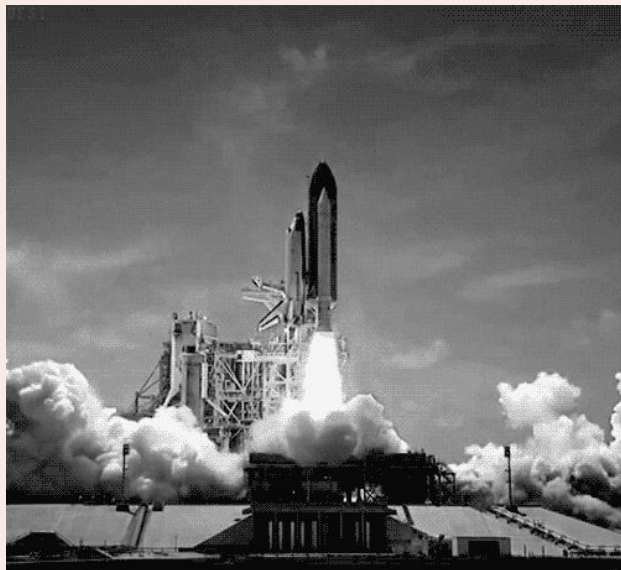


# Deployment

CSCI 39549  
October 24, 2018

# Deployment



## Agenda

---

- 1 What is deployment?
- 2 *How does it work in the industry?*
- 3 Workshop
- 4 Homework assignment

# What is Deployment?

“Software deployment is all of the activities that make a software system available for use.”

---

In other words....



**WORLD**  
**WIDE**  
**WEB** 

---

# In other words....



**WORLD  
WIDE  
WEB** 

Deployment is the process of getting code off your computer and onto a place accessible by others.



---

*How does it work in the  
industry?*



---

# Software releases – the early days

## Long-running projects

---

A team might work for years to get something out.

## Expensive to fix bugs

---

Fixing a bug meant physically delivering a software update to consumers.

## Releases owned by a separate team

---

Developers built software and other functions (eg a release engineer) were responsible for deploying the software.



---

# Waterfall software development

In software development, it tends to be among the less iterative and flexible approaches, as progress flows in largely one direction ("downwards" like a waterfall) through the phases of:

- Conception,
- Initiation,
- Analysis,
- Design,
- Construction,
- Testing,
- Deployment, and
- Maintenance.

["Waterfall Model" on Wikipedia](#)



---

# Nowadays, most companies release more frequently:

## Continuous Integration

---

Developers push their code to a shared master branch as frequently as possible.

---

# Nowadays, most companies release more frequently:

## Continuous Integration

---

Developers push their code to a shared master branch as frequently as possible.

## Continuous Delivery

---

Have tooling in place so that you can push a button and deploy whatever's on master at any given time.

**NOTE:** this means that anything pushed to master needs to be release-ready!

---

# Nowadays, most companies release more frequently:

## Continuous Integration

---

Developers push their code to a shared master branch as frequently as possible.

## Continuous Delivery

---

Have tooling in place so that you can push a button and deploy whatever's on master at any given time.

**NOTE:** this means that anything pushed to master needs to be release-ready!

## Continuous Deployment

---

Have tooling in place so that every push to master is *automatically* deployed to production.

---

# Continuous Integration: Example

Example: Rebecca and Arylee work on a project where they're continuously pushing commits to a shared git repository.

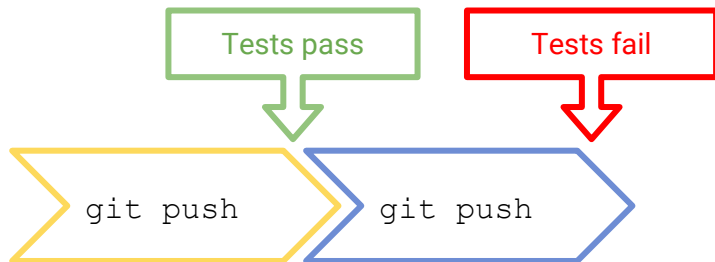


---

# Continuous Integration: Example

Example: Rebecca and Arylee work on a project where they're continuously pushing commits to a shared git repository.

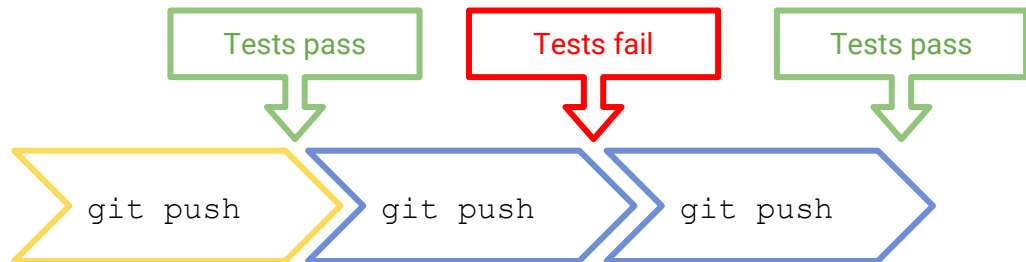
Each time either of them push, a tool runs all of the automated tests associated with the product. These tests validate that their code **integrated** well with the product.



# Continuous Integration: Example

Rebecca and Arylee work on a project where they're continuously pushing commits to a shared git repository.

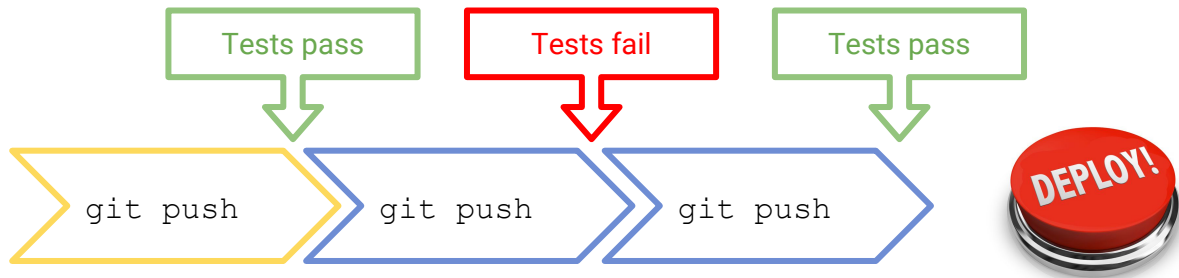
Each time either of them push, a tool runs all of the automated tests associated with the product. These tests validate that their code integrated well with the product. When a test fails, they know their next push needs to fix the tests.



# Continuous Deployment: Example

Once Arylee decides she needs to see her change live, she decides to deploy the code on the master branch.

The ability to do this at any time is called **Continuous Deployment**.

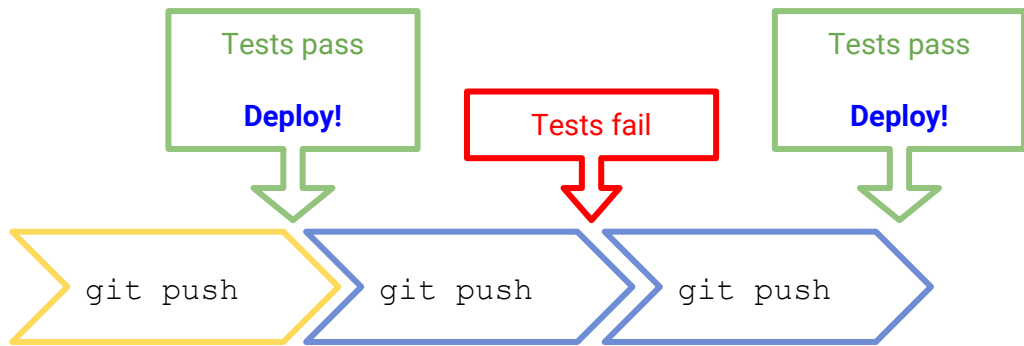




---

# Continuous Delivery: Example

If they automate their deployment so that every git push (with passing tests!) is deployed, that's **Continuous Delivery**.



Releasing  
more  
frequently  
makes us  
more agile



# What does Etsy do?

- Continuous Deployment
- Internal tool called “Deployinator” contains our Deploy button

The screenshot displays the Deployinator web interface. At the top, an orange header bar contains the 'Deployinator' logo in a stylized, outlined font. Below the header, a white sidebar on the left provides account information: 'Account: rsliter [logout](#)', 'Deploy Host: deploy.etsycorp.com is up', 'Stack: web' (with a dropdown arrow), 'Current Deploys: [Watch](#)', and 'Help: [Deployinator Runbook](#)'. The main content area has a grey top bar with version information: 'Version: Test: 97e8bfc6975821f9-20181024-004957-UTC | Princess: 97e8bfc6975821f9-20181024-004957-UTC | Production: 97e8bfc6975821f9-20181024-004957-UTC'. Below this, a '#push topic: salem (HO)' section includes a 'Watch' button. The main area is divided into two columns. The left column, titled '02. Deploy to Production', shows a commit hash '97e8bfc6975821f9' with a green arrow icon, a yellow box stating 'Tests are running.' with a bug icon, and a green 'Deploy to Production' button. Below this are links for 'What to watch after a push', 'Log Watcher (supergrep)', and 'Forums'. The right column, titled 'Deploy Logs', features a 'Deploy Logs' link and a list of deployment events. Each event includes a timestamp, environment (e.g., CONFIG, PRODUCTION, PRINCESS), user (e.g., jarnold, jwanga, agulati), and deployment details (old/new commit hashes). Each log entry has links for 'RUN LOG', 'DIFF', and 'DASHBOARD'.

Account: rsliter [logout](#)  
Deploy Host: deploy.etsycorp.com is up  
Stack: web  
Current Deploys: [Watch](#)  
Help: [Deployinator Runbook](#)

Version: Test: 97e8bfc6975821f9-20181024-004957-UTC | Princess: 97e8bfc6975821f9-20181024-004957-UTC | Production: 97e8bfc6975821f9-20181024-004957-UTC

#push topic: salem (HO) [Watch](#)

**02. Deploy to Production**  
97e8bfc6975821f9 →  
97e8bfc6975821f9  
**Tests are running.**  
[Deploy to Production](#)

What to watch after a push  
Log Watcher (supergrep)  
Forums

**Deploy Logs** [Deploy Logs](#)

web\_config – 2018-10-24 13:05:56 | **CONFIG** | jarnold | CONFIG Princess Deploy: old 5f6158946c5e4b5b, new: 7ea071e632164c69 [RUN LOG](#) | [DIFF](#)  
[DASHBOARD](#)

web – 2018-10-24 01:12:32 | **PRODUCTION** | jwanga | Production Deploy: old 25e1539b67e38623, new: 97e8bfc6975821f9 [RUN LOG](#) | [DIFF](#)  
[DASHBOARD](#)


web – 2018-10-24 00:56:32 | **PRINCESS** | jwanga | Princess Deploy: old: 25e1539b67e38623, new: 97e8bfc6975821f9 [RUN LOG](#) | [DIFF](#) [DASHBOARD](#)

web – 2018-10-23 22:55:13 | **PRODUCTION** | agulati | Production Deploy: old 5f6158946c5e4b5b, new: 25e1539b67e38623 [RUN LOG](#) | [DIFF](#)  
[DASHBOARD](#)

# Workshop

# Deploying to Heroku

Heroku, Inc. [US] | <https://www.heroku.com>



DEVELOPERS   CTOs   TEAMS   STARTUPS   ENTERPRISES   AGENCIES   STUDENTS

Developers, teams, and businesses of all sizes use Heroku to deploy, manage, and scale apps.

Whether you're building a simple prototype or a business-critical product, Heroku's fully-managed platform gives you the simplest path to delivering apps quickly. [Learn more](#) about how Heroku can benefit your app development.

# Why use Heroku?

- Enables us to release our code
- Free hosting
- Good tooling = relatively easy to use

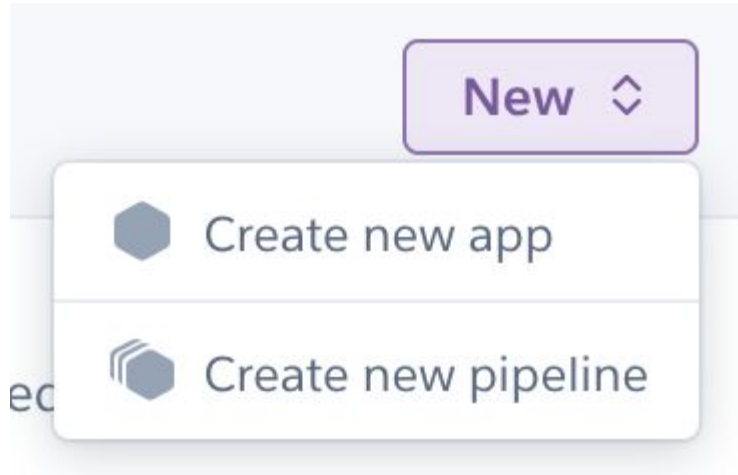
# Step One: Workshop Setup

 **TODO:** Install the Heroku command line tool on your laptop

 **TODO:** clone the workshop app with the following command:

```
$ git clone https://github.com/rsliter/csci-39549-calculator.git
```

# On Heroku, create a new app





# Follow the setup instructions...

Add this app to a pipeline

Create a new pipeline or choose an existing one and add this app to a stage in it.

Add this app to a stage



apps to  
between

We can ignore this :)

ected to GitHub can  
apps, and create apps  
uests. [Learn more.](#)

Deployment method



Heroku Git  
Use Heroku CLI

We'll do Heroku Git deploys

```
$ heroku logs --tail
```

```
2018-10-23T23:50:14.321206+00:00 heroku[router]: at=error  
code=H14 desc="No web processes running" method=GET path="/"  
host=sliter-calculator.herokuapp.com  
request_id=f6b6b24d-5460-4f1c-b48b-e06664ca2994  
fwd="68.173.122.251" dyno= connect= service= status=503  
bytes= protocol=https
```





https://stackoverflow.com/questions/19754262/flask-app-dont-start-on-heroku-server



stackoverflow



18



Is there actually a running dyno called `web` ? It looks like you might have forgotten to [scale your web dyno](#):

Add an entry like this in your Procfile:

```
heroku ps:scale web=1
```

You can use

```
heroku ps
```

to confirm that your `web` dyno is running.

[share](#) [improve this answer](#)

answered Nov 3 '13 at 16:30



[Lukas Graf](#)

16k ● 3 ● 47 ● 63

# Time to scale up!



HEROKU

Jump to Favorites, Apps, Pipelines, Spaces...



Personal



sliter-calculator



Open app

More

Overview

Resources

Deploy

Metrics

Activity

Access

Settings

Dynos

This app has no process types yet

Add a Procfile to your app in order to define its process types. [Learn more](#)



HEROKU

Jump to Favorites, Apps, Pipelines, Spaces...



Personal



sliter-calculator



Open app

More



Overview

Resources

Deploy

Metrics

Activity

Access

Settings

Dynos

This app has no process types yet

Add a Procfile to your app in order to define its process types. [Learn more](#)




**TODO: work with your teammates to write a Procfile that runs your app with one worker.**

How was that?



# Automating Deployment

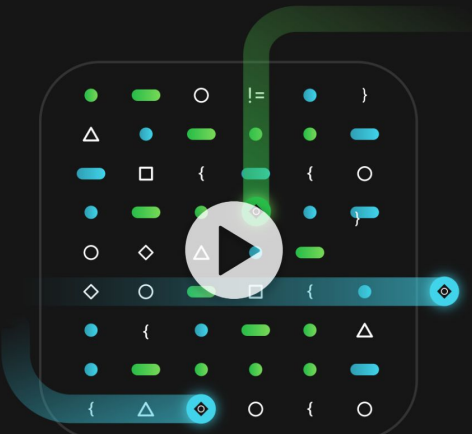
 [Product](#) [Pricing](#) [Enterprise](#) [Developers](#) [Company](#) [Contact Us](#) [Support](#) [Log In](#) [Sign Up](#)

Improve your engineering team's velocity with these 3 metrics. [Download Report](#) →

## The shortest distance from idea to execution.

Automate your development process quickly, safely, and at scale.

[Start Building for Free](#) [Explore the Docs](#)



# Why use CircleCI?

- Heroku gives us deployment, but it's manual
  - Requires each teammate to download Heroku, understand the tool
- CircleCI can automate this for us!



# How does CircleCI work?

- It does Continuous Integration!
  - Listens for changes on our git repo
  - Automatically runs tests (we don't have any yet)
  - Provides a DEPLOY button to release code to Heroku

# Let's try it out



**TODO: set up your app to release to Heroku from CircleCI**

# Let's try it out



**TODO: configure your app to release to Heroku from CircleCI**

A screenshot of the CircleCI website's Heroku integration page. The browser's address bar shows the URL `https://circleci.com/integrations/heroku/`. The navigation bar includes the CircleCI logo and links for Product, Pricing, Enterprise, Developers, Company, Contact Us, and Sign Up. Below the navigation bar, there are links for GitHub, Bitbucket, Heroku (which is highlighted with a blue underline), Sauce Labs, and AWS. The main content area has a dark background with the heading "Deploy to Heroku from CircleCI" in white. Below the heading is the text "Experience a simple, modern continuous delivery workflow now." and two buttons: "Start Building for Free" (a solid green button) and "Explore Heroku Docs" (a white button with a black border).

→ ↺ `https://circleci.com/integrations/heroku/` ☆

circleci Product Pricing Enterprise Developers Company Contact Us Sign Up

GitHub Bitbucket Heroku Sauce Labs AWS

## Deploy to Heroku from CircleCI

Experience a simple, modern continuous delivery workflow now.



[Start Building for Free](#) [Explore Heroku Docs](#)

How was that?



# Homework

# Homework: deploy your app!

-  **TODO: With your team, deploy your app to Heroku. Set up CircleCI for your project so that you don't need to manually deploy your app.**
-  **TODO: Don't forget that there is an assigned reading on testing for the next class! Check the syllabus for details. We will also have a guest lecturer!**