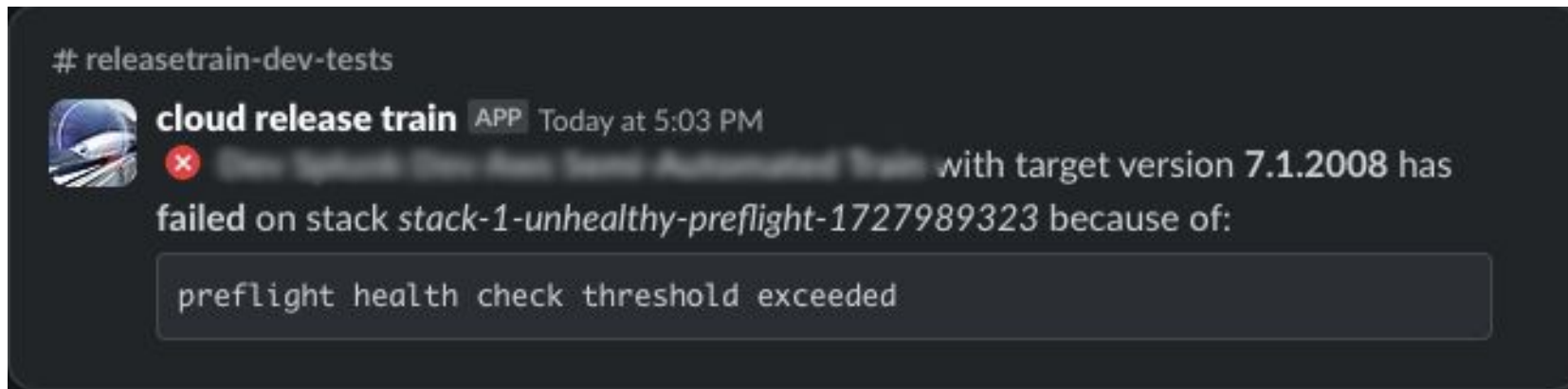
# Version Control



# Version Control

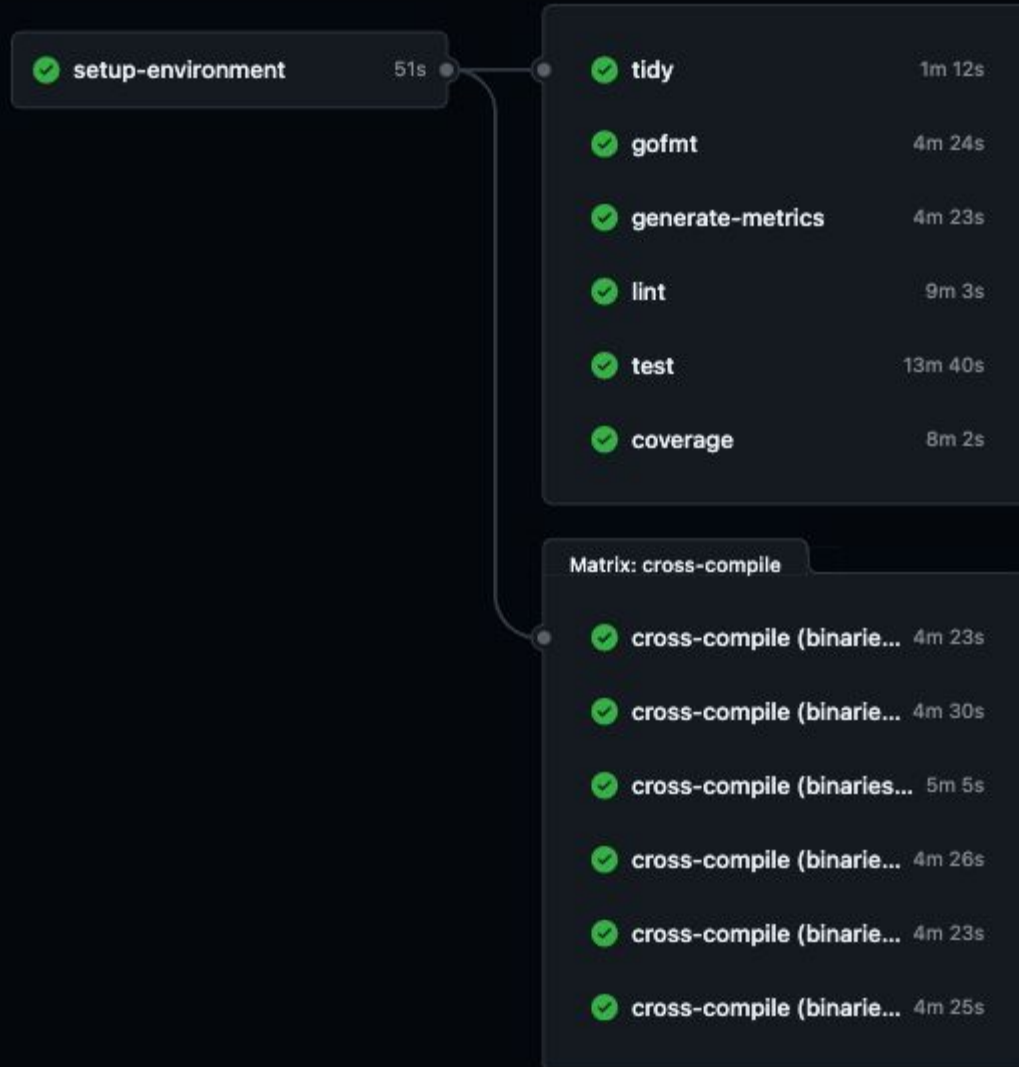


# CI/CD

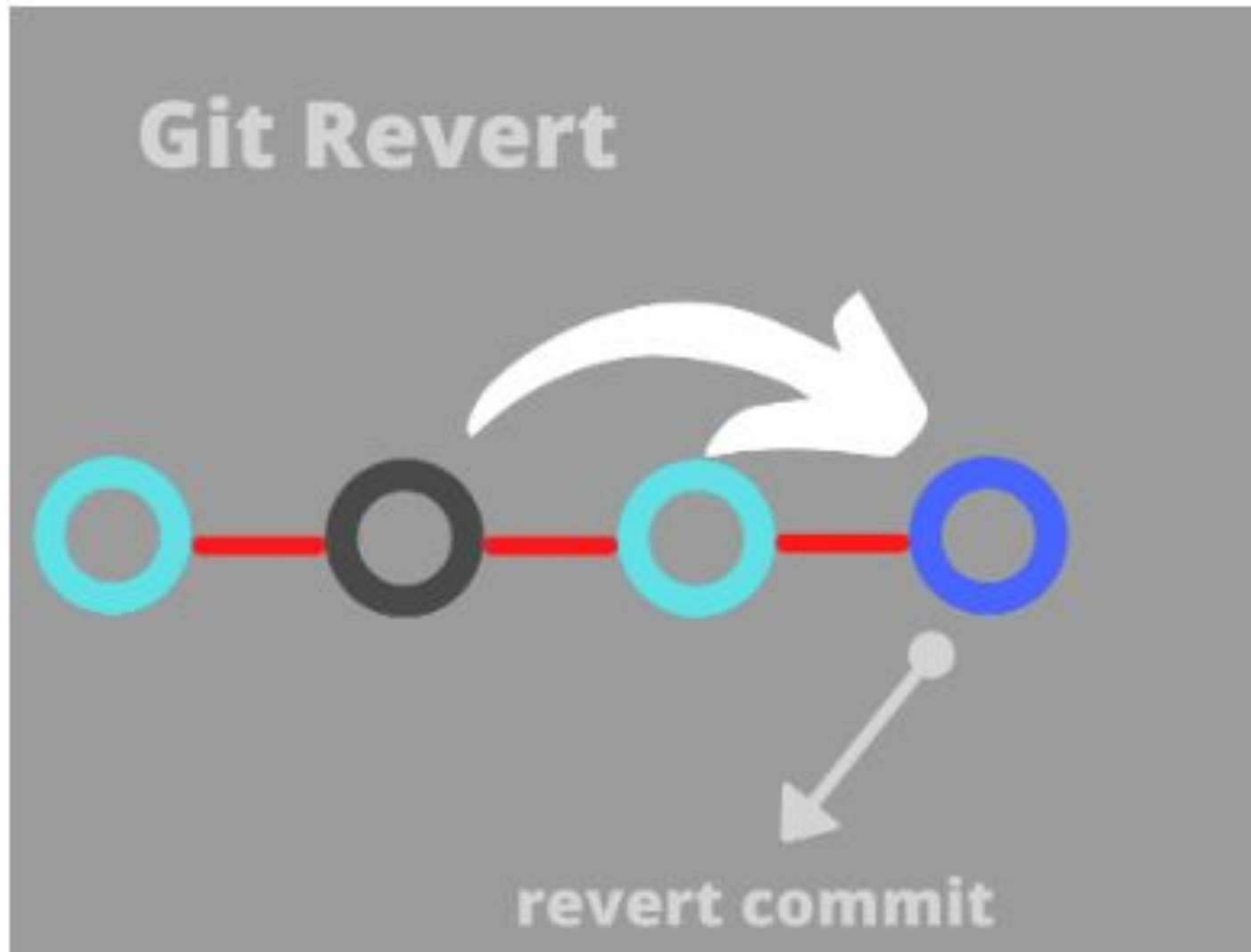




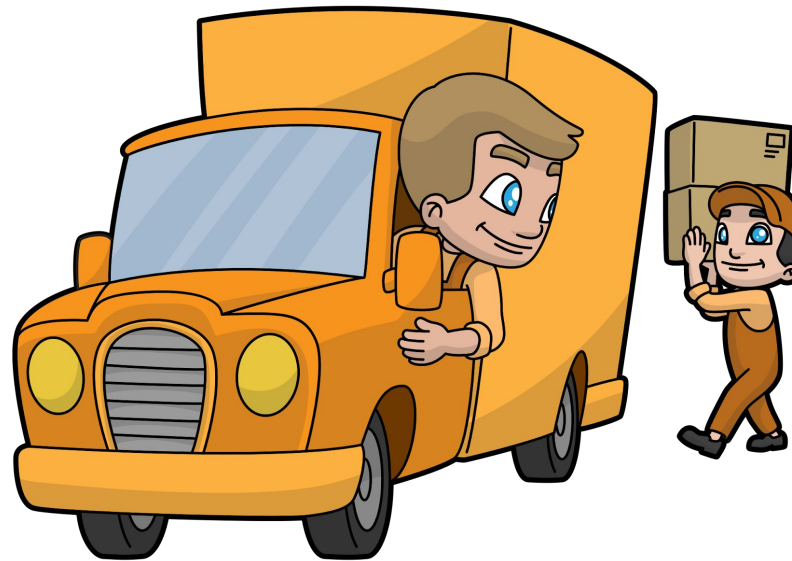
# CI/CD



# Version Control



# CI/CD



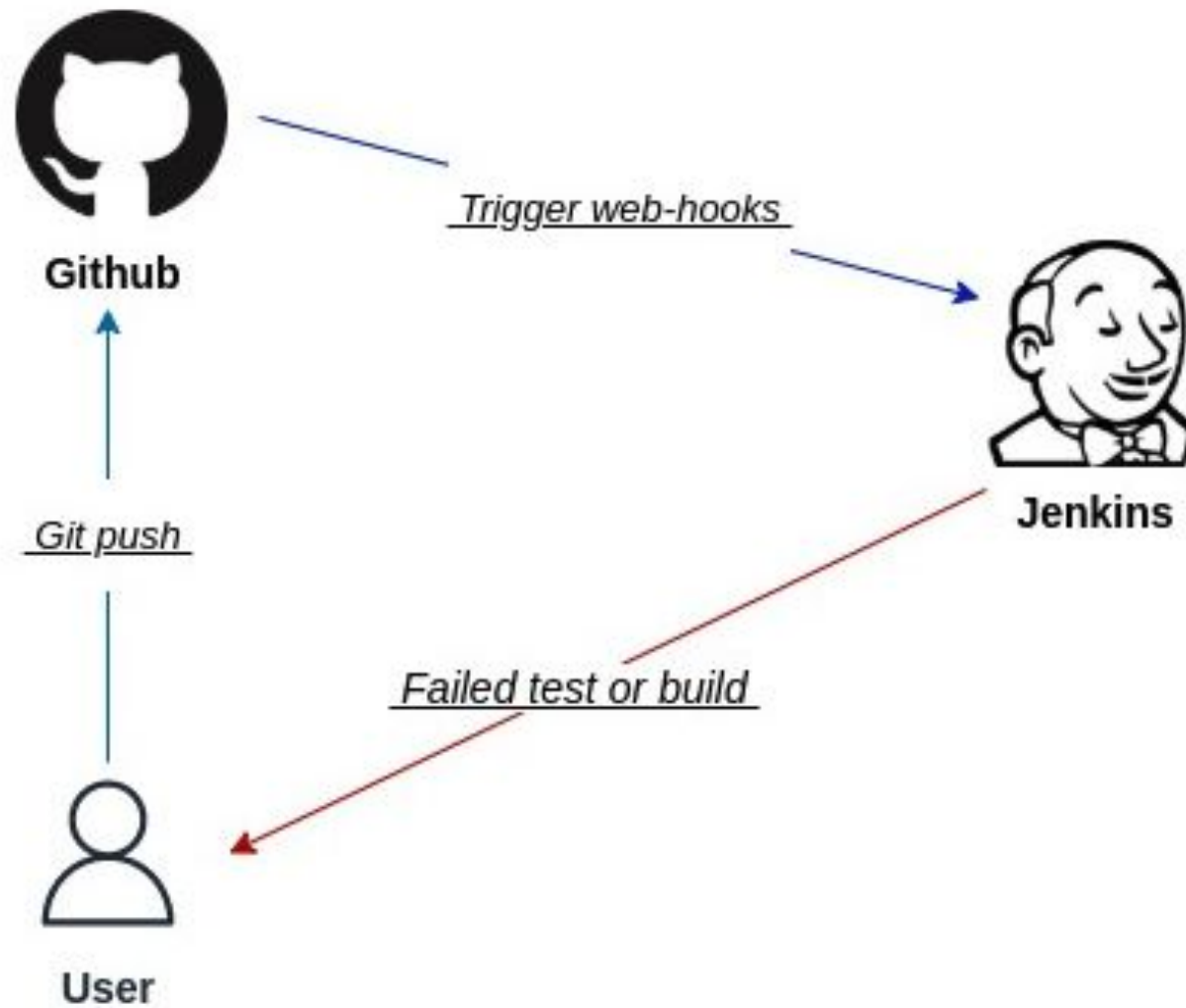
# CI/CD

- Continuous integration is the practice of automatically and frequently integrating code changes to a shared source code repository.
- Continuous delivery refers to the integration, testing, and delivery of code changes.
- Continuous deployment automatically releases the updates into a production environment.

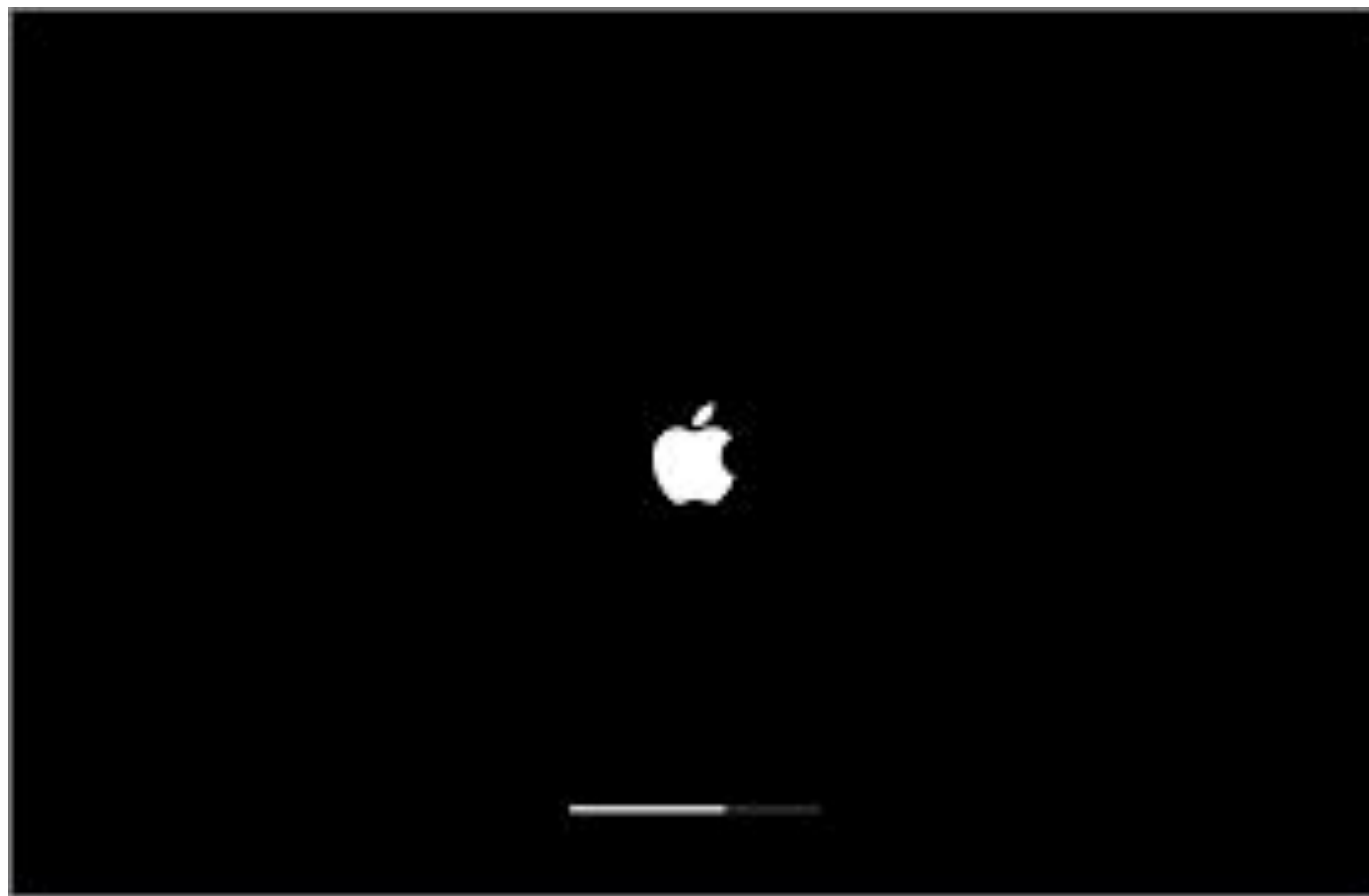
# CI/CD



# CI/CD

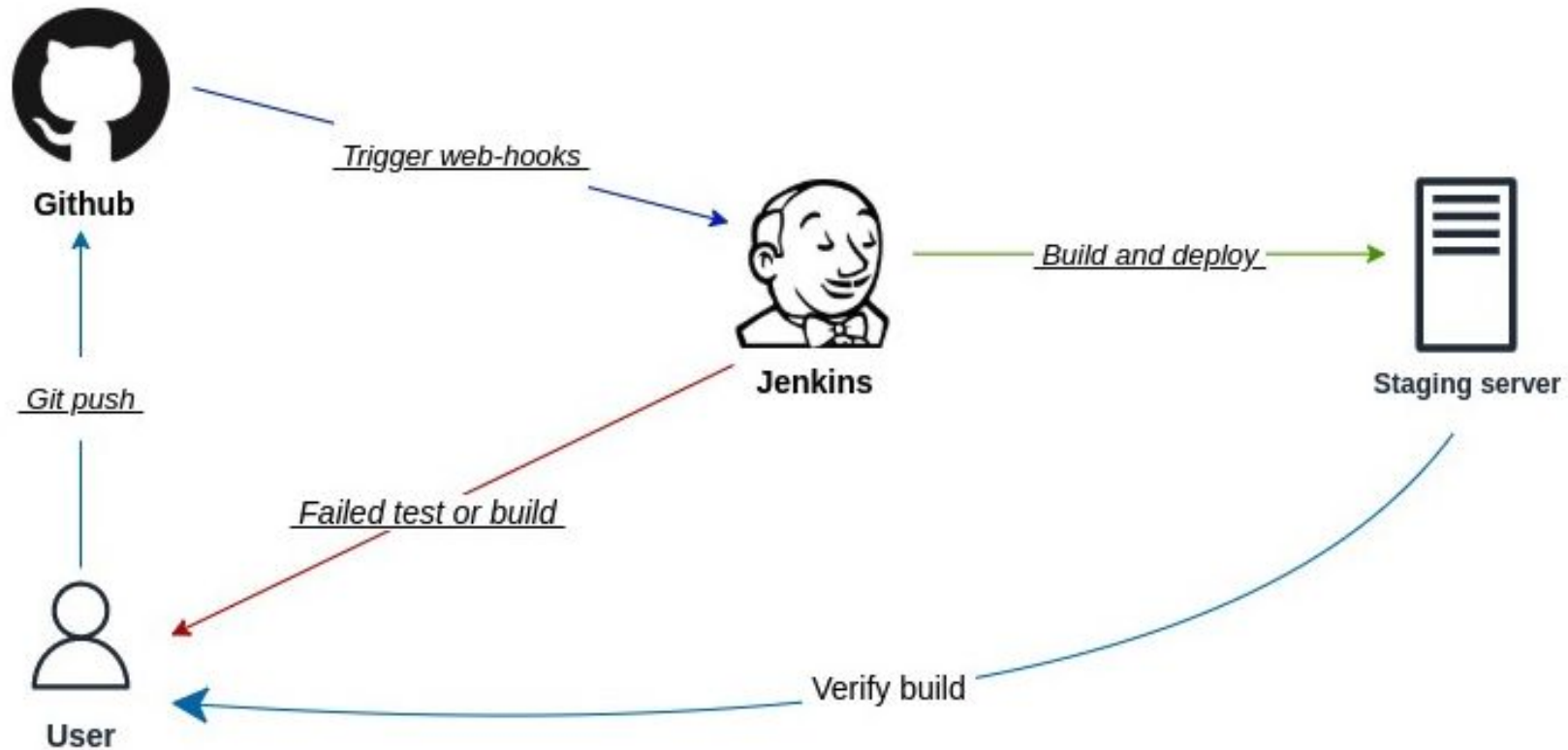


# CI/CD





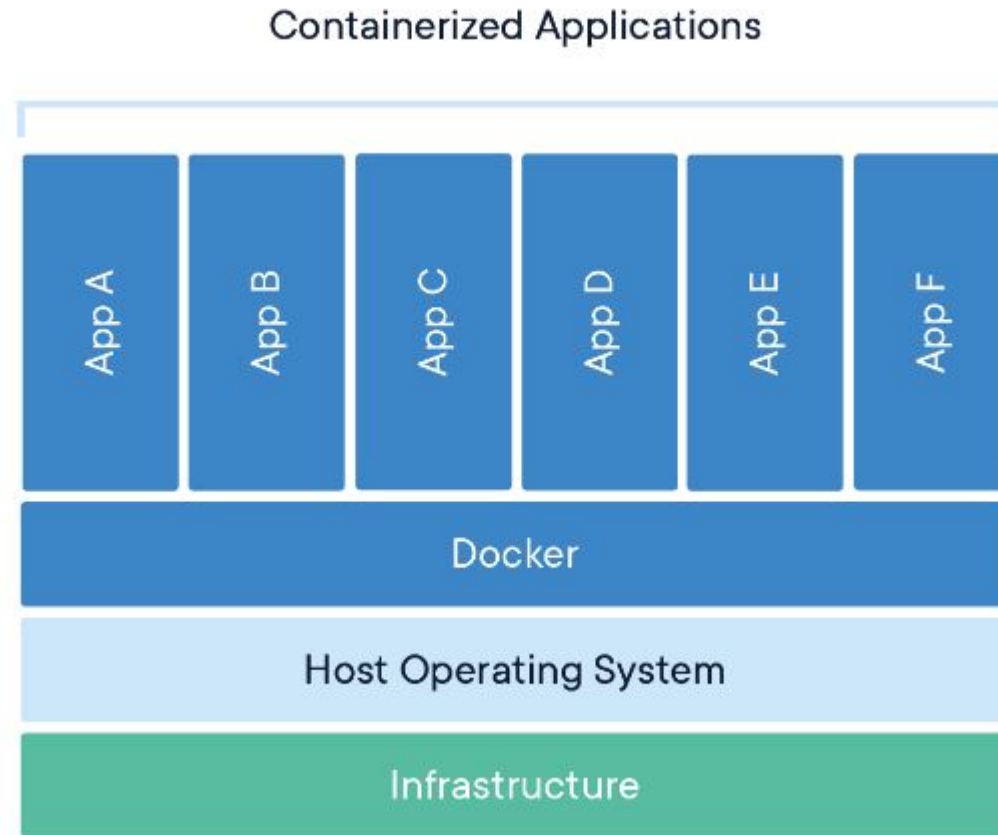
# CI/CD



# Linux & Containers



# Linux & Containers



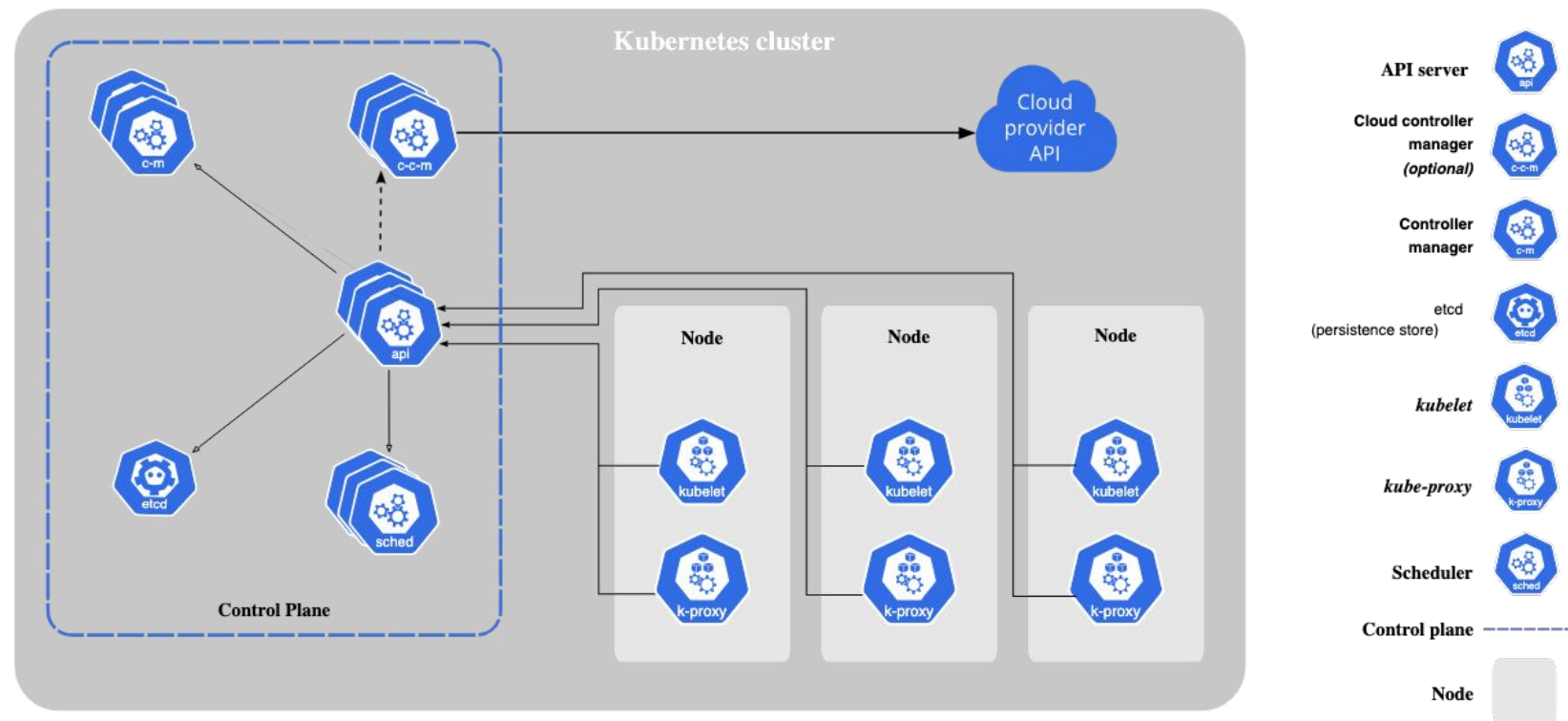
# Linux & Containers

```
GNU GRUB version 2.06

Boot Linux Lite 6.0 Live System
Direct install Linux Lite 6.0
Compatibility Mode Linux Lite 6.0
*OEM install Linux Lite 6.0 (for manufacturers)
Check for file corruption - boots into Live desktop if none found
Reboot
Shut Down

Use the ↑ and ↓ keys to select which entry is highlighted.
Press enter to boot the selected OS, 'e' to edit the commands
before booting or 'c' for a command-line.
```

# Linux & Containers







# Summary

---

1. Contributing to existing code
2. Software debugging
3. Scripting and software testing
4. Version control
5. Building code
6. Continuous Integration/Continuous delivery
7. Linux and Containers



# Slides

---

The slides are available at:

<https://github.com/swahba/conferencetalks>

# Acknowledgements

Special thanks to my colleagues:

- Amneet Kaur
- Gail Carmichael
- Jessi Jullie

# Questions?

# Thank you!

**@sallyky**  
**[linkedin.com/in/sallywahba/](https://www.linkedin.com/in/sallywahba/)**

**Image sources referenced in speaker notes.**