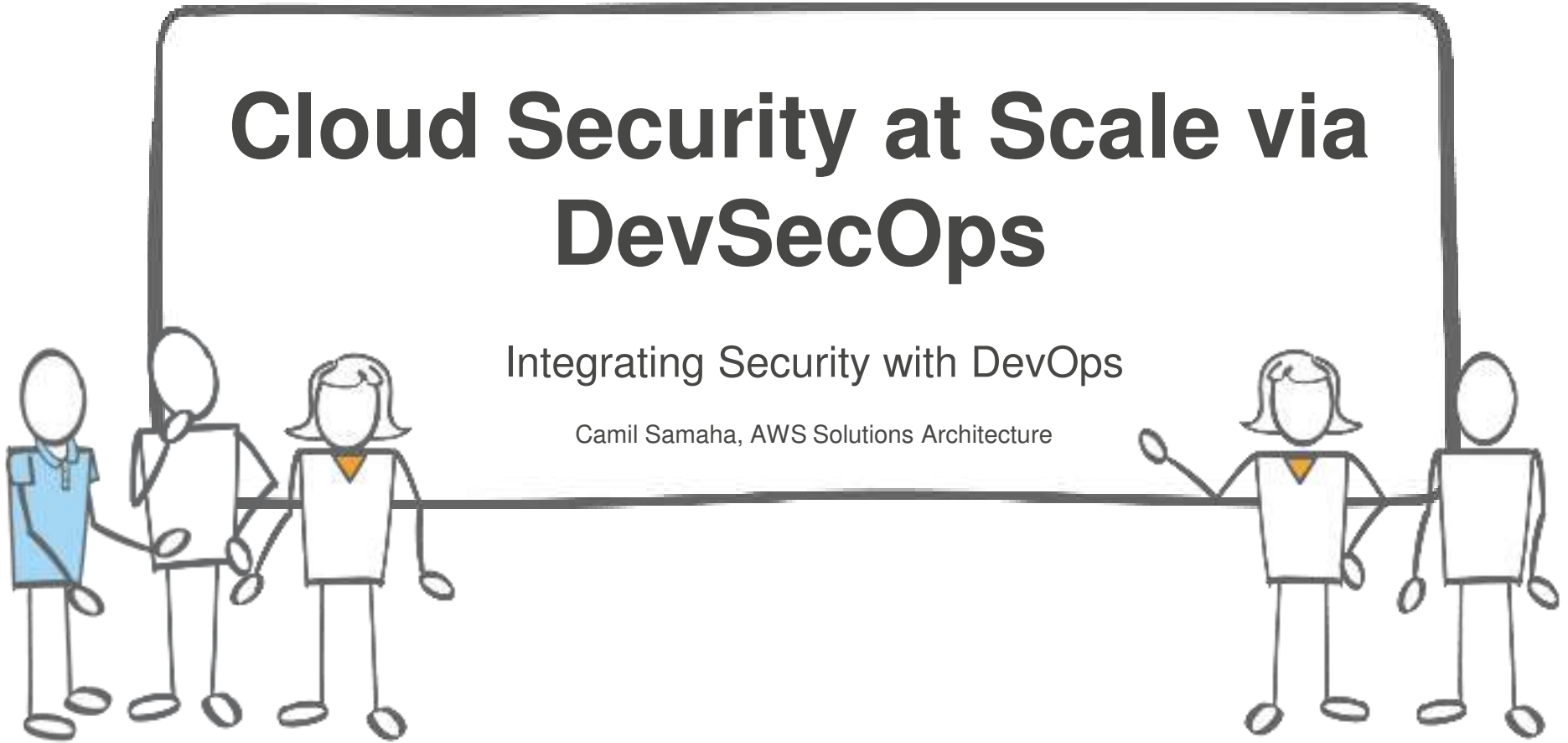
Cloud Security at Scale via DevSecOps



Integrating Security with DevOps

Camil Samaha, AWS Solutions Architecture

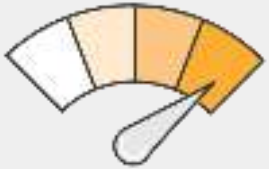
What Is (True) Cloud Computing?

The on-demand delivery of IT resources over public or private networks with zero up-front costs, no long-term contracts, and pay-as-you-go pricing

AWS Global Infrastructure

New Business Model

Focus on differentiating your company



Innovate at start-up like speed

Reduce risk

Move Fast

OR

Stay Secure

Move Fast

AND

Stay Secure

Infrastructure Evolution

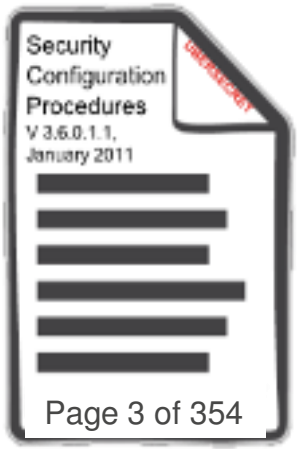
Then

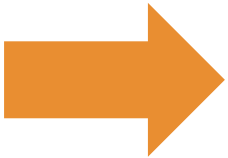
- Big Perimeter
- End-to-End Ownership
- Build it all yourself
- Server-centric approach
- Self-managed Services
- Static Architecture
- De-centralized Administration

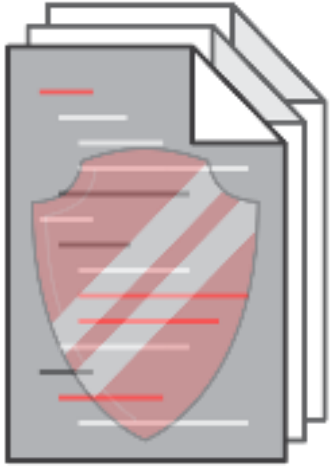
Now

- Micro-Perimeters
- Own just enough
- Focus on your core value
- Service-Centric
- Platform Services
- Continuously Evolving
- **Control Plane API**

Security Evolution







Security as code

Security as Code

1. Use the cloud to protect the cloud
2. Security infrastructure should be cloud aware
3. Expose security features as services via API
4. Automate everything so everything scales

What is DevOps?

Cultural Philosophy  Practices  Tools

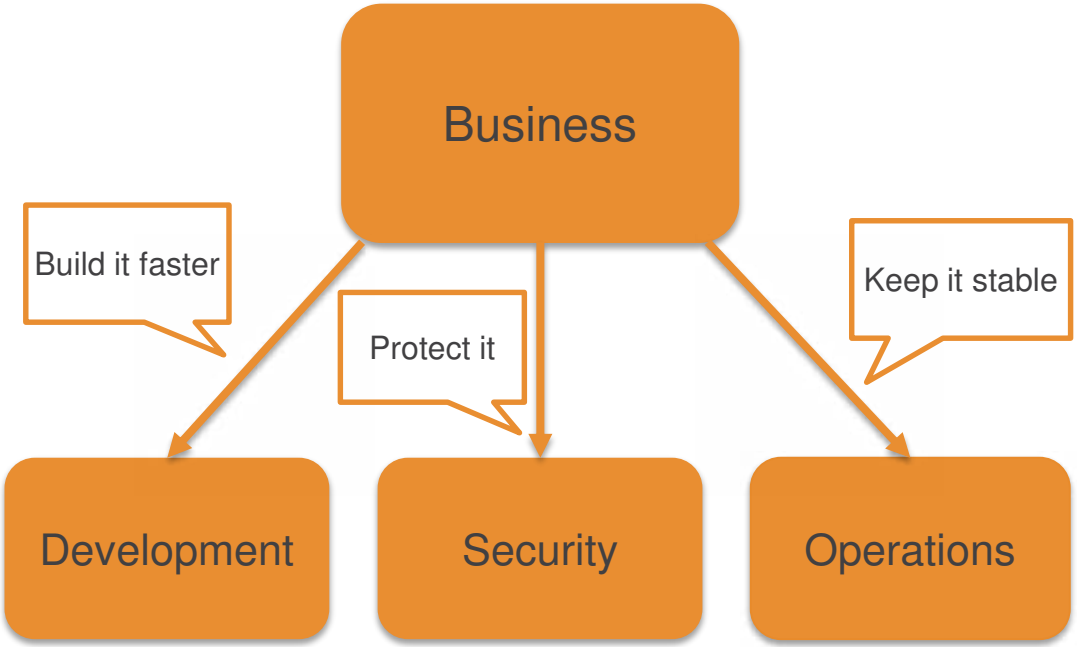
What is DevOps?

- Breakdown the barriers
- Work as one team end to end
- Support business and IT agility
- Collaboration & Communication
- Treat Infrastructure as code
- Automate everything
- Test, measure & monitor everything

Culture

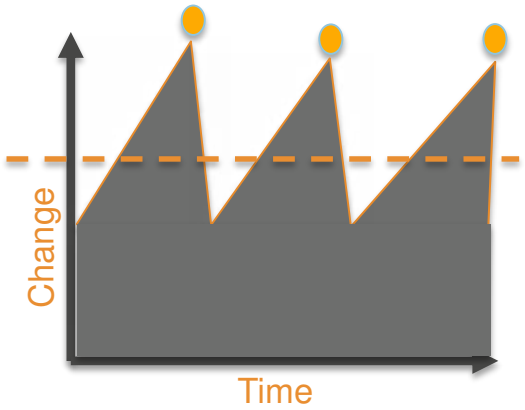
Technology

Security as Code: Innovation, Stability & Security



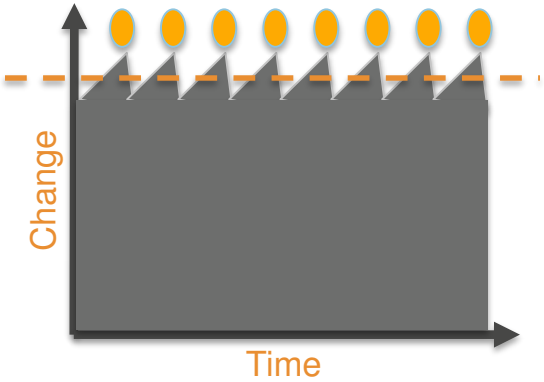
Security as Code: Deploying More Frequently Lowers Risk

Rare release events:
"Waterfall methodology"



"Increased risk"

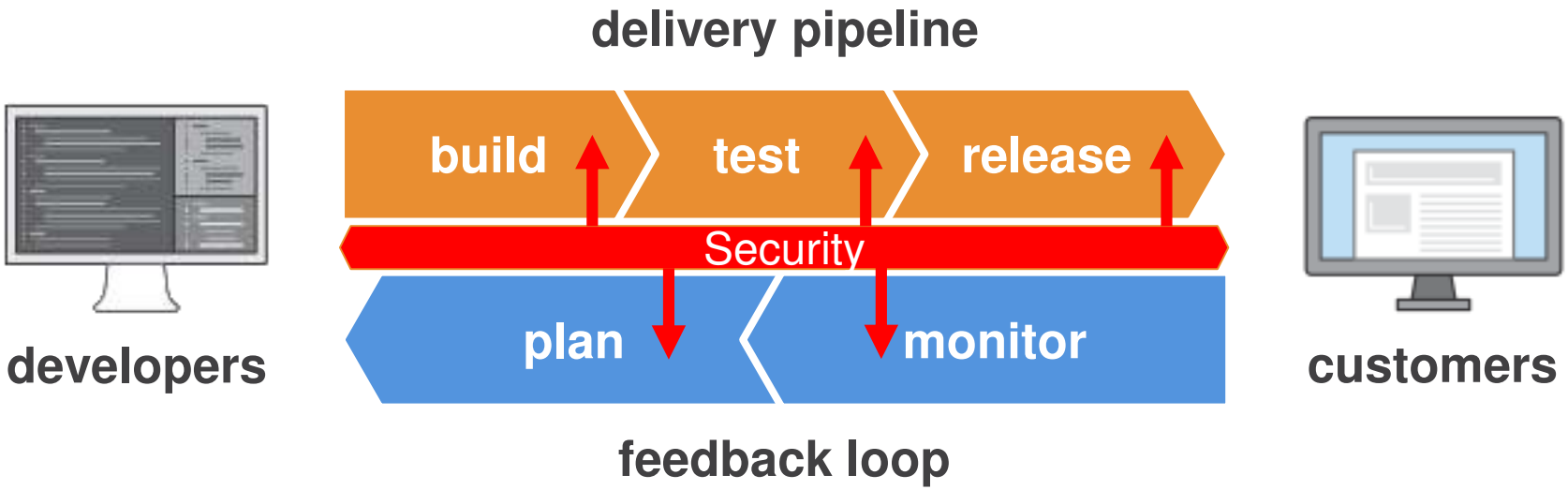
Frequent release events
"Agile methodology"



"Minimized risk"

What is DevSecOps?

Software development lifecycle

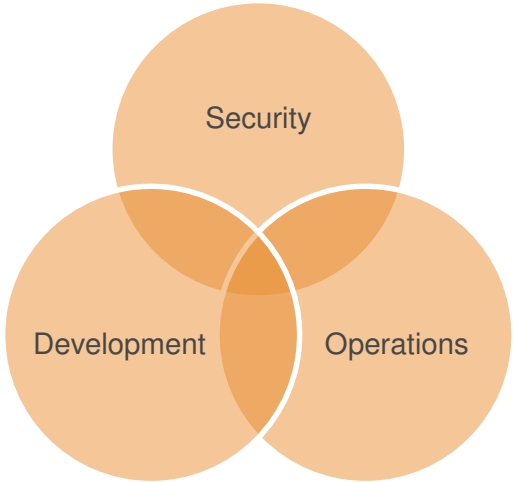


DevOps = Efficiencies that speed up this lifecycle
DevSecOps = Validate building blocks without slowing lifecycle

Who is DevSecOps?

DevSecOps is

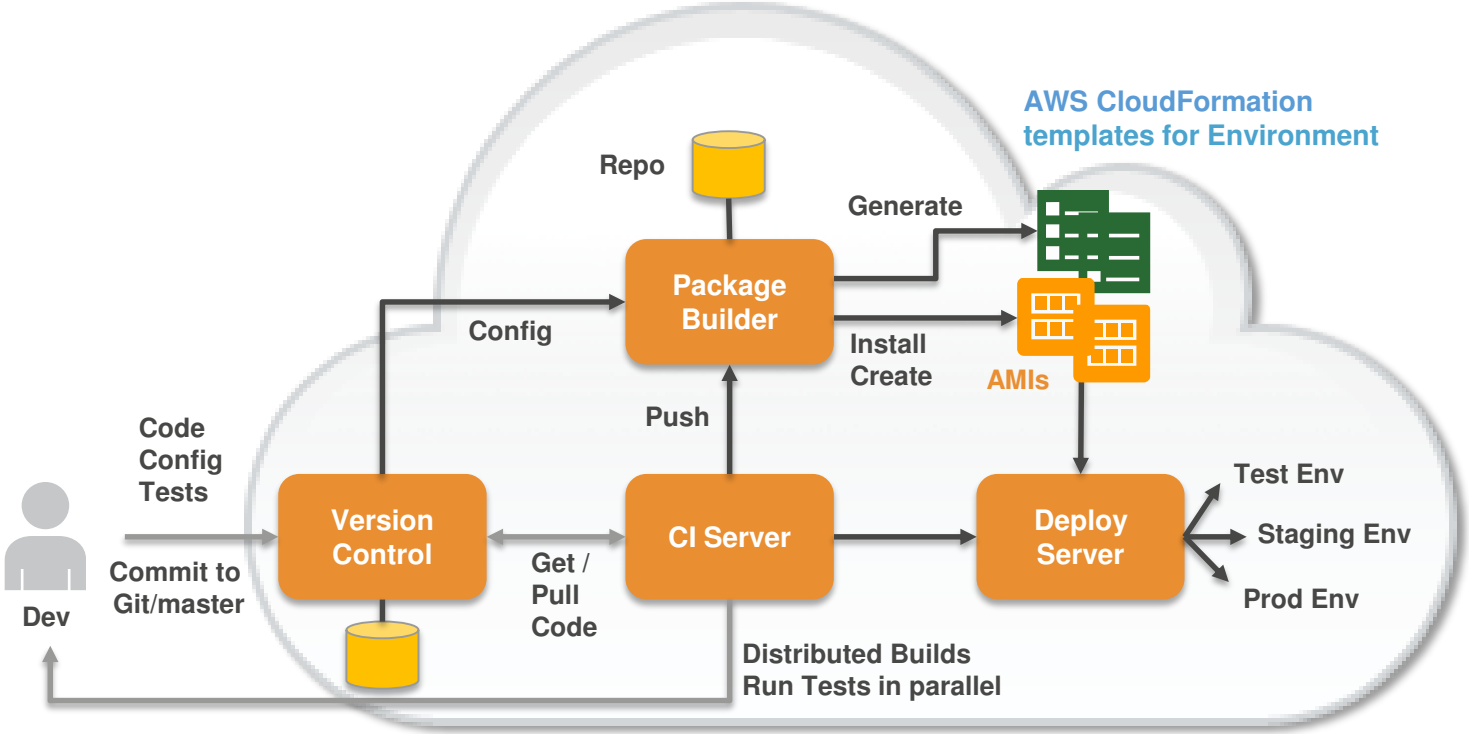
- Team/Community, not a person
- Automated and autonomous security
- **Security at scale**



DevSecOps role

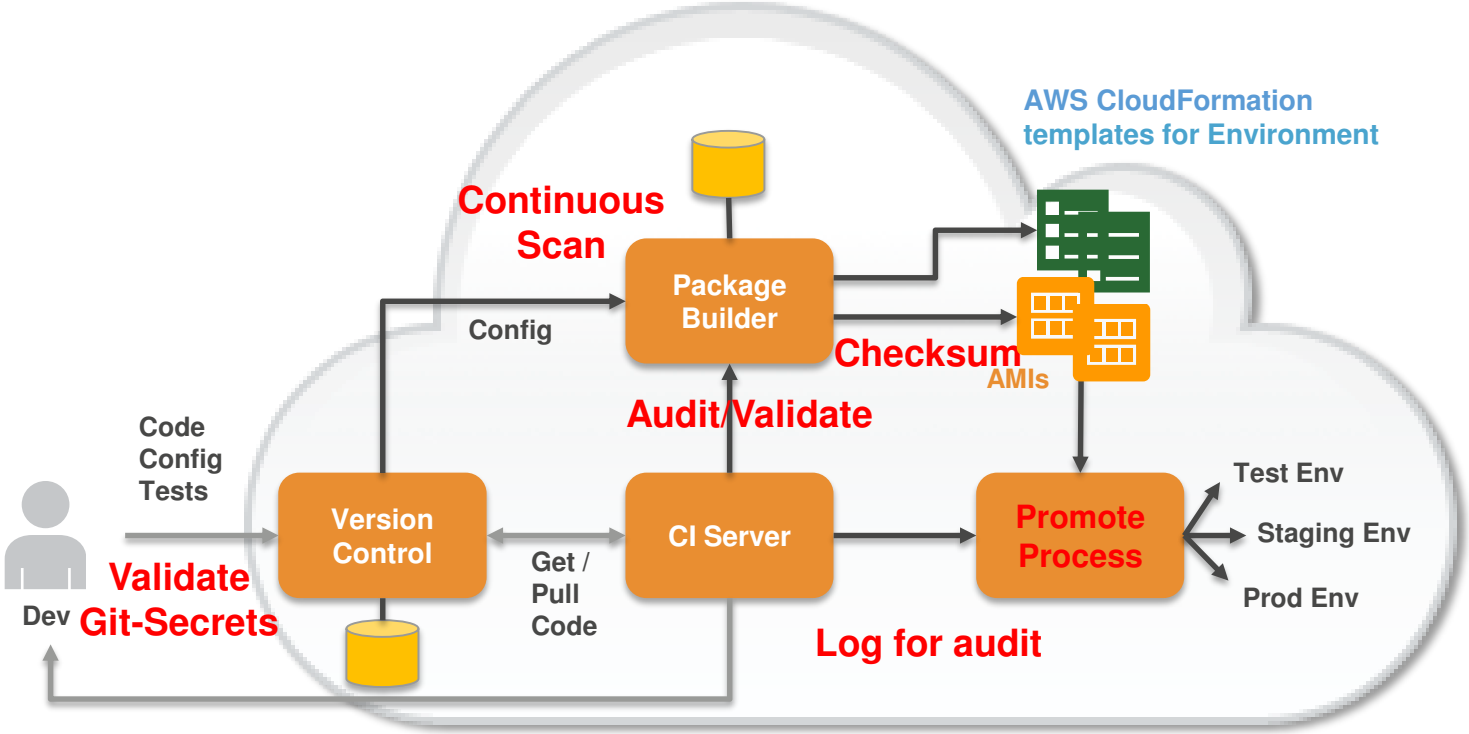
- **Not** there to audit code
- Implement the control segments to validate and audit code and artifacts as part of the CI/CD process

CI/CD for DevOps

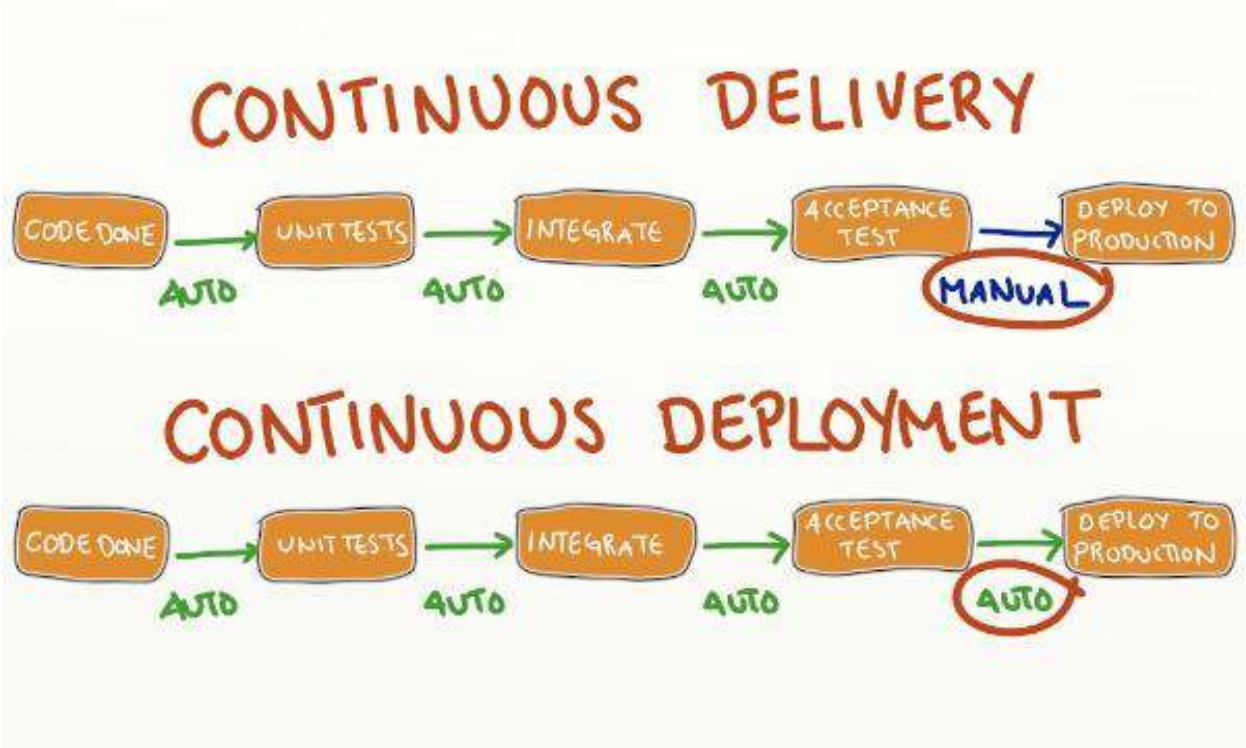


Send Build Report to Dev
Stop everything if build failed

CI/CD for DevSecOps



Promotion Process in Continuous Deployment



What Does DevSecOps CI/CD Give Us?

- Confidence that our code is validated against corporate security policies.
- Avoid infrastructure/application failure in a later deployment due to different security configuration
- Match DevOps pace of innovation
- Audit and alert
- Security at scale!

AWS CloudFormation primer

Infrastructure is code

AWS CloudFormation Primer

Allows you to define a “template”

- Composed of different “resources”
- Provision that template into repeatable, live, “stacks”.

CloudFormation (CFn) provides a single service interface

- Let CFn perform state changes and govern who calls CFn

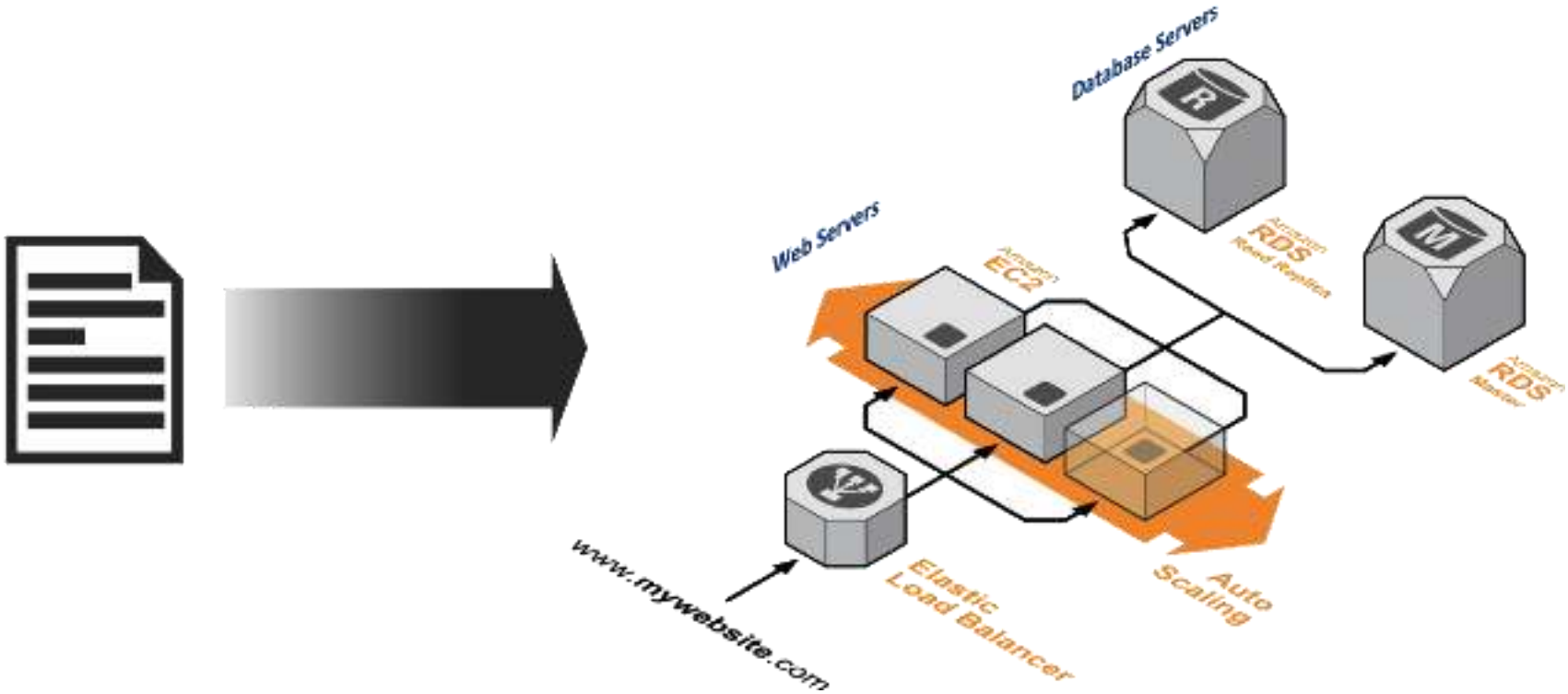
Treat as Code

- Check in your templates

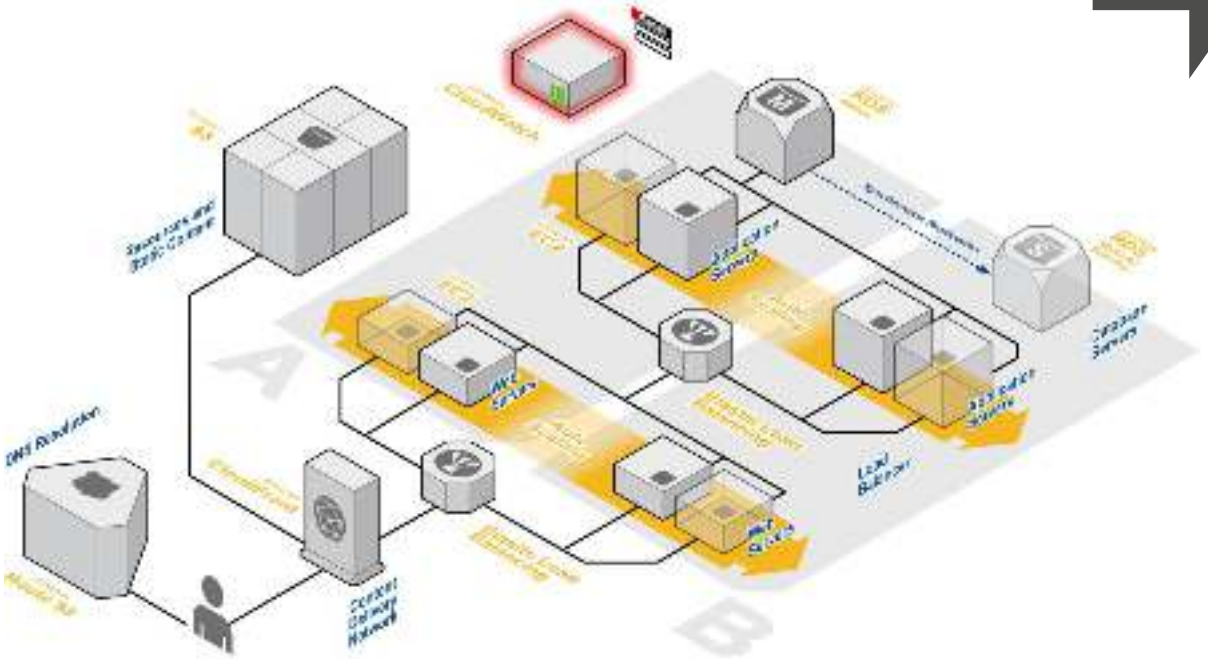
CFn templates can hook into external configuration management frameworks

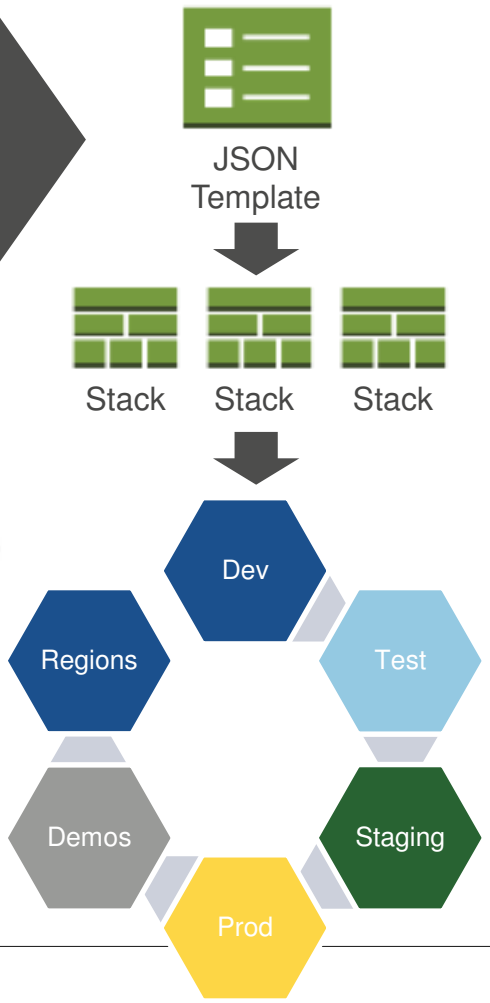
- Jenkins/Chef/Puppet/etc.

AWS CloudFormation Primer



AWS CloudFormation Stacks





Split Ownership Configurations

Who knows your solution best?

- Dev, Infra, Sec…?
- Delegate ownership
  - Infra – VPC design, IGW Deployment, Subnets, etc.
  - DevOps – EC2, Elastic BeanStalk, RDS, DynamoDB, etc.
  - OS Patching, Security Agent Deployments, IAM Roles, etc.

Use Yaml and split file into chunks or functions

- Separate file sources with access control – Use IAM/VPC-E/etc.
- Push files -> **Validate** -> Merge files -> **Validate** -> Deploy -> **Validate**

Jenkins for deployment

- Promotion flows
  - Move from manual to Automation based on validation quality
- Excellent for merging jobs of split configurations

Merging

From single file or multiple files

- Maintain access control using policies
- Use different source stores if needed

Based on function/state

Reusable patterns

Maintain order, especially of validation

- Security validation last to execute
- Security should always win

Validation

Keep track of what section you are validating

- Stage vs Prod
- Merged vs separated

Validate often and log/alert

- Validate part and end result
- Run-time validation

Use external agents/services

- Amazon Simple WorkFlow
- AWS Lambda
- AWS Config
- Amazon CloudWatch Logs
- Etc.

```
f = File.open("template.json", "rb")
g = f.read
g.gsub(",",",\n").each_line {|line|
    if (line.downcase.include? "cidrip") && !line.include?("0.0.0.0/0")
        cfnCorpCIDRArray.push line[/(([0-9]|[1-9][0-9]|1[0-9]{2}|2[0-4][0-9]|
    elsif (line.downcase.include? "cidrip") && line.include?("0.0.0.0/0")
        cfnCIDRArray.push line[/(([0-9]|[1-9][0-9]|1[0-9]{2}|2[0-4][0-9]|25[0
        p errorCIDRSource
        fail = true
    elsif (line.downcase.include?("fromport") && line.include?("0"))
        cfnPortArray.push line[/(([0-9]|[1-9][0-9]|1[0-9]{2}|2[0-4][0-9]|25[0
        p errorPortSource
        fail = true
    end
    }

# Check Corp CIDR blocks
strFail = ""
cfnCorpCIDRArray.each {|y|
    found = false
    corpCIDRArray.each {|x|
        cidr = NetAddr::CIDR.create(x)
        if ((cidr.contains?(y)) || (y == x))
            found = true
        end
        }
    if !found
        p errorCorpCIDRSource + " (#{y})"
        fail = true
    end
}
```

Where else can this be applied?

AWS CloudFormation

Amazon EC2 Container Service

AWS CodeDeploy

CFn Template

```
"name": "wordpress",
"links": [
  "mysql"
],
"image": "wordpress",
"essential": true,
"portMappings": [
  {
    "containerPort": 80,
    "hostPort": 80
```

Task Definition

```
version: 0.0
os: operating-system-
files:
  source-destination-
permissions:
  permissions-specifi
hooks:
  deployment-lifecycl
```

AppSpec File

…and more.

Other tools to keep close

Tracking

- AWS Config Rules *(Preview)*
- Amazon CloudWatch Events
- AWS CloudTrail
- AWS Inspector

Execution

- AWS Lambda

Track/Log

- Amazon CloudWatch Logs
- Amazon DynamoDB

Alert

- Amazon SNS

Why should I do this?

Implement “Compliance Status” for easy overview

- Use predefined checks
- Create extended custom checks
- Fix the iss

Always Log and Alert!

Evaluate/remediate changes/events in your account

- Doesn’t replace log analysis (Machine Learning FTW)
- Protect against changes made by (un)authorized accounts
- Automatic remediation for critical events
- Do forensics on the fly

Thank you!