



# The DevSecOps Journey: Migrating Your Containerized Applications to the AWS Cloud Platform

Enabling customers to seamlessly implement security in a developer-friendly way

James Bland, Sr. Solutions Architect, AWS

Angel Rivera, Developer Advocate, Circle CI

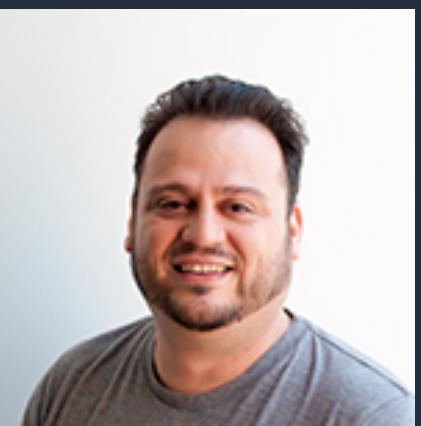
Jay Yeras, Head of Cloud & Cloud Native Solutions Architecture, Snyk



# Presenters



James Bland, AWS Sr. Partner Solutions Architect is a 25+ year veteran in the IT industry helping organizations from startups to ultra large enterprises achieve their business objectives. He has held various roles in software development, worldwide infrastructure automation and operations, and enterprise architecture. James has been practicing DevOps long before the term became popularized. He holds a doctorate in computer science with a focus on leveraging machine learning algorithms for scaling systems.



Angel Rivera, CircleCI's Developer Advocate, discovered his love for technology and software development at the start of his military career. Angel's passions are positive disruption, learning, teaching, mentoring but most of all inspiring all forms of technologists & building awesome tech communities.



Jay Yeras, Head of Cloud Solution Architecture at Snyk, brings over 20 years of experience in various technical roles and is a subject matter expert in building highly technical cloud architectures. His passion for efficiency, process driven development and automation permeates his family life as he is often found organizing kitchen cupboards for optimal use or setting up Alexa skills much to his wife's dismay.

# Table of contents

- Amazon's story (abbreviated)
- How to Learn More
- CircleCi
- Snyk Demo
- Q & A

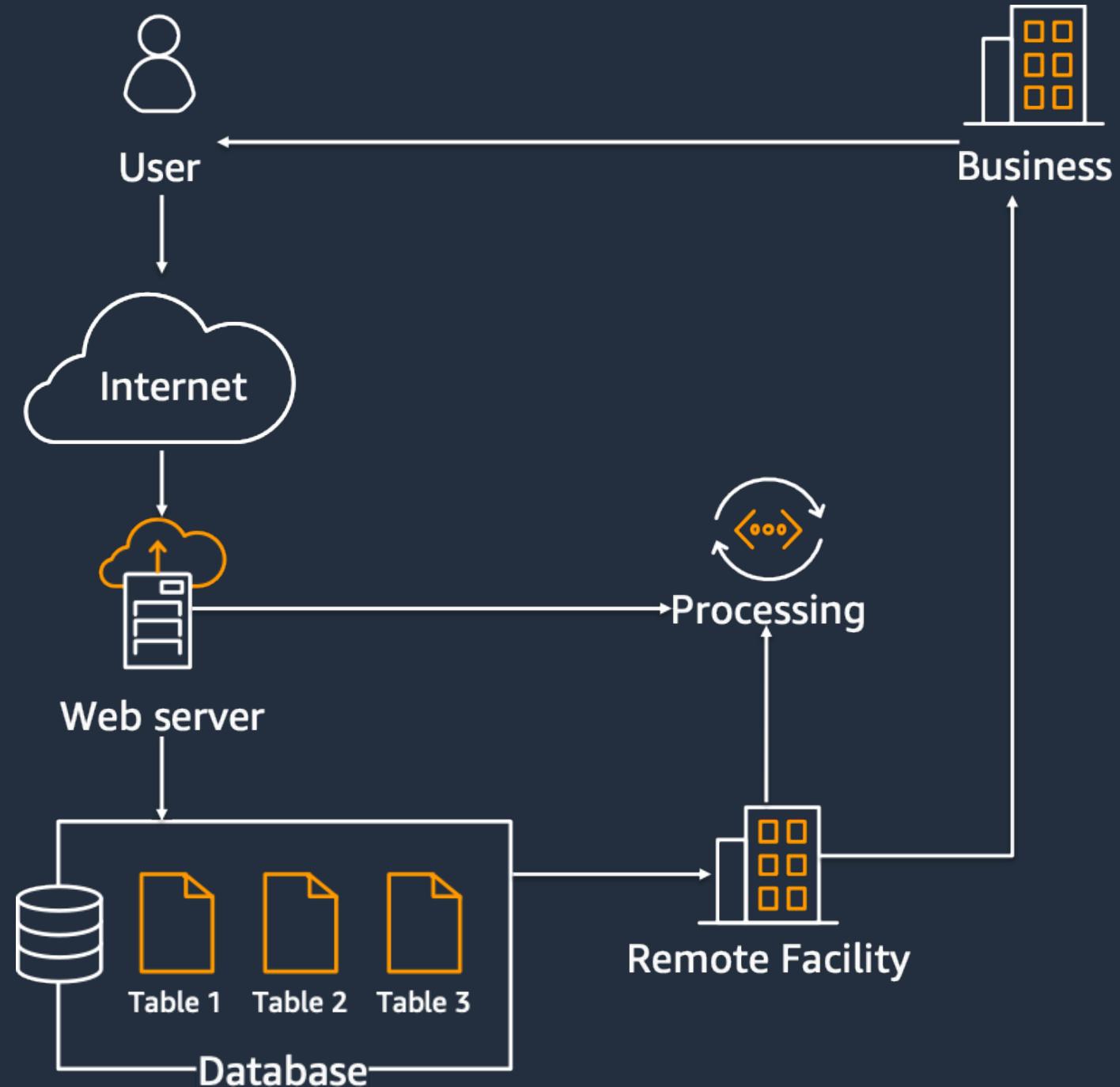
# Amazon's story

An abbreviated version

# Just starting out

This is how many web architectures started out, and it's how Amazon started too...

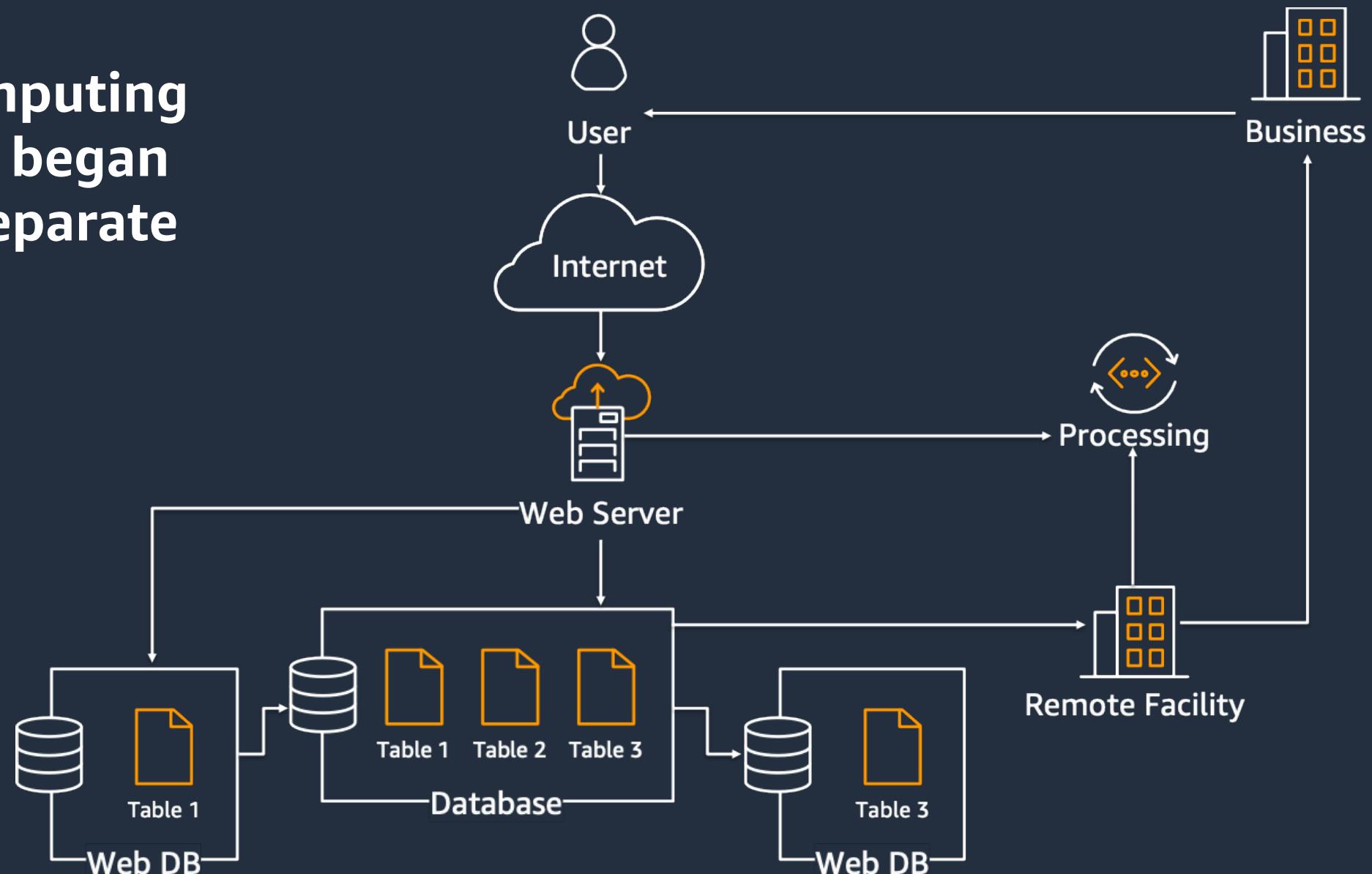
There are many bottlenecks, and scaling of the web server was an immediate factor



# Scaling v1

In 1998 the “Distributed Computing Manifesto” came out and we began breaking things down into separate components...

This was a bit better,  
still not very scalable

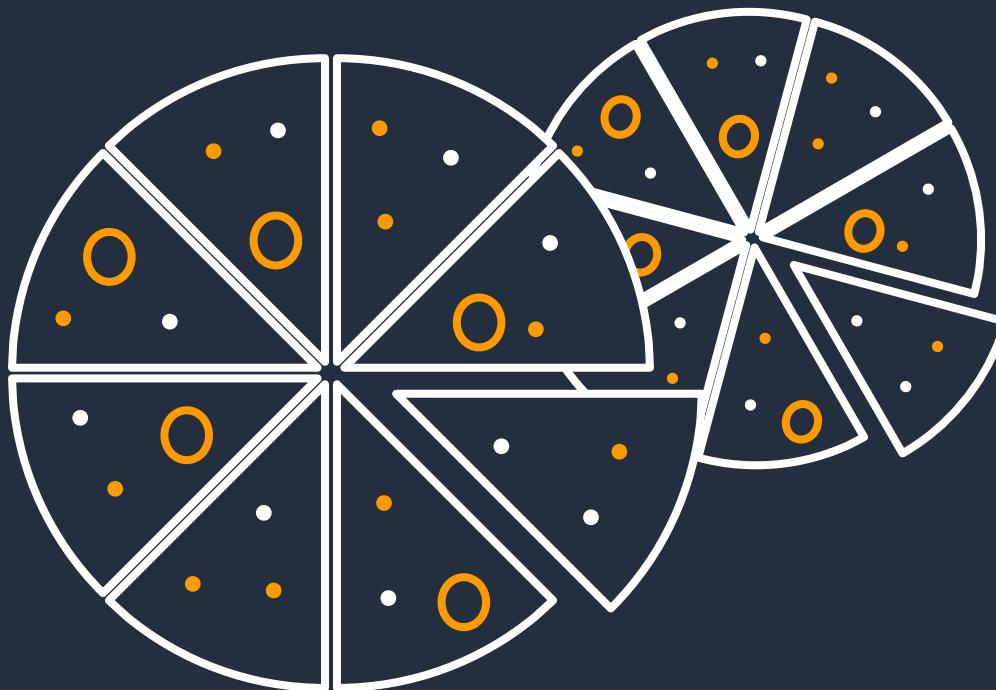


# Going further

## Principles

- **Make units as small as possible (Primitives)**
- **De-couple based on scaling factors, not functions**
- **Each service operates independently**  
**“Communication is terrible!” —Jeff Bezos**
- **APIs (contracts) between services**

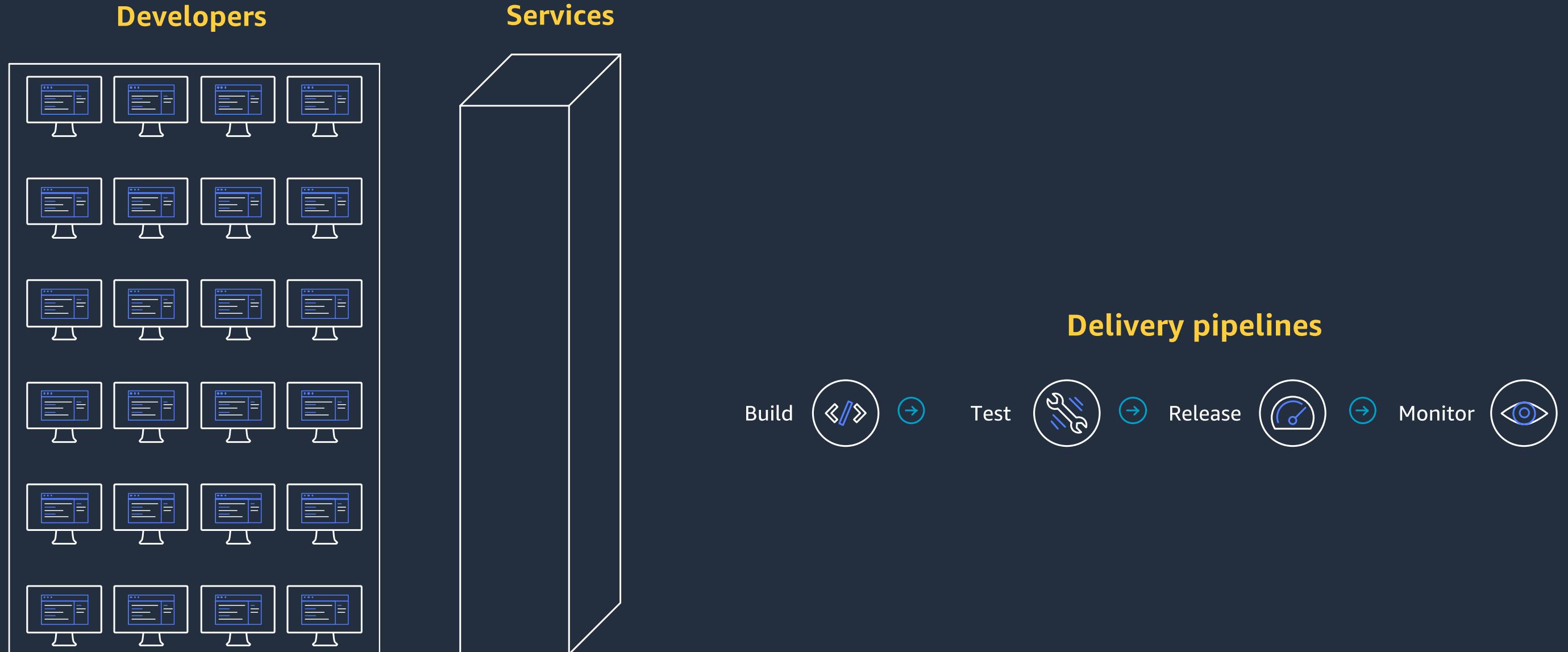
# Getting (re)organized



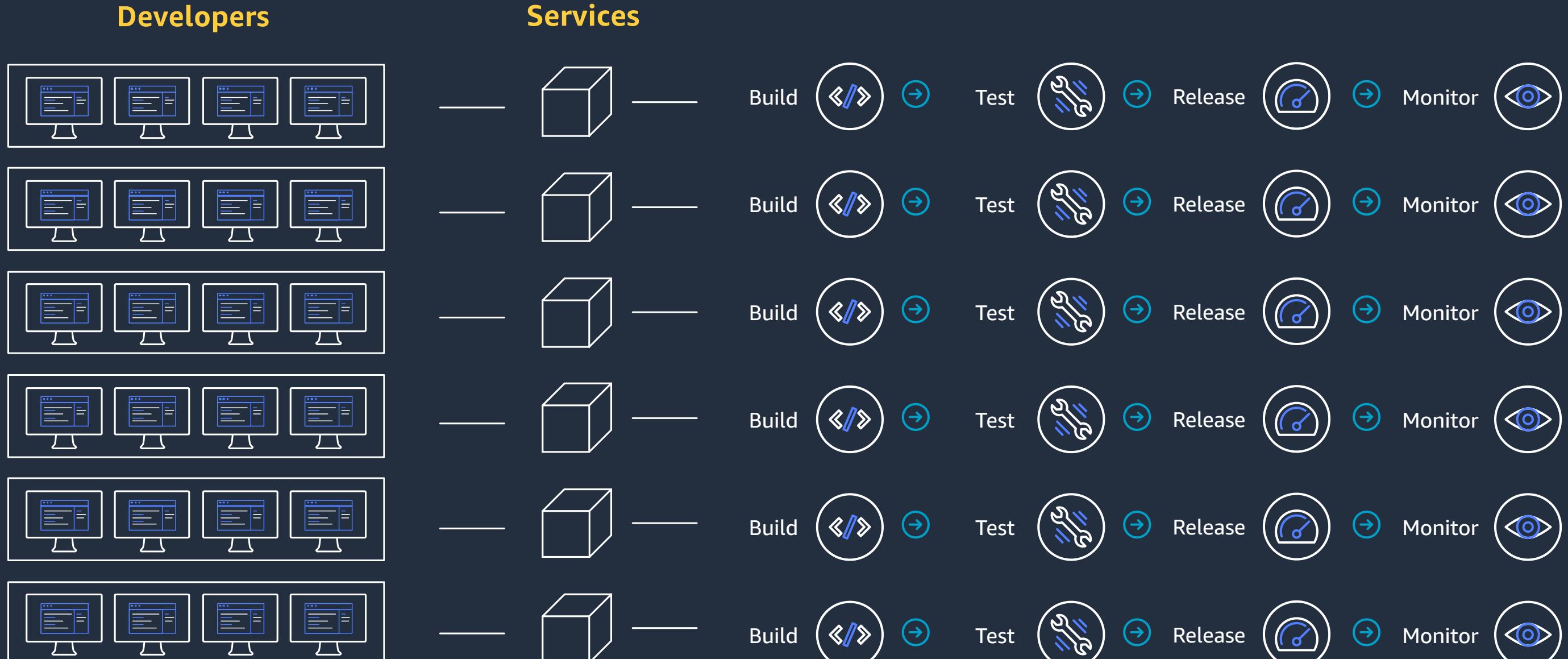
## “Two-pizza” teams

- Own a service
- Minimizes social constraints  
(Conway's law)
- Autonomy to make decisions

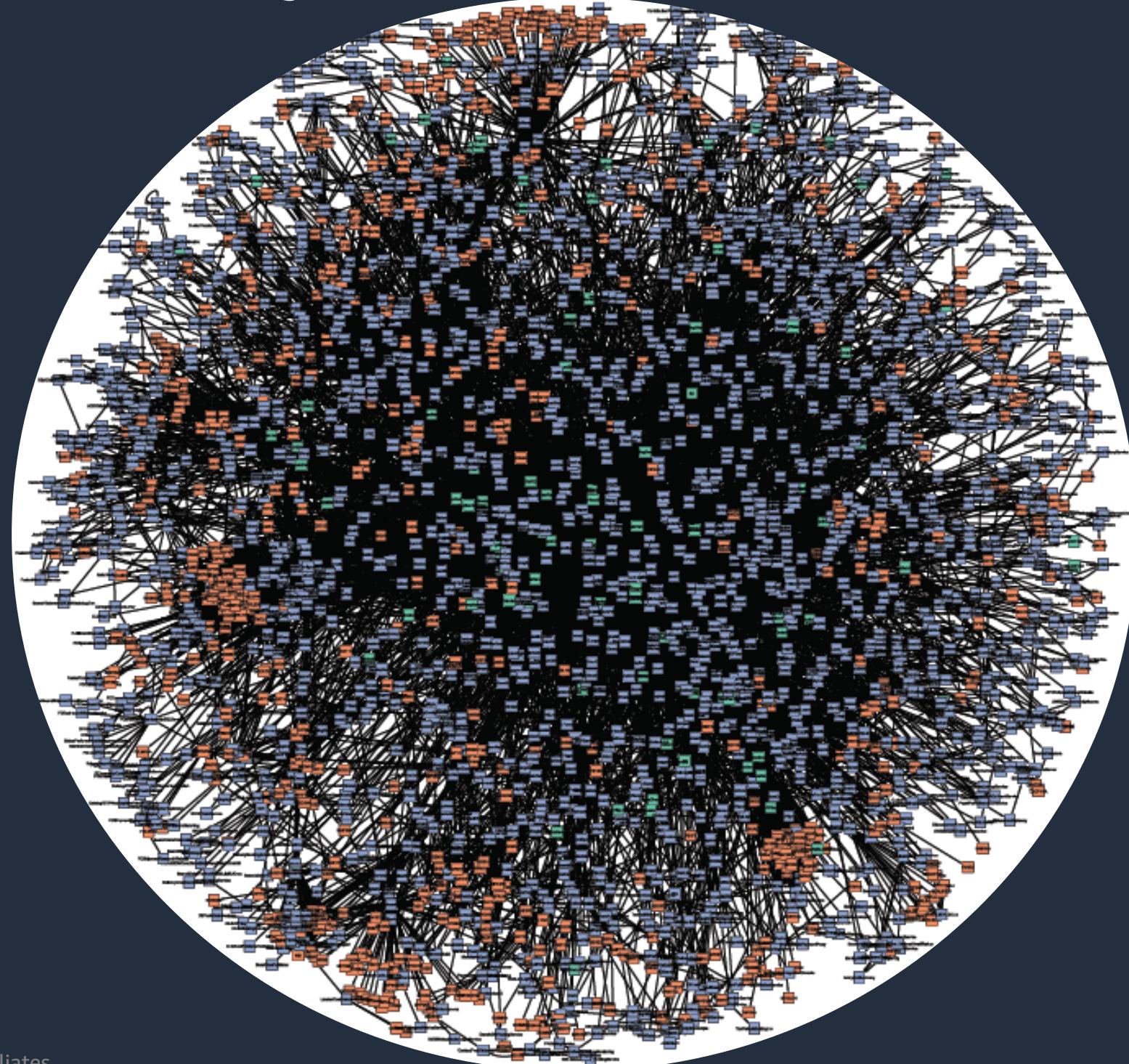
# Monolith development lifecycle



# Development lifecycle refactored for speed and agility



# Fast forward to today



# Want to learn more

# The Amazon Builders' Library

## How Amazon builds and operates software



Architecture, software delivery, and operations

---

By Amazon's senior technical executives and engineers

---

Real-world practices with detailed explanations

---

Content available for free on the website

# AWS Modernization Workshops

Curated workshops created by AWS Partners

<https://awsworkshop.io>



[Home](#)

Search

## Modernization Workshops

This website is a dedicated resource for curated workshops and training Modules created by the teams at AWS and AWS Partners. The workshops will teach you how to modernize various aspects of your business and provide you with detailed insight into what technology will drive this change.



### 10 workshops and growing

New workshops and content added all the time



### Created by experts

Created by specialists in the field from AWS and AWS Partners



### Available for access anywhere

Globally available without restriction and mobile friendly



© 2020, Amazon Web Services, Inc. or its Affiliates.





Our mission is to make it as easy as possible to go from idea to delivery.

# Introduction to CircleCI

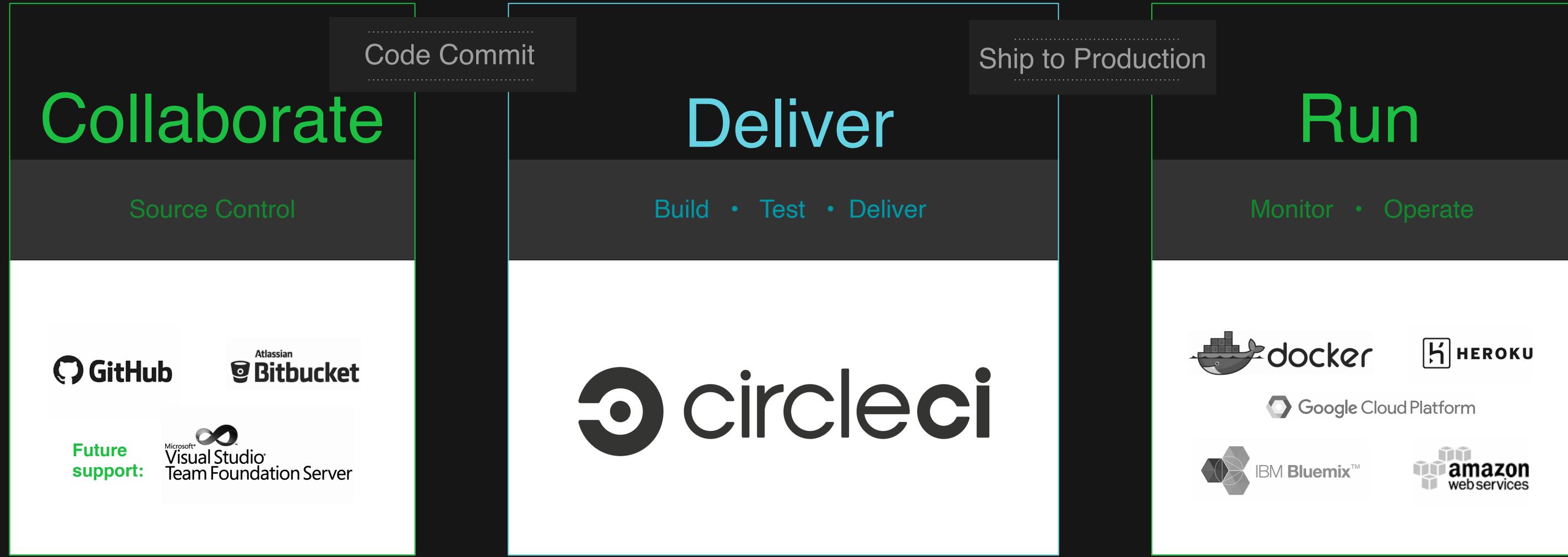
- Founded in 2011 with 280+ employees across 5 continents
- We help teams build better software, quicker, more reliably
- We re-architected the platform in 2017 and launched CircleCI 2.0
- Built for the Cloud - 1M+ builds per day run on our platform

# Where CircleCI sits in the toolchain:

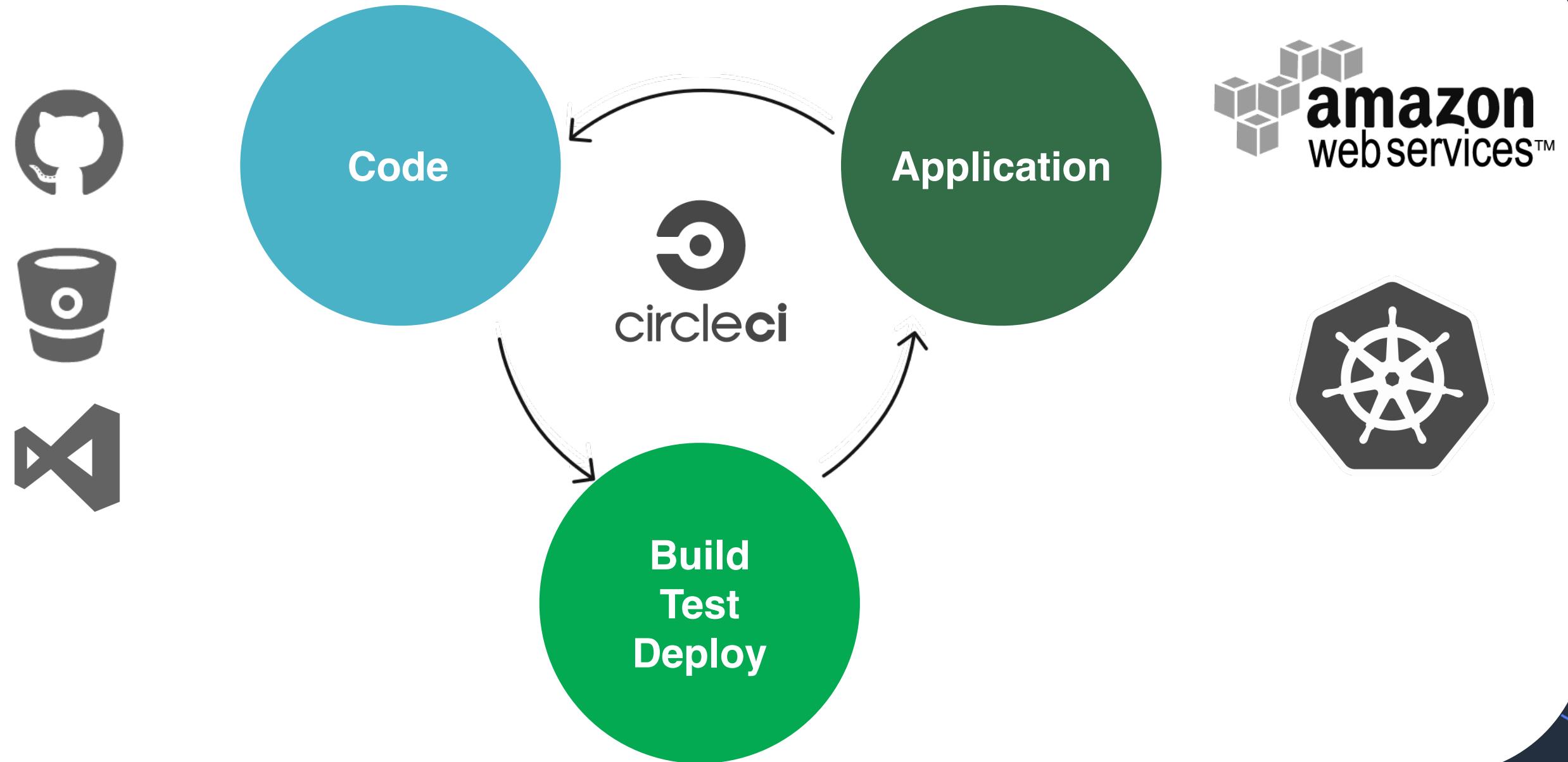
CREATION

ORCHESTRATION

LOGISTICS



# CircleCI automates process from code to production software



# Orbs. Config Reuse is Here.

---

/ôrb/ • noun

A reusable, shareable package of CircleCI configuration.

## What is an orb?

An orb is a package of CircleCI YAML configuration that contains predefined commands, executors, and jobs. Orbs are shareable across users, teams, and organizations, making it easy to keep your CircleCI configuration up-to-date and in-sync.

## orb.yml

```
1 version: 2.1
2 description: An orb that says hello
3 commands:
4   say-hello:
5     parameters:
6       to:
7         default: ${CIRCLE_USERNAME}
8         type: string
9     steps:
10      - run: echo "Hello << parameters.to >>"
11 executors:
12   default:
13     parameters:
14       tag:
15         default: latest
16         type: string
17     docker:
18       - image: bash:<< parameters.tag >>
19 jobs:
20   hello-build:
21     executor: default
22     steps:
23       - say-hello
```

## Commands

Commands are parameterized sets of steps that can be invoked as a step in a job.

## Executors

Executors define the runtime environment settings for executing jobs.

## Jobs

Jobs in orbs are invoked similarly to local jobs in config and can include parameters.



# CircleCI Pipeline Config using Orbs

```
version: 2.1
orbs:
  aws-ecr: circleci/aws-ecr@0.0.4

workflows:
  build_test_deploy:
    jobs:
      - build_test
      - docker_hub_build_push_image:
          requires:
            - build_test
      - aws-ecr/build_and_push_image:
          region: us-east-1
          account-url: ${AWS_ECR_ACCOUNT_URL}
          repo: ${CIRCLE_PROJECT_REPONAME}
          tag: ${CIRCLE_BUILD_NUM}
          requires:
            - build_test
jobs:
  build test:
    docker:
      - image: circleci/python:2.7.14
    steps:
      - checkout
      - run:
          name: Setup VirtualEnv
          command: |-
      - run:
          name: Run Tests
          command: |-
```

```
docker_hub_build_push_image:
  docker:
    - image: circleci/python:2.7.14
  steps:
    - checkout
    - setup_remote_docker:
        docker_layer_caching: false
    - run:
        name: Build and push Docker image to Docker Hub
        command: |
          echo 'export TAG=0.1.${CIRCLE_BUILD_NUM}' >> ${BASH_ENV}
          echo 'export IMAGE_NAME=${CIRCLE_PROJECT_REPONAME}' >> ${BASH_ENV}
          source ${BASH_ENV}
          docker build -t ${DOCKER_LOGIN}/${IMAGE_NAME} -t
                      ${DOCKER_LOGIN}/${IMAGE_NAME}:${TAG}
          echo ${DOCKER_PWD} | docker login -u ${DOCKER_LOGIN}
                               --password-stdin
          docker push ${DOCKER_LOGIN}/${IMAGE_NAME}
```

# CircleCI Orbs Registry

circleci Product Pricing Enterprise Developers Company Contact Us Support Go to app

## Explore Orbs

Orbs are shareable packages of CircleCI configuration you use in your builds.

[Read the docs](#) | [Learn more about Orbs](#)

NAME	DESCRIPTION	CERTIFIED	LAST MODIFIED
circleci/welcome-orb@0.3.1	Help new users get started building their projects with CircleCI.	CERTIFIED	Apr 10, 2019
circleci/aws-ecr@6.0.0	Build images and push them to the Amazon Elastic Container Registry. S...	CERTIFIED	May 22, 2019
circleci/aws-cli@0.1.13	Install and configure the AWS command-line interface (awscli)	CERTIFIED	Mar 27, 2019
circleci/slack@2.5.1	Easily integrate custom Slack notifications into your CircleCI projects. Cr...	CERTIFIED	May 22, 2019
circleci/aws-s3@1.0.10	A set of tools for working with Amazon S3. Requirements: bash Source: ...	CERTIFIED	May 29, 2019
codecov/codecov@1.0.4	Upload your coverage reports to Codecov without dealing with complex ...	PARTNER	Feb 19, 2019
circleci/hello-build@0.0.14	A simple "Hello, World!" orb	CERTIFIED	May 23, 2019
circleci/aws-ecs@0.0.8	An orb for working with Amazon Elastic Container Service (ECS)	CERTIFIED	Apr 21, 2019
cypress-io/cypress@1.7.0	Run your Cypress.io end-to-end browser tests without spending time con...	PARTNER	Apr 25, 2019
circleci/heroku@0.0.8	Install the Heroku CLI and deploy applications to Heroku. Source: https://...	CERTIFIED	May 22, 2019
circleci/node@1.0.1	Simplify common tasks for building and testing Node projects. Source: h...	CERTIFIED	Apr 03, 2019
circleci/gcp-cli@1.3.0	Install and configure the Google Cloud CLI (gcloud)	CERTIFIED	Mar 21, 2019

# Snyk Demo

© 2020, Amazon Web Services, Inc. or its Affiliates.



# Free self-paced workshop!

Solutions

Home

SNYK ACADEMY

Getting Started with Snyk

Snyk Open Source >

Snyk Container >

PATTERNS LIBRARY

Amazon Web Services >

CircleCI

Sign up for Snyk

Securing Kubernetes Workloads on AWS

Getting started

Create EKS cluster

CircleCI Configuration

config.yml >

Kubernetes manifests

CircleCI Project

Test deployment

Interpret scan results >

## Securing Kubernetes Workloads on AWS

The following exercises will walk you through building a CI/CD pipeline using CircleCI and various Orbs on the AWS Cloud. You will also be guided on how to leverage Snyk to scan each phase of your development process. Along the way, you will receive instruction on how to interpret scan results and the impact of these vulnerabilities through hands-on exercises. Lastly, you will receive guidance on how Snyk can help you fix these.

Let's get started!

# Free samples on GitHub!

The screenshot shows a GitHub repository page for `snyk-partners / snyk-circleci-eks`. The repository has 4 pull requests, 1 issue, 1 star, and 1 fork. The `Code` tab is selected. The file `config.yml` is displayed, showing configuration for CircleCI. The commit `ef3ece6` was made 28 days ago by `jayyeras`, adding project templates and samples.

```
version: 2.1

orbs:
  aws-eks: circleci/aws-eks@0.2.7
  aws-ecr: circleci/aws-ecr@6.8.2
  kubernetes: circleci/kubernetes@0.11.0
  snyk: snyk/snyk@0.0.10

defaults:
  docker:
    - image: circleci/node:9.11.2
  working_directory: ~/repo

jobs:
  test_app:
    <<: *defaults
    steps:
      - checkout
      - run:
```

# Workshop flow

1. Create an EKS cluster
2. Clone the sample Git repository
3. Create an AWS IAM user
4. Configure your CircleCI project
  - a. Define environment variables for AWS & Snyk
5. Enable Snyk integrations
6. Deploy application
7. Find vulnerabilities
8. Fix them! (Homework 😊)

# Create EKS cluster



# CircleCI project

 **Project Settings**  
snyk-circleci-eks

 Organization Settings X

[Overview](#)  
[Advanced](#)  
**Environment Variables**    
[SSH Keys](#)  
[API Permissions](#)  
[Jira Integration](#)  
[Slack Integration](#)

## Environment Variables

Environment variables let you add sensitive data (e.g. API keys) to your jobs rather than placing them in the repository. The value of the variables cannot be read or edited in the app once they are set.

Name	Value	Add Variable
ACCESS_KEY_ID_ENV_VAR_NAME	xxxxUW53	X
AWS_ECR_ACCOUNT_URL_ENV_VAR_NAME	xxxx.com	X
AWS_REGION_ENV_VAR_NAME	xxxxst-2	X
SECRET_ACCESS_KEY_ENV_VAR_NAME	xxxx0NhR	X
SNYK_TOKEN	xxxxb593	X

# config.yml

```
version: 2.1

orbs:
  aws-eks: circleci/aws-eks@0.2.7
  aws-ecr: circleci/aws-ecr@6.8.2
  kubernetes: circleci/kubernetes@0.11.0
  snyk: snyk/snyk@0.0.10
```

- CircleCI Certified & Partner Orbs
- Reduce complex lines of code

# Application scanning

```
scan_app:  
- snyk/scan:  
  fail-on-issues: false  
  monitor-on-build: true  
  project: '${CIRCLE_PROJECT_REPONAME}/${CIRCLE_BRANCH}-app'  
  severity-threshold: high  
  token-variable: SNYK_TOKEN  
target-file: ./submodules/goof/package.json
```

# Container image scanning

```
build_and_scan_image:  
- aws-ecr/build-image:  
  account-url: AWS_ECR_ACCOUNT_URL_ENV_VAR_NAME  
  dockerfile: Dockerfile  
  path: ./submodules/goof/  
  repo: ${CIRCLE_PROJECT_REPONAME}  
  tag: ${CIRCLE_SHA1}  
- snyk/scan:  
  docker-image-name: '$AWS*/${CIRCLE_PROJECT_REPONAME}:${CIRCLE_SHA1}'  
  project: '${CIRCLE*}/${CIRCLE_BRANCH}-container'  
  severity-threshold: high  
  target-file: ./submodules/goof/Dockerfile  
  token-variable: SNYK_TOKEN
```

# Push to ECR

```
build_and_push_image:  
- setup_remote_docker  
- aws-ecr/build-and-push-image:  
    account-url: AWS_ECR_ACCOUNT_URL_ENV_VAR_NAME  
    aws-access-key-id: ACCESS_KEY_ID_ENV_VAR_NAME  
    aws-secret-access-key: SECRET_ACCESS_KEY_ENV_VAR_NAME  
    region: AWS_REGION_ENV_VAR_NAME  
    repo: ${CIRCLE_PROJECT_REPONAME}  
    create-repo: true  
    checkout: true  
    dockerfile: Dockerfile  
    path: ./submodules/goof/  
    tag: ${CIRCLE_SHA1}
```

# Deploy to EKS

```
deploy_app:  
steps:  
- run:  
  command: |  
    BUILD_DATE=$(date '+%Y%m%d%H%M%S')  
    cat deployment/goof-deployment-template.yaml |\  
      sed "s|DOCKER_IMAGE_NAME|<< parameters.docker-image-name >>|g"\>  
      deployment/goof-deployment.yaml  
- kubernetes/create-or-update-resource:  
  resource-file-path: "deployment/goof-deployment.yaml"  
- kubernetes/create-or-update-resource:  
  resource-file-path: "deployment/goof-service.yaml"
```

# CircleCI pipeline run!

## Pipelines

Filters

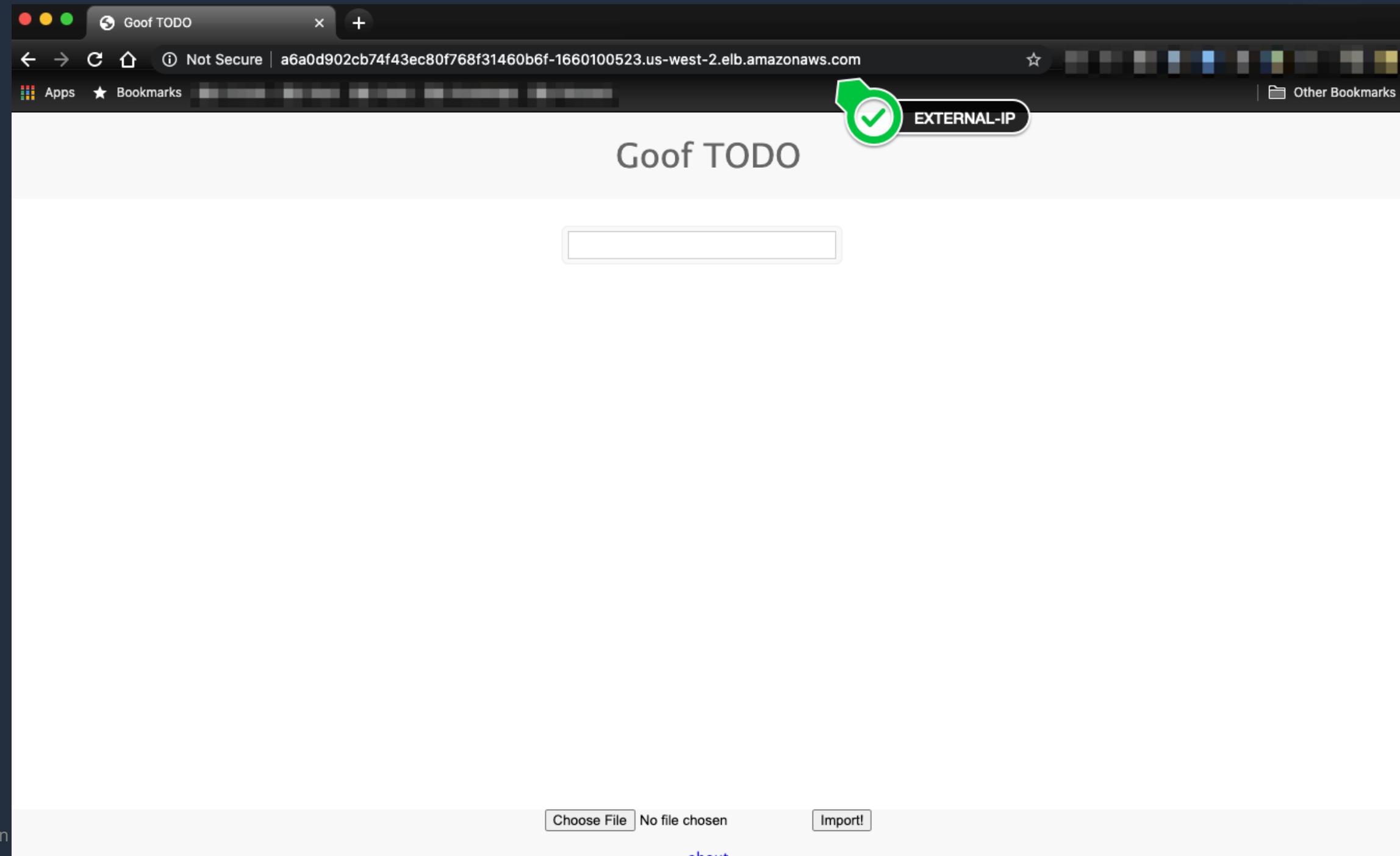
PIPELINE STATUS WORKFLOW BRANCH

PIPELINE	STATUS	WORKFLOW	BRANCH
#57	RUNNING	build_and_deploy	develop
	Success	test_app	
	Success	scan_app	
	Running	build_and_scan_image	
	Blocked	build_and_push_image	
	Blocked	deploy_app	

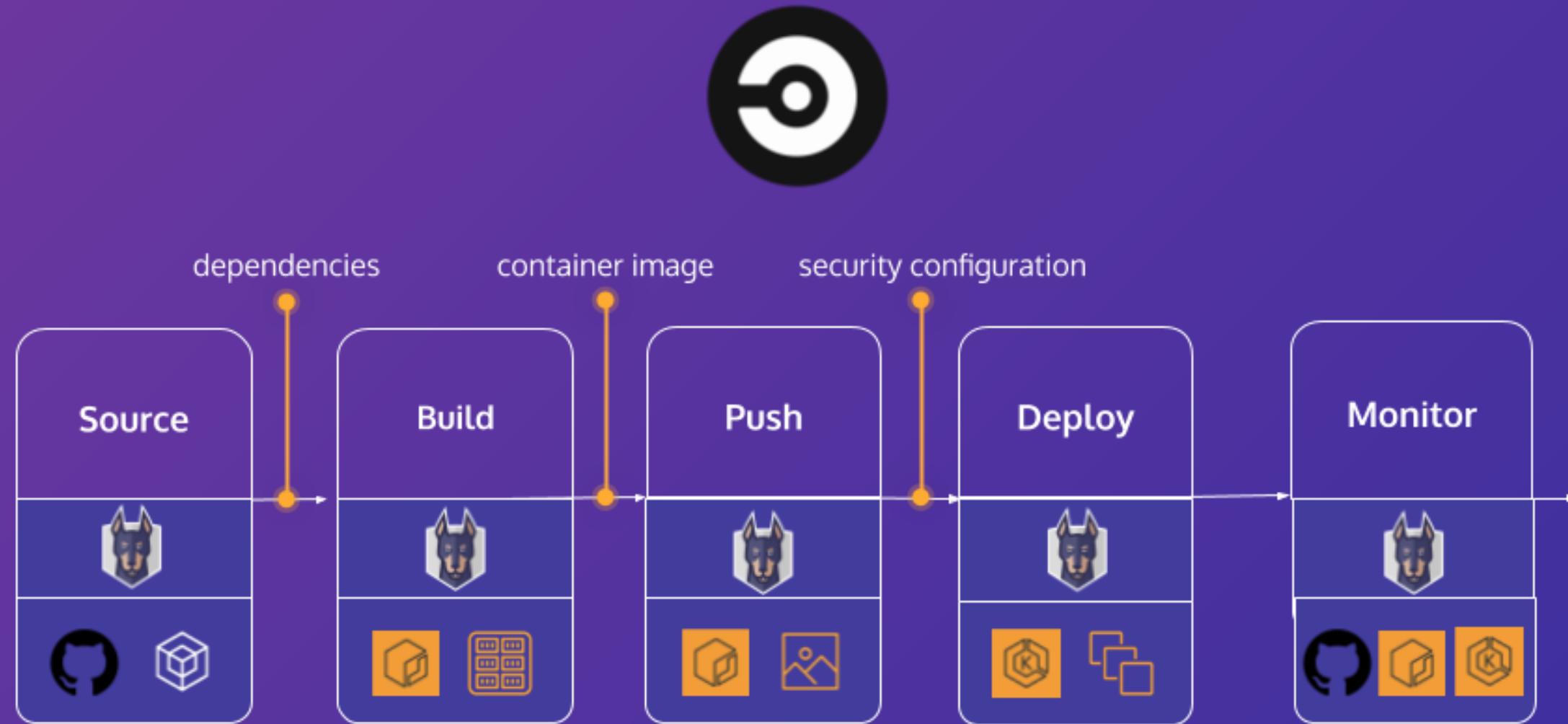
# Get external IP

1

# Our Goof application



# Remember our flow?



snyk

Dashboard Reports Projects Integrations Settings

5

## snyk-partners/goof:package.json

Snapshot taken by snyk.io 8 minutes ago.

Retest now

Vulnerabilities	54 via 246 paths
Taken by	Web
Repository	goof
Imported by	[REDACTED]
Dependencies	471
Created on	Fri 12th Jun 2020
Branch	master
Project owner	<a href="#">+ Add a project owner</a>

New: Keep your project healthy and enable automatic dependency upgrades now. [Learn more](#)

Issues Remediation Dependencies Runtime

Search issues...

Choose how to fix these vulnerabilities and open a pull request.

[Open a fix PR](#)

Issue type

Vulnerabilities (53)

License issues (1)

Severity

High (26)

HIGH SEVERITY EXPLOIT: MATURE ?

Arbitrary File Write via Archive Extraction (Zip Slip)

© 2020, Amazon

aws

# Application scan

snyk

Dashboard Reports Projects Integrations Settings

New vulnerabilities for this package have been disclosed. Run `snyk wizard` to explore remediation options.

## snyk-circleci-eks/develop-app

Overview History Settings

Snapshot taken by cli 2 hours ago. Retest now

Vulnerabilities	54 via 246 paths	Dependencies	471	Source	CI/CLI
Taken by	CI/CLI	Created on	Tue 12th May 2020	Hostname	652065c493f7
Runtime	v8.16.2	Imported by	 circleci	Project owner	 Add a project owner

Issues Remediation Dependencies Runtime

Search issues...

Vulnerabilities (53)

License issues (1)

High (26)

Medium (23)

Low (5)

**HIGH SEVERITY EXPLOIT: MATURE ?**

 Arbitrary File Write via Archive Extraction (Zip Slip)

Vulnerable module: `adm-zip@0.4.7`  
Introduced through: `adm-zip@0.4.7`  
Exploit maturity: Mature  
Fixed in: 0.4.11

**Detailed paths and remediation**

- Introduced through: `goof@1.0.1 > adm-zip@0.4.7`  
Remediation: Upgrade to `adm-zip@0.4.11`

© 2020, Amazon Web Services, Inc. or its affiliates. All rights reserved.

aws

# Container registry scan

snyk

Dashboard Reports Projects Integrations Settings

⌚ snyk-circleci-eks:0900d605289b0a9e1b6e940065001c490f7a7ca5 Overview History Settings

Snapshot taken by snyk.io a few seconds ago. Retest now

Vulnerabilities	726 via 7680 paths	Dependencies	414	Source	ECR
Taken by	Web	Created on	Tue 9th Jun 2020	Target OS	debian:9
Image tag	0900d605289b0a9e1b6e940065001c490f7a7ca5	Base image	node:6-stretch	Imported by	
Project owner	<a href="#">+ Add a project owner</a>				

Recommendations for base image upgrade

	BASE IMAGE	VULNERABILITIES	SEVERITY
Current image	node:6-stretch	726	<span>27 H</span> <span>172 M</span> <span>527 L</span>
Major upgrades	node:10-stretch	656	<span>16 H</span> <span>141 M</span> <span>499 L</span>

Show more upgrade types

© 2020, Amazon Web Services, Inc. or its Affiliates.

aws

# Kubernetes scan

snyk

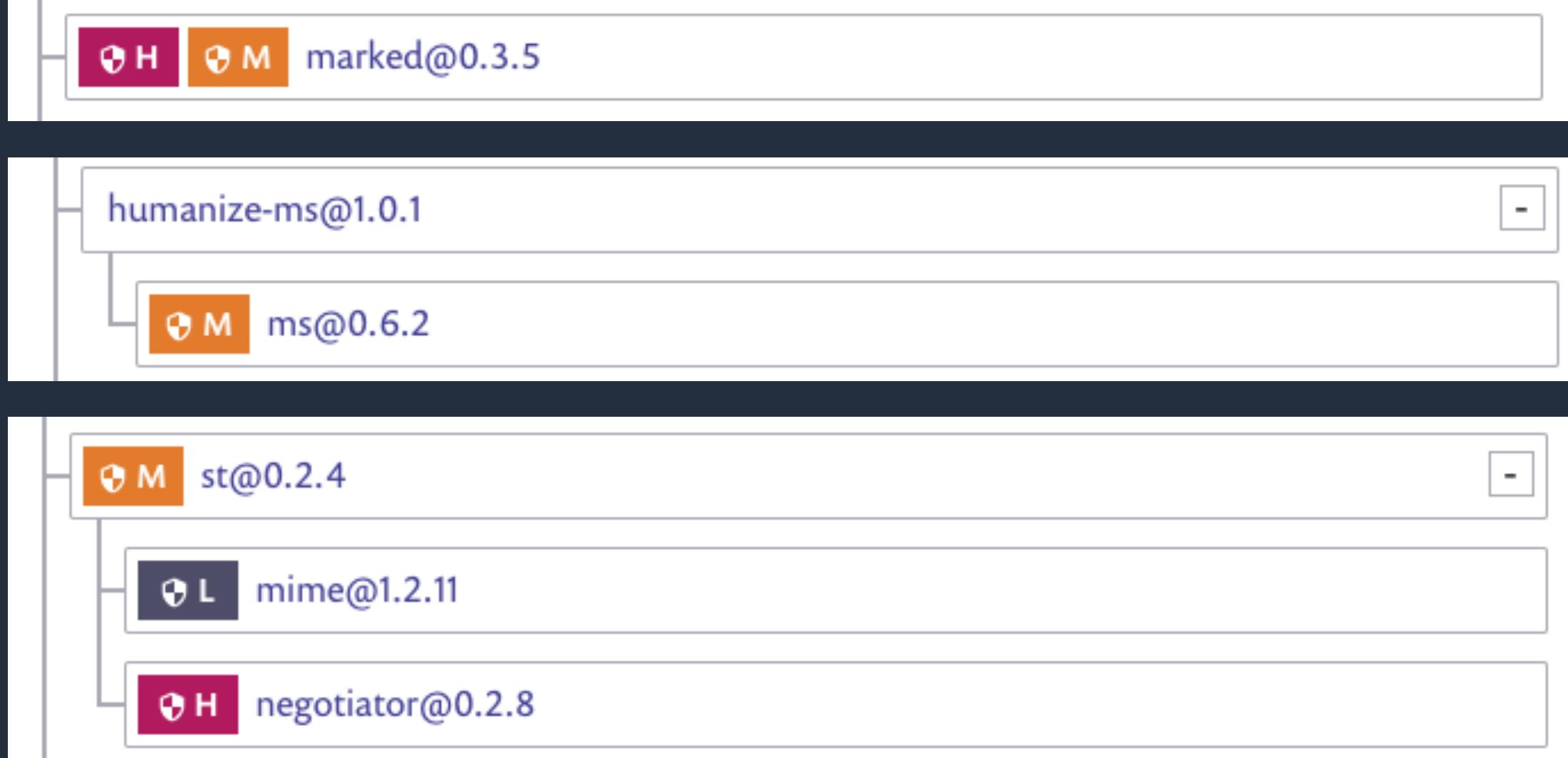
5

Dashboard Reports Projects Integrations Settings

## default/deployment.apps/goof

Vulnerabilities	716 via 7635 paths	Kind	Deployment	Secure configuration	Privileged	PASS	?
Dependencies	414	Cluster	snyk-circleci-eks		CPU limits	FAIL	?
Source	 Kubernetes	Namespace	default		Memory limits	FAIL	?
		Revision	2		Run as non-root	FAIL	?
					Read-only root file system	FAIL	?
					Drop capabilities	FAIL	?

# Three example vulnerabilities



# Static pages & the st package vulnerability



# Directory traversal

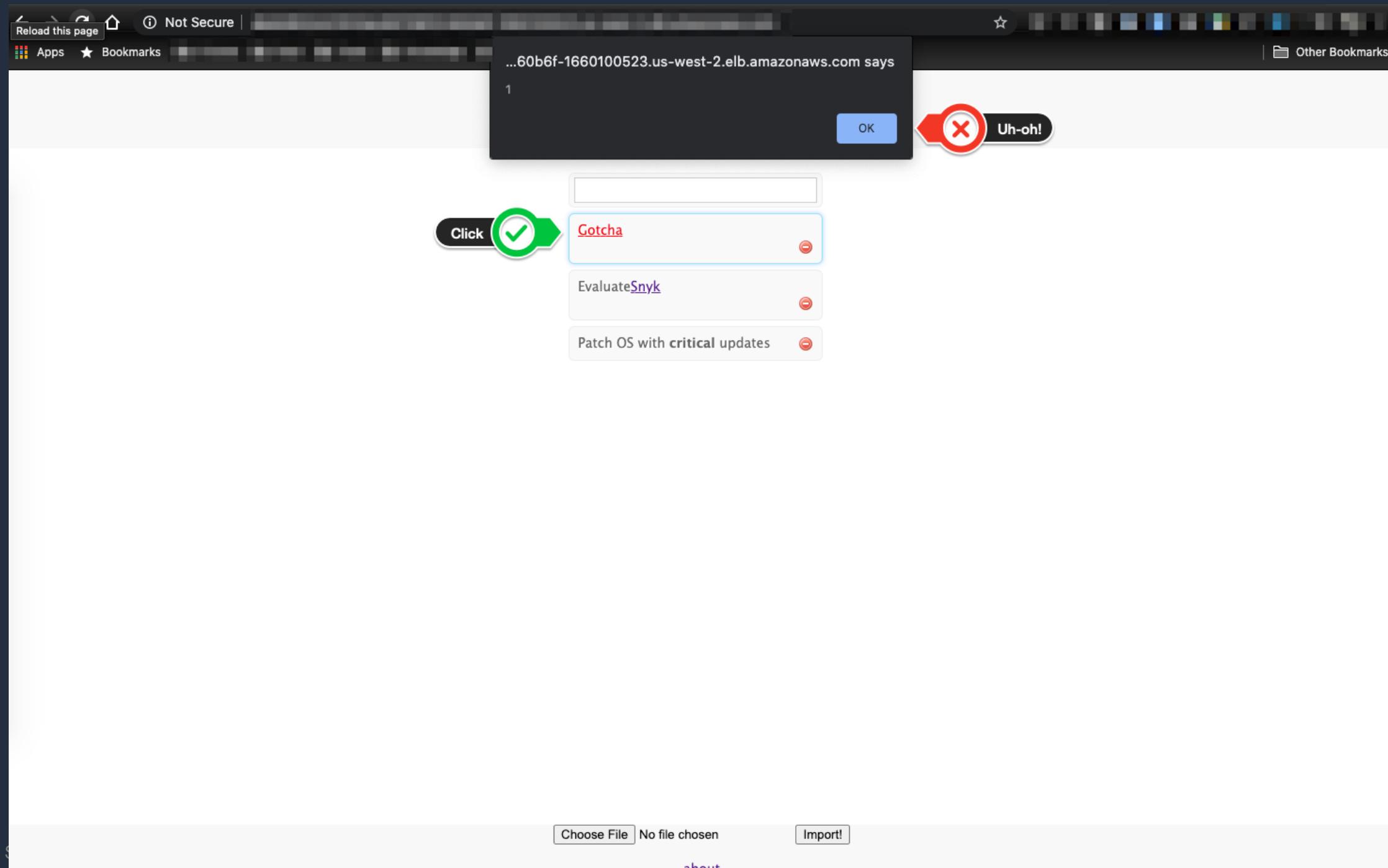
1

# Markdown with marked package

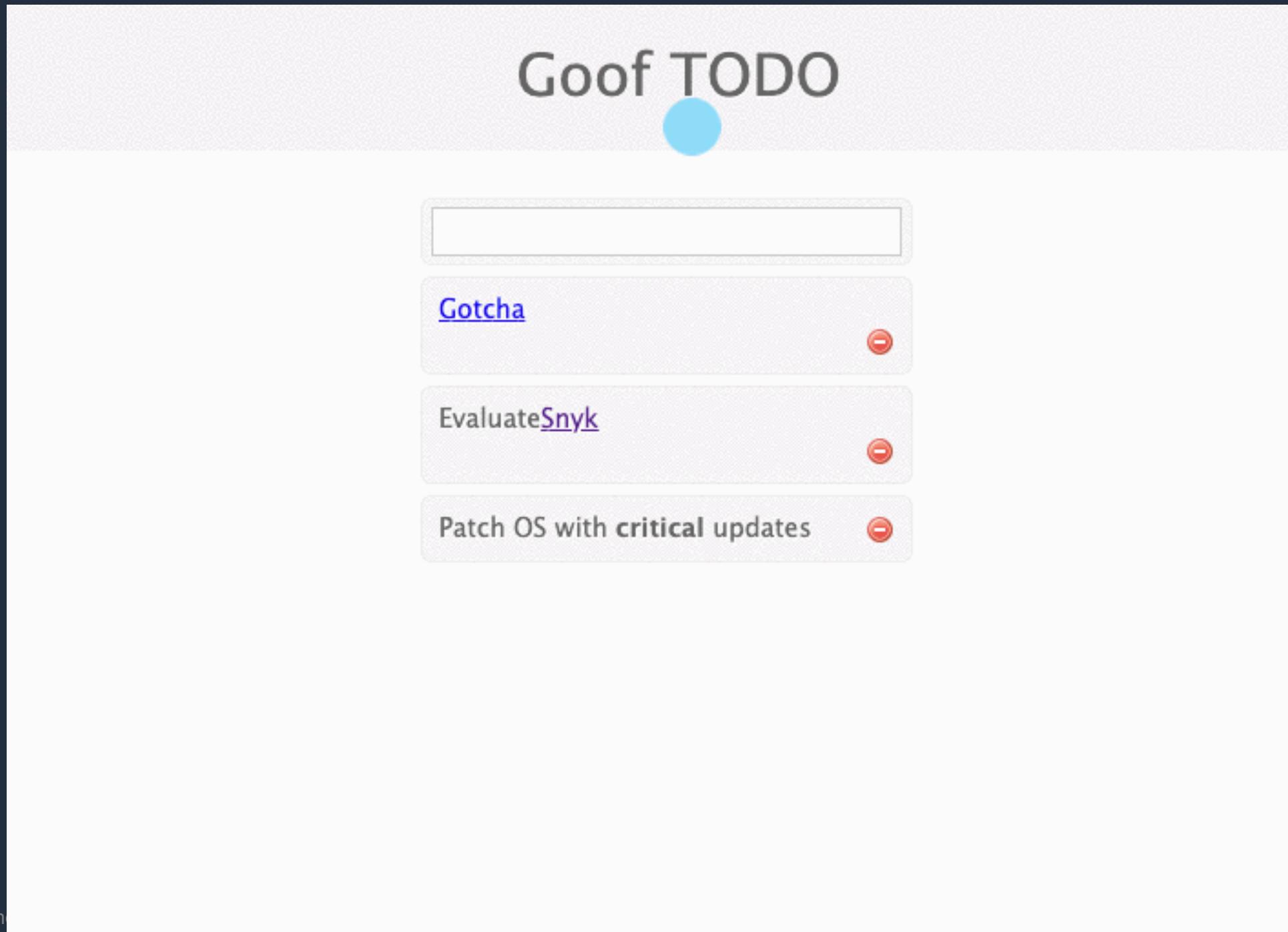
Goof TODO



# Cross-site Scripting (XSS)



# Time format conversion with ms package



# (ReDos) Regular Expression Denial of Service

```
echo 'content=Reboot server in \'printf  
">%..0s5" {1..60000}\' minutea' | http --  
form $GOOF_HOST/create -v
```

Goof TODO

- 
- Reboot server [20m] -
- Apply updates [30m] -
- [Gotcha](#) -
- Evaluate[Snyk](#) -
- Patch OS with **critical** updates -

# What did we learn?

- It's easy to build security into your workflows.
- Secure your pipeline end to end.
- Monitor configuration drifts that may expose you to potential attacks.
- Finding vulnerabilities is great, fixing them before it's a problem is better!

# Where do you go to learn more?

<https://solutions.snyk.io/>

# Q&A

# Thank You