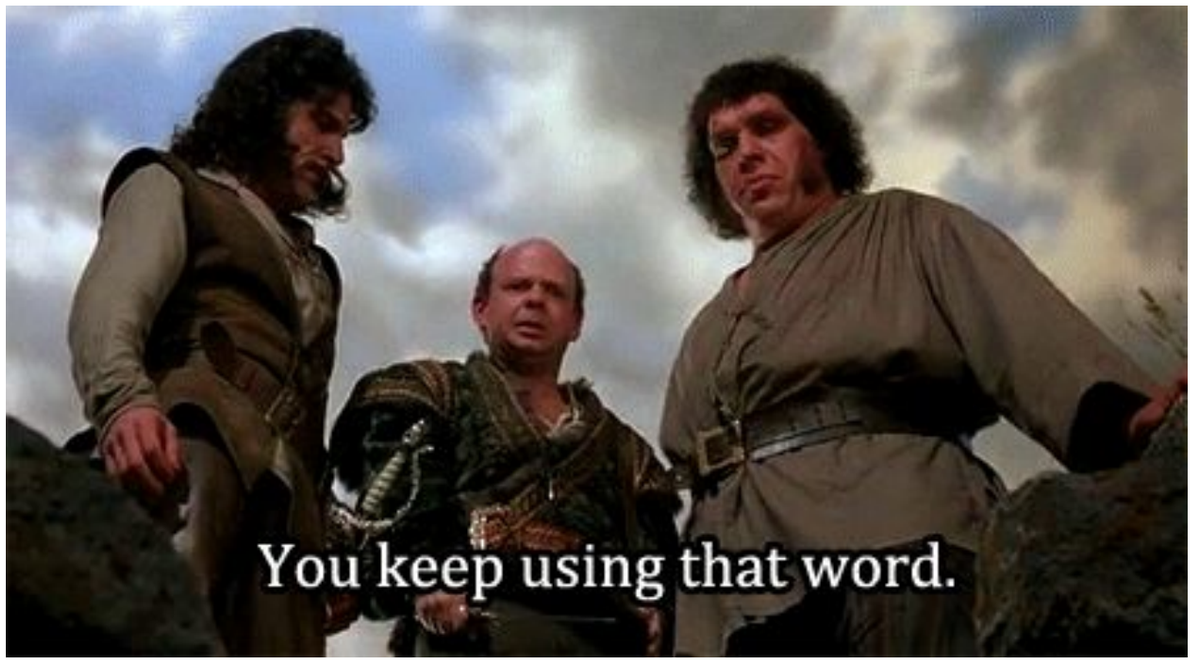
DevOps @ Domain

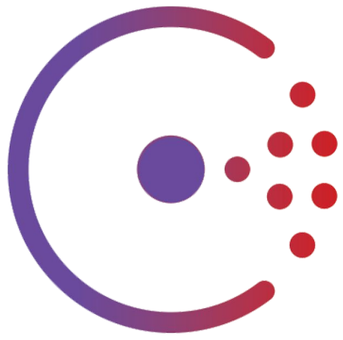
Jordan Simonovski
Nikita Gorenbukh

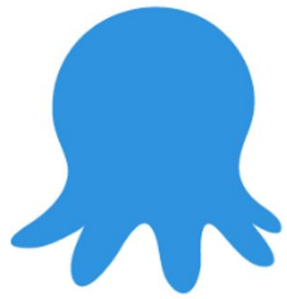
What *is* DevOps, really?

- DevOps is the practice of operations and development engineers participating together in the entire service lifecycle, from design through the development process to production support.

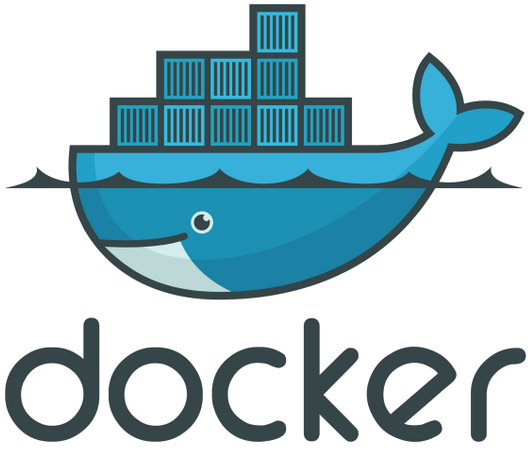


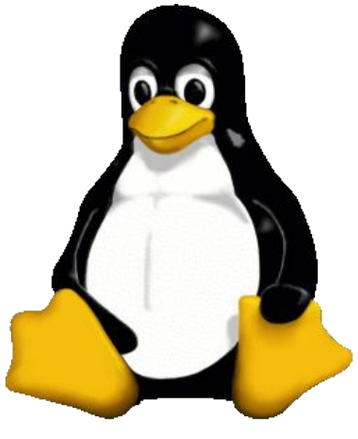
What Kind of apps do we Write?





Our Infra





So, how Does all of this fit together?

```
1   FROM node:8.1.4-slim
2
3   EXPOSE 3000
4
5   ENV DEBIAN_FRONTEND=noninteractive
6   ENV APP_DIR /var/www
7   ENV TZ Australia/Sydney
8
9   ENTRYPOINT ["/usr/local/bin/dumb-init", "--"]
10
11  RUN mkdir -p ${APP_DIR}
12  WORKDIR ${APP_DIR}
13
14  # Add needed files (.npmrc needed for npm login)
15  COPY yarn.lock .npmrc package.json ./
16  RUN yarn install --quiet
17
18  # Add an empty .env file (if it doesn't exist)
19  RUN touch .env
20
21  # Add app files
22  COPY . ./
23
24  # Build it
25  RUN npm run build
26
27  # Remove files in the cache directory
28  RUN rm -f ci/cache/*
29
30  CMD ["yarn", "run", "start:prod"]
31
```

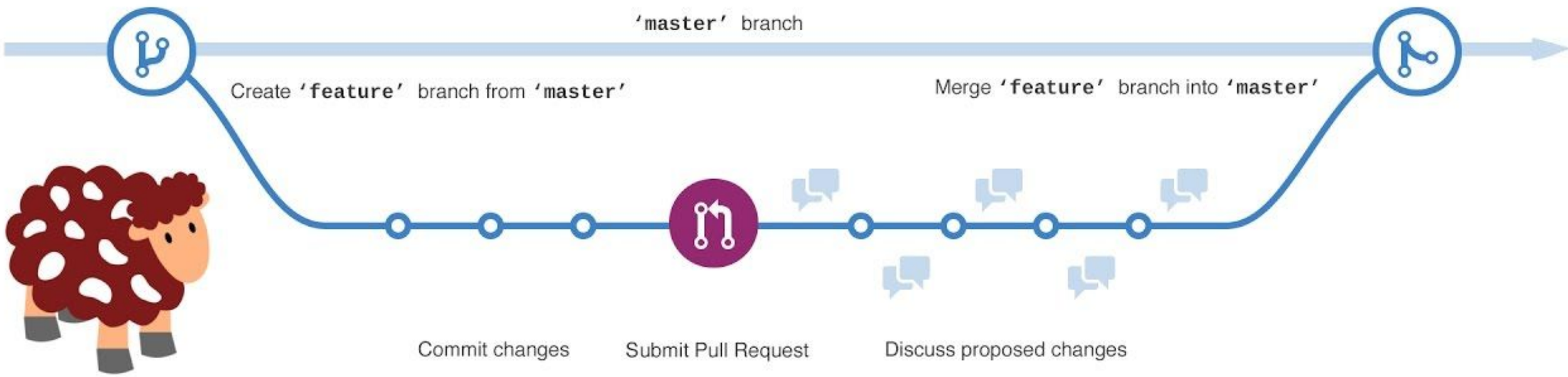
- Introducing the Dockerfile
- Create an image once and run anywhere.
- Lets us change the underlying infra.

So, how do things get deployed?

It all starts with our devs:

- Create feature branch
- Write Code
- Write Tests
- Raise Pull Request

GitHub Workflow



What happens after code gets merged?

✓ Domain Group / fe-server-search-results #94

Pipeline | Changes | Tests | Artifacts

Branch: master — 25m 7s — No changes

Commit: bb0971c — an hour ago

Checkout ✓ — BuildDecision ✓ — InitialNotify ✓ — Build ✓ — setEnv ✓ — Check and Create Cloudwatch Sub... ✓ — yarnBuild ✓ — **Test** ✓ — E2E Test ✓ — Deploy ✓ — Notify Slack Promotion ✓ — Notify Slack Rollback ✓

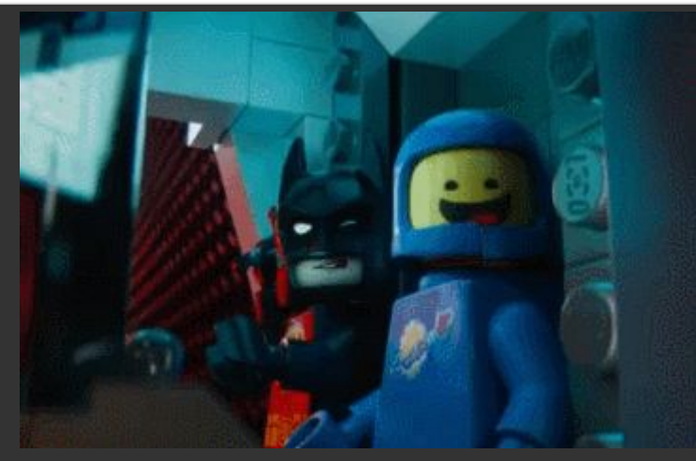
Steps - Test

✓ › Print Message

✓ ⌄ Shell Script

Show complete log

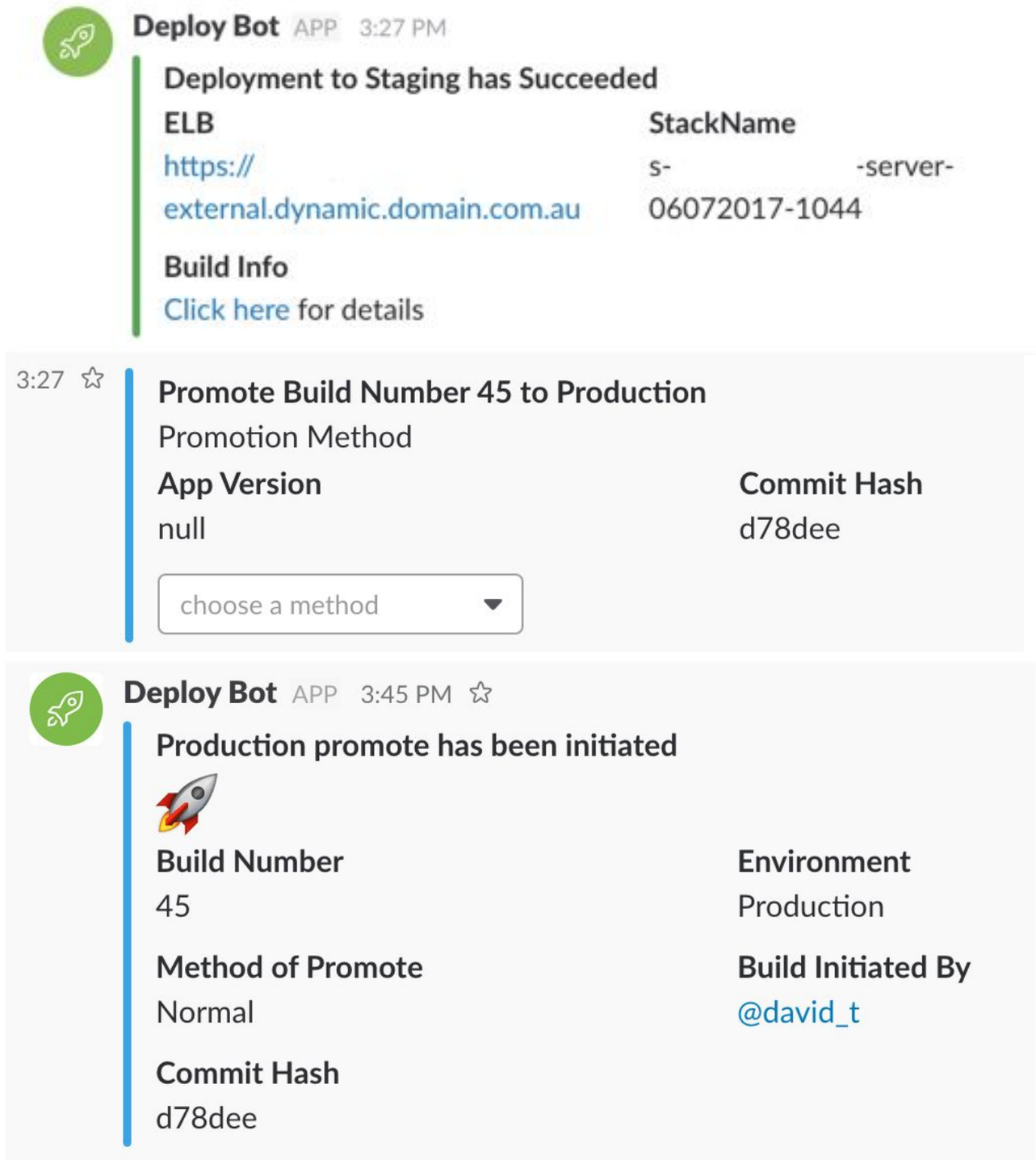
```
rvice.js
src/services/search-results/data-cards.js -> lib/services/search-results/data-cards.js
src/services/search-results/featured-agencies.js -> lib/services/search-results/featured-agencies.js
src/services/search-results/featured-properties.js -> lib/services/search-results/featured-properties.js
src/services/search-results/search-results.js -> lib/services/search-results/search-results.js
src/services/search-results/shortlisted-listings.js -> lib/services/search-results/shortlisted-listings.js
src/services/slugmap.js -> lib/services/slugmap.js
src/services/user/auth.js -> lib/services/user/auth.js
src/services/user/saved-search-legacy.js -> lib/services/user/saved-search-legacy.js
src/services/user/saved-search.js -> lib/services/user/saved-search.js
src/services/user/shortlist-count.js -> lib/services/user/shortlist-count.js
src/services/user/shortlist-set.js -> lib/services/user/shortlist-set.js
src/services/user/user-cookie-and-data.js -> lib/services/user/user-cookie-and-data.js
src/services/user/user-details.js -> lib/services/user/user-details.js
src/services/user/user-nav-shortlisted-listings.js -> lib/services/user/user-nav-shortlisted-listings.js
src/services/user/user-stats.js -> lib/services/user/user-stats.js
[2017-08-02 20:37:08] File /var/www/node_modules/weinre/web/client/nls/English.lproj/localizedStrings.js is malformed
```



How do we make our deployments easier?

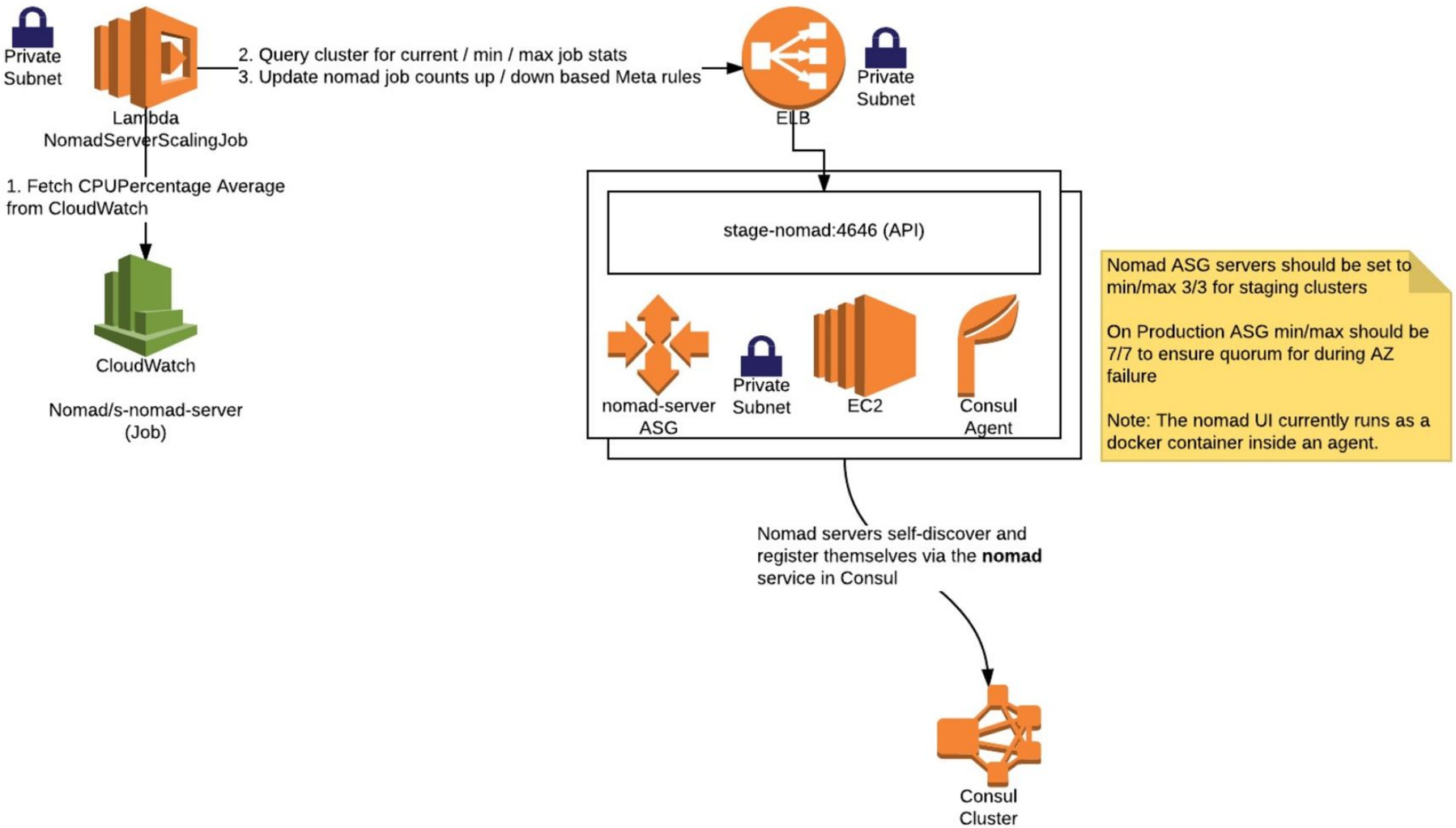
Introducing Deploy Bot

- Serverless Slack Bot
- Written in Python/Flask
- Using the Zappa Framework
- Doesn't cost anything when it's not being used
- Deploy from anywhere!



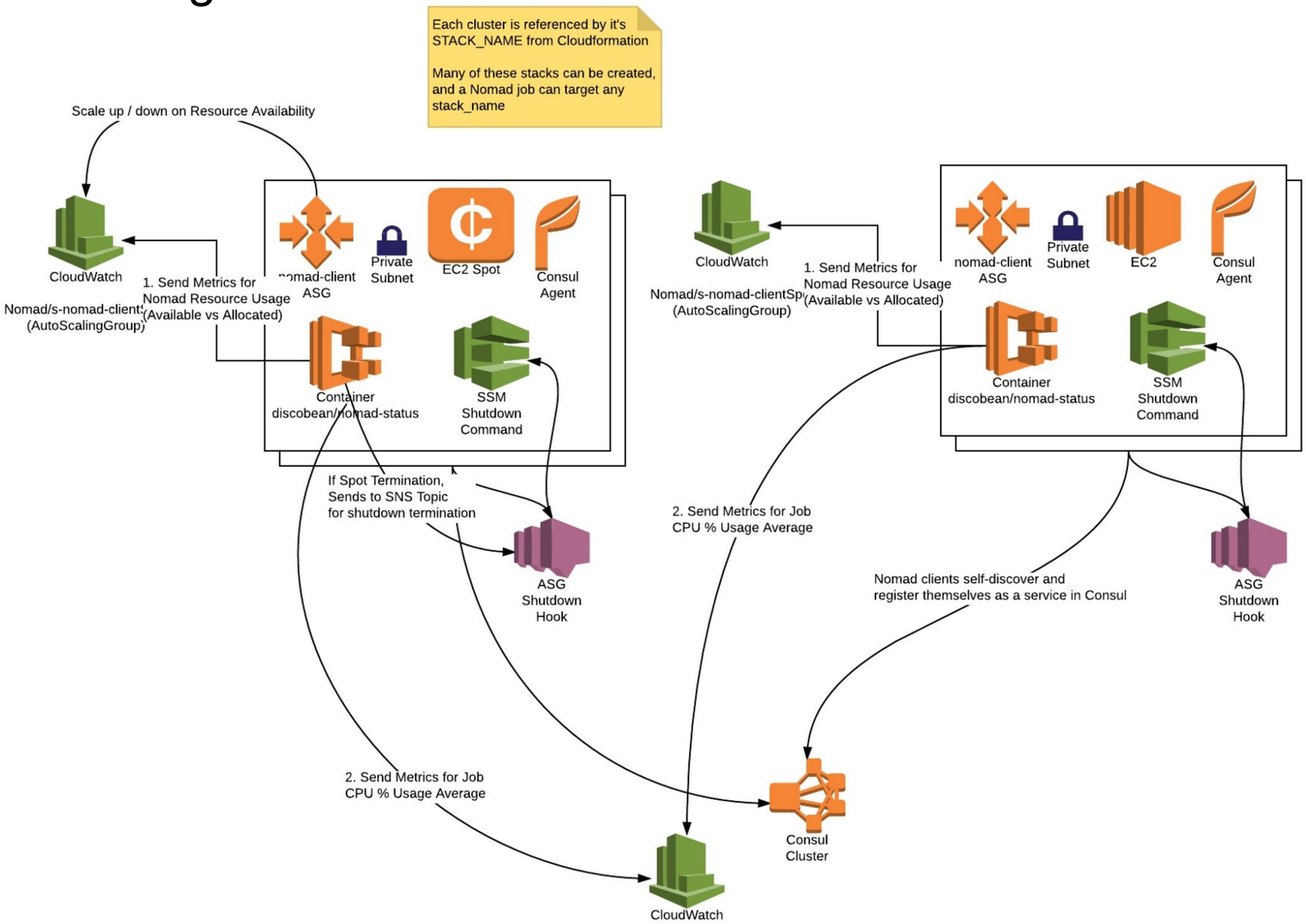
Linux Infrastructure

Nomad Server



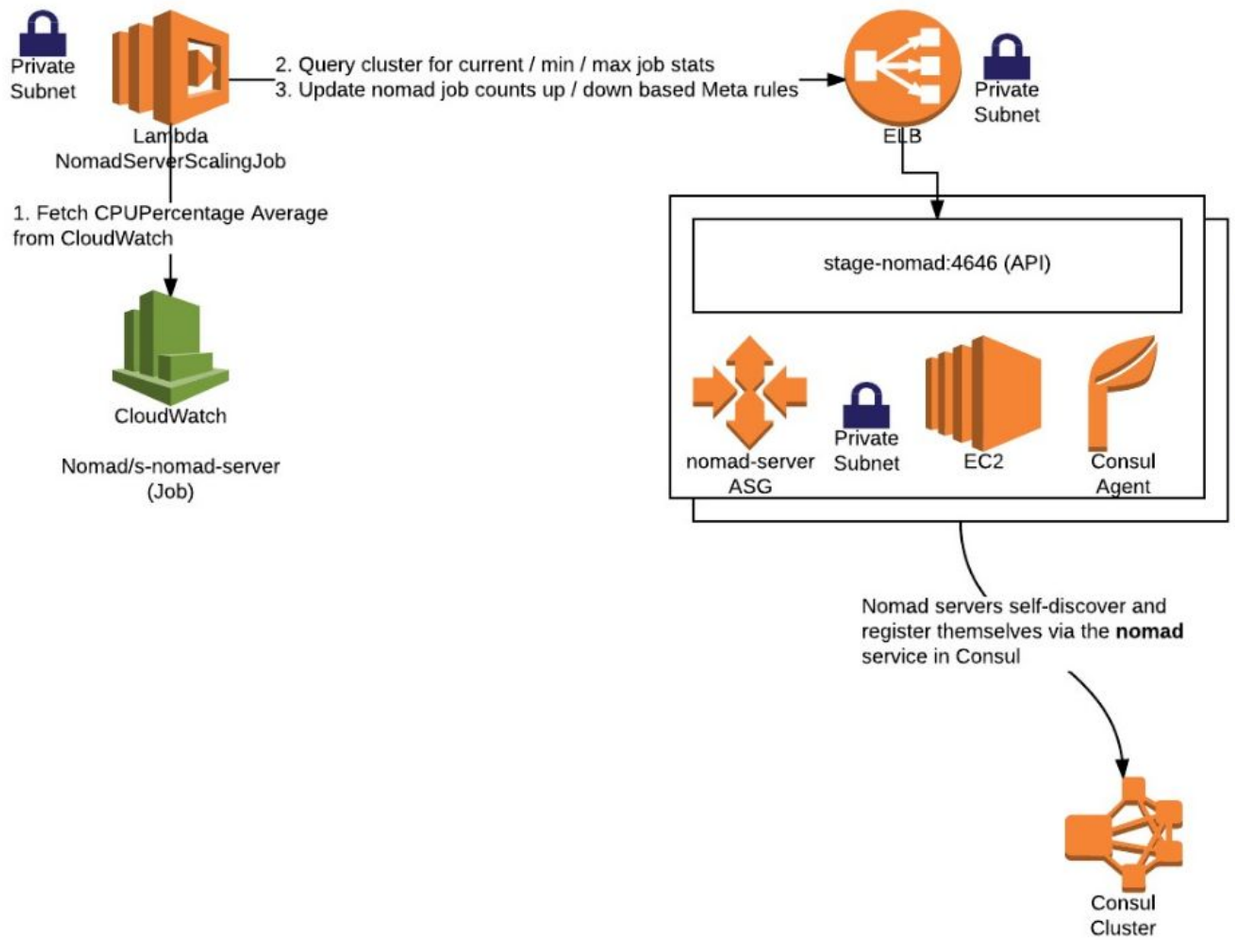
Linux Infra (Cont.)

Nomad Agents



Scaling Containers

1. Service is started with metadata metrics per task group
   a. count_max = 10
   b. count_min = 2
   c. scaledown_decrease_count = 1
   d. scaledown_threshold = 20
   e. scaleup_increase_count = 2
   f. scaleup_threshold = 40
   g. threshold_metric = AverageCpuPercent
   h. ...
2. Service runs on each Nomad Agent
   a. Sends metrics to Cloudwatch
3. Lambda executes
   a. Checks metrics against ruleset
   b. Scales up/down



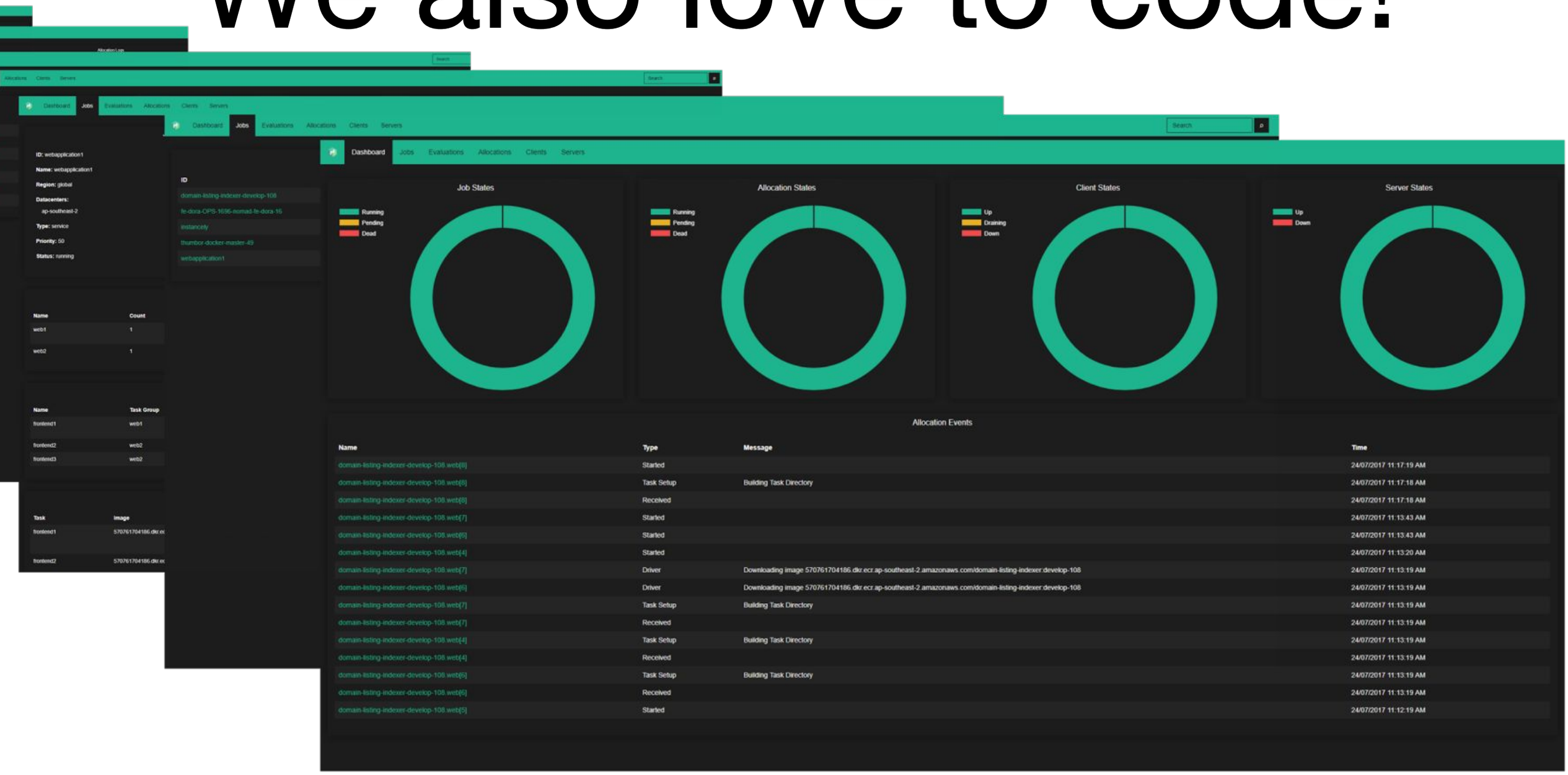
How do we simplify all of this complexity?

```
node('docker') {
    checkout scm

    if(!do_deploy)
        return false

    stage("ECR") {
        try {
            sh returnStatus: true, script: "aws --region ${region} ecr create-repository --repository-name ${app_name}"
        }
        catch (Exception e) {
            echo "Repository already exists. Skipping create."
        }
    }

    stage("Cloudformation") {
        stack_name = "${environment}-${app_name}"
        cf_template = "${environment}-cloudformation.yaml"
        outputs = cfnUpdate stack: stack_name, file: cf_template
        custom_params << ["IAM_ROLE": outputs.RoleArn ]
    }

    stage('Docker') {
        docker.withRegistry(env.AWS_ECR) {
                npmLogin credentialId: "n/a"
            def container = docker.build("${app_name}")
                container.push(version)
        }
    }

    stage("Deploy") {
        envFile = loadenv.getEnvFile(environment: environment)
        docker_env = loadenv filename: envFile, decrypt: true, extraParams: custom_params
        appDeploy(environment: environment, appName: app_name, dockerEnv: docker_env, version: version)
    }
}
```

What we get

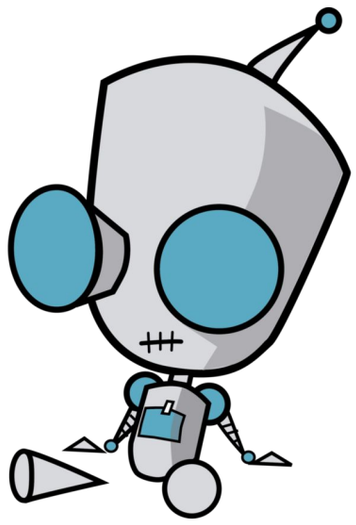
- Windows + Linux containers
- Fast container scale-up
- Fast scale-out
- Cheap instances via Spot, managing spot termination
- Historical metrics job CPU vs deploy versions
- IAM Policies
- Fast deploy/rollback times
- 1hr to build new stacks

We also love to code!



github.com/cvandal/nomad-ui

Windows Infrastructure

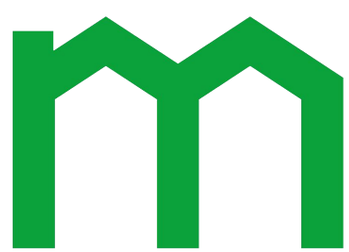


Robot Army

Octopus Deploy

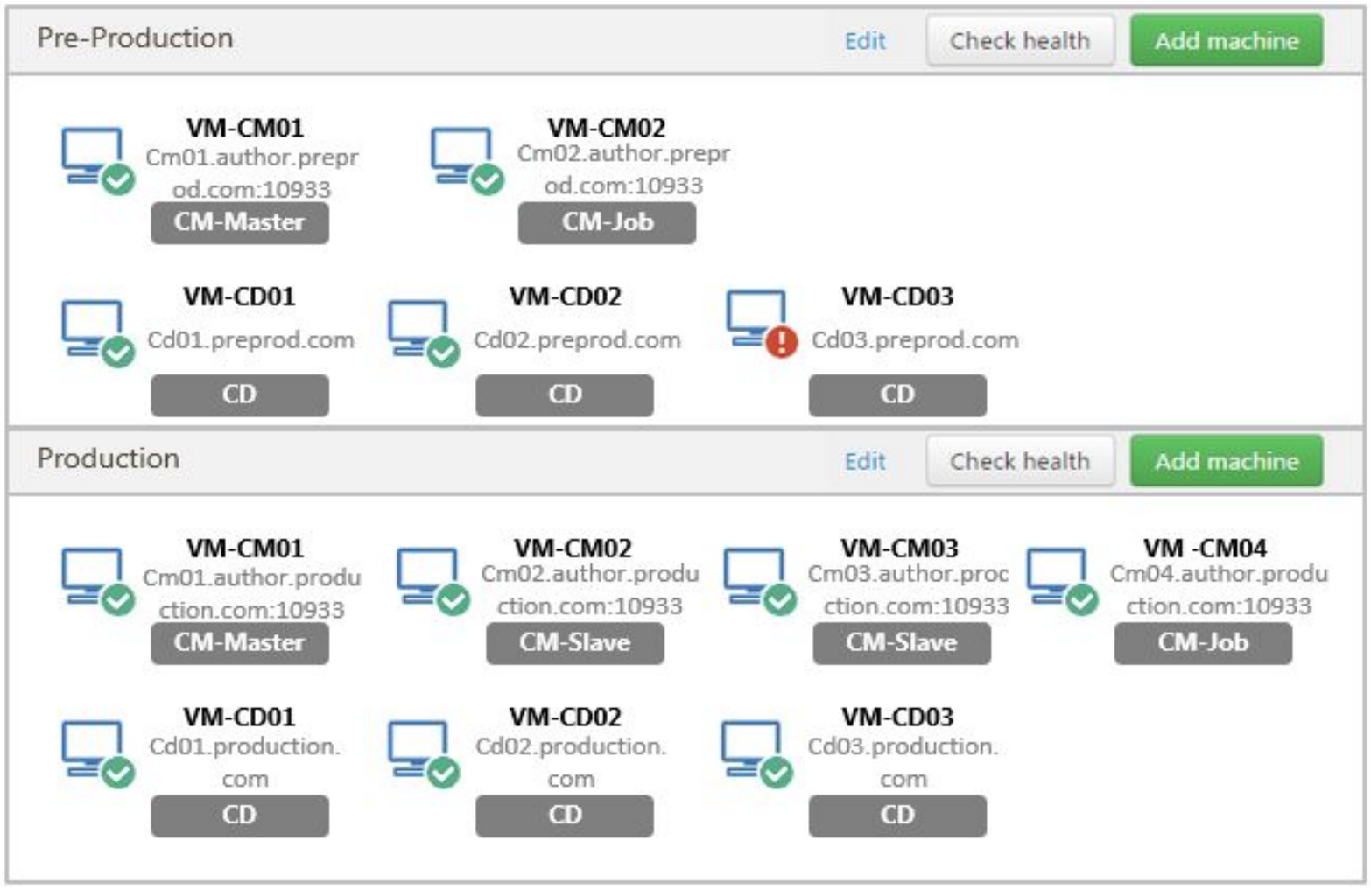
Octopus Deploy

- **Environments**
- **Projects**
- **Project Deployment**
- **Processes**
- **Step templates**



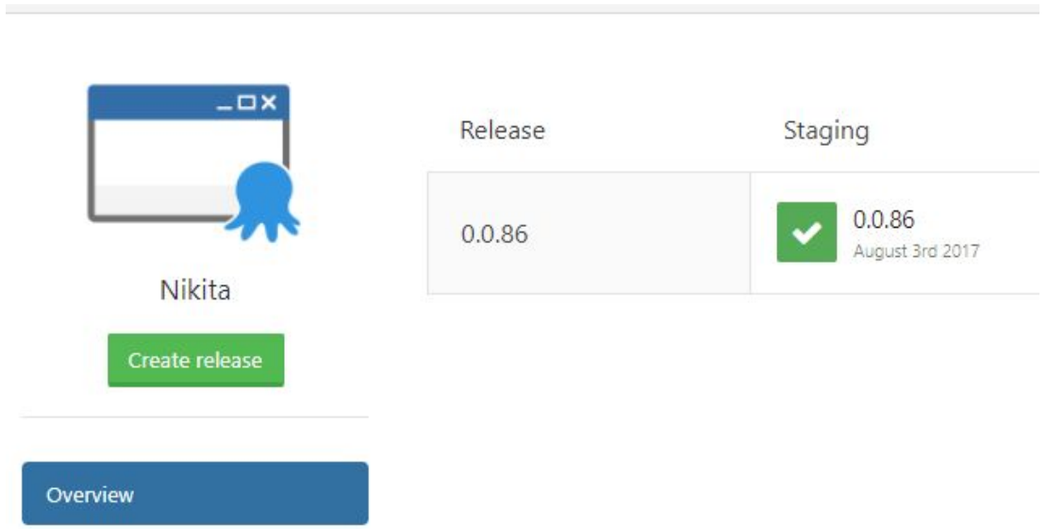
Octopus Deploy

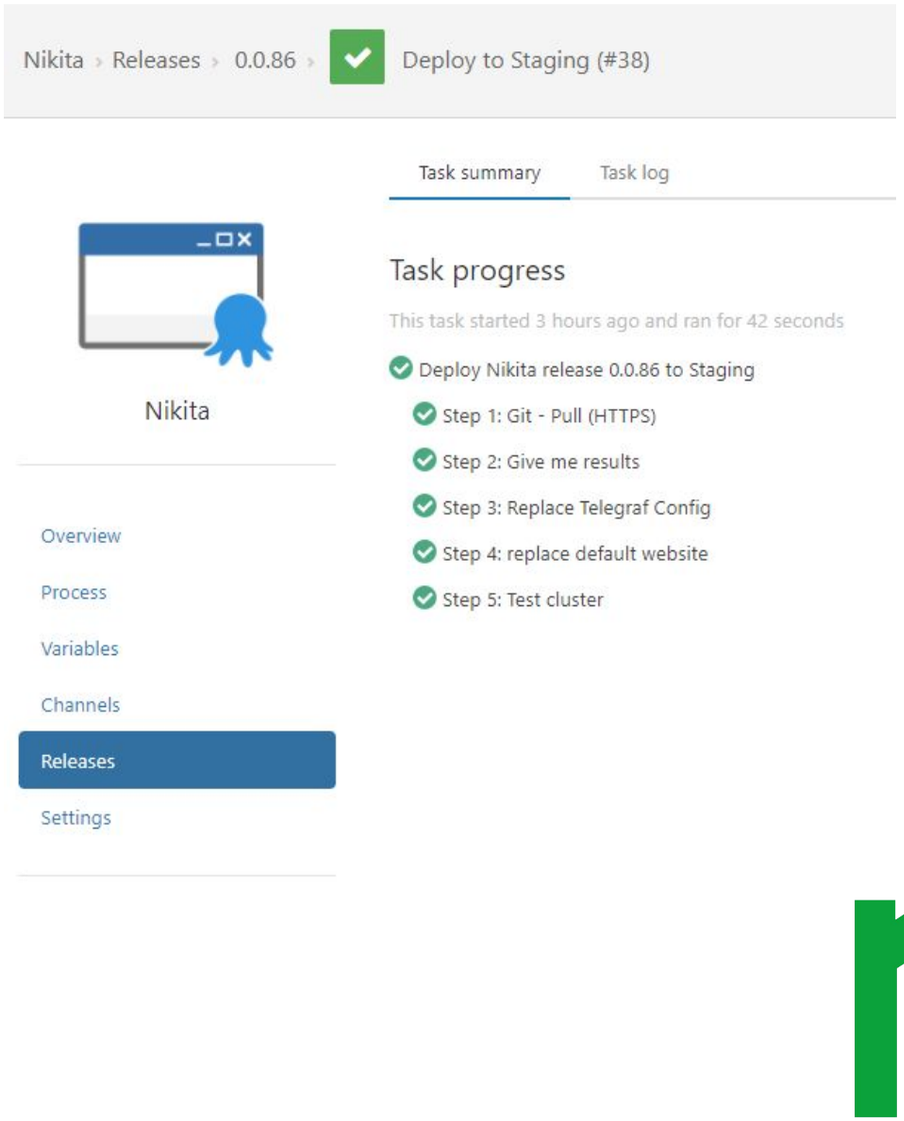
Environments



Octopus Deploy

Projects





Octopus Deploy

Project Deployment

Task summary | Task log

This task started 3 hours ago and ran for 42 seconds

Deploy Nikita release 0.0.86 to Staging

```
The deployment completed successfully.
```

Step 1: Git - Pull (HTTPS)

Step 2: Give me results

Step 3: Replace Telegraf Config

Step 4: replace default website

WIN-2P28NA2JQ6M

```
Executing script on 'WIN-2P28NA2JQ6M'
Name             Application pool   Protocols    Physical Path
----             ----------------   ---------    -------------
GiveMeWeather    .NET v4.5          http         C:\Octopus\Applications\GitPul
                                                 l\Staging\Nikita\Git - Pull
                                                 (HTTPS)\published website
```

Octopus Deploy

Processes

Deployment process

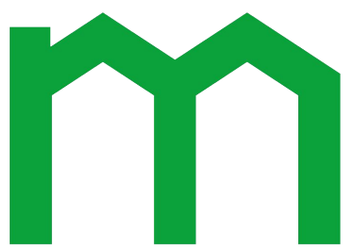
1. Git - Pull (HTTPS)
Run a PowerShell script across machines in roles: nikita nikita-ra4

2. Give me results
Run a PowerShell script across machines in roles: nikita

3. Replace Telegraf Config
Run a PowerShell script across machines in roles: nikita nikita-ra4-test

4. replace default website
Run a PowerShell script across machines in roles: nikita nikita-ra4

Add step Reorder steps



Octopus Deploy

Step Templates

Library › Step Templates › Git - Pull (HTTPS)

Step | Parameters | Settings | Usage 1

- Packages
- External feeds
- Lifecycles
- Script modules
- Step templates
- Variable sets

Configure the step using the fields below. Use the **Parameters** tab to define variables that can be used in scripts and bound fields.

Script

PowerShell C# Bash

```powershell
if ($env:path -notlike "*git*") {
    Write-Output "Adding the Git 'bin' directory to the 'Path' environment variable..."
    $path = $env:Path += ";C:\Program Files\Git\bin"
    [Environment]::SetEnvironmentVariable("Path", $path, "Machine")
}

[System.Reflection.Assembly]::LoadWithPartialName("System.Web")
function Format-UriWithCredentials($url, $username, $password) {
    $uri = New-Object "System.Uri" $url

    $url = $uri.Scheme + "://"
    if (-not [string]::IsNullOrEmpty($username)) {
        $url = $url + [System.Web.HttpUtility]::UrlEncode($username)

```

Robot Army

Cluster Definition

```
RobotPlatoon stagingNikita
{
    Ensure = "Present"
    Name = "s-nikita"
    Team = "DevOps"
    OctopusProject = "Nikita"
    PlatoonVersion = "ra3"
    Sleepytime = "Yes"
    InstanceType = "t2.nano"
    OctopusVersion ="3.0"
    PublicInstances = $false
    VPC = "Staging"
    HealthCheckEndpoint = "TCP:80"
    minSize = 1
    maxSize = 1
}
```

So, DevOps at Domain

- Incredibly fun
- Nothing like traditional operations
- Lots of coding involved
- Requires a holistic view of what we run and how we're running it
- Always lots to learn
- Always on the bleeding edge

/DomainGroupOSS

/Domaincomau

tech.domain.com.au
developer.domain.com.au

Thank you.
Domain

Jordan Simonovski
Nikita Gorenbukh