ConcourseCI

Background

- CI solution based completely around pipelines
- Pipelines are declared using YAML
- Interacting with CI servers is done with the `fly` command line utility

AD-ConcourseCI

- Uses Terraform
- Complete stack presented in a similar style to segment stack
- Use the stack module and you'll have a CI environment in minutes!

AD-ConcourseCI

main.tf — /Users/georges.haidar/Work/myob/EX-CD-Common

main.tf

```
variable "github_app_id" { type = "string" }
variable "github_app_secret" { type = "string" }

provider "aws" {
  region = "ap-southeast-2"
}

module "stack" {
  source = "git::ssh://git@github.com/MYOB-Technology/AD-ConcourseCI.git//terraform/modules/stack"
  region = "ap-southeast-2"
  team = "ex"
  name = "common"
  owner = "Georges Haidar georges.haidar@myob.com"
  environment = "Development"
  ci_version = "0.1.18"
  ecr = "587143430827.dkr.ecr.us-east-1.amazonaws.com"
  hosted_zone_id = "Z2YQW7IT433ZD6"
  hosted_zone_name = "addevcloudservices.com.au"
  ssl_certificate_arn = "arn:aws:iam::587143430827:server-certificate/addevcloudservices_wildcard_apse2"
  ssh_key_name = "AD-ConcourseCI"

  github_app_id = "${var.github_app_id}"
  github_app_secret = "${var.github_app_secret}"
}

```

File 0 Project 0 ✔ No Issues main.tf 1:1 LF UTF-8 Plain Text master 1 update

Claims (or the Why?)

- Very simple to deploy
- Predictable CI: everything is driven by changes to resources (e.g. git, docker)
- No GUI: Entire pipeline is in YAML and sits next to your service (myservice/ci/pipeline.yml)
- Every thing runs in Docker containers so no need to do maintenance/clean up on workers
  - Also every run of a build step is independent of anything else
- Very cheap to run after amortising initial development costs

Cons

- New CI system to learn. Learning curve can be intimidating for new users*
  - *However, at MYOB, we have each other :)
- Usual arguments of hosted vs self-managed CI
  - You have to maintain the Admin instances and a database

Architecture

- ATC: also known as web or admin server
- TSA: custom-built SSH server that is used solely for securely registering workers with the ATC.
- Worker
- PostgreSQL

Concepts

- Pipelines
  - Resources
  - Jobs
    - Tasks

Concepts - Pipeline

- A directed graph of resources and jobs
- You’ll see it in the demo

Concepts - Resources

- A resource is any entity that can be checked for new versions, pulled down at a specific version, and/or pushed up to idempotently create new versions.
- Examples:
  - Git
    - check: git log
    - get: git clone && git checkout
    - put: git push
  - Slack
    - put: curl -XPOST <slack_webhook_url>

Concepts - Jobs

- Describes a set of tasks as a plan
- Triggered when dependent resources change (or manually)
- Example:
- when a git resource change:
  - do:
    - get: git resource
    - task: npm install
    - In parallel:
      - task: npm test
      - task npm run lint
  - on failure:
    - put: slack

Concepts - Task

- A task describes a single step of work in a job (a script)
- Every task is run in its own docker container so there is no polluting the worker instances

Try It Out

- https://ci.concourse.ci/
- https://ex-cd-common.addevcloudservices.com.au/
- https://zoidberg-cd-taxonline.addevcloudservices.com.au/