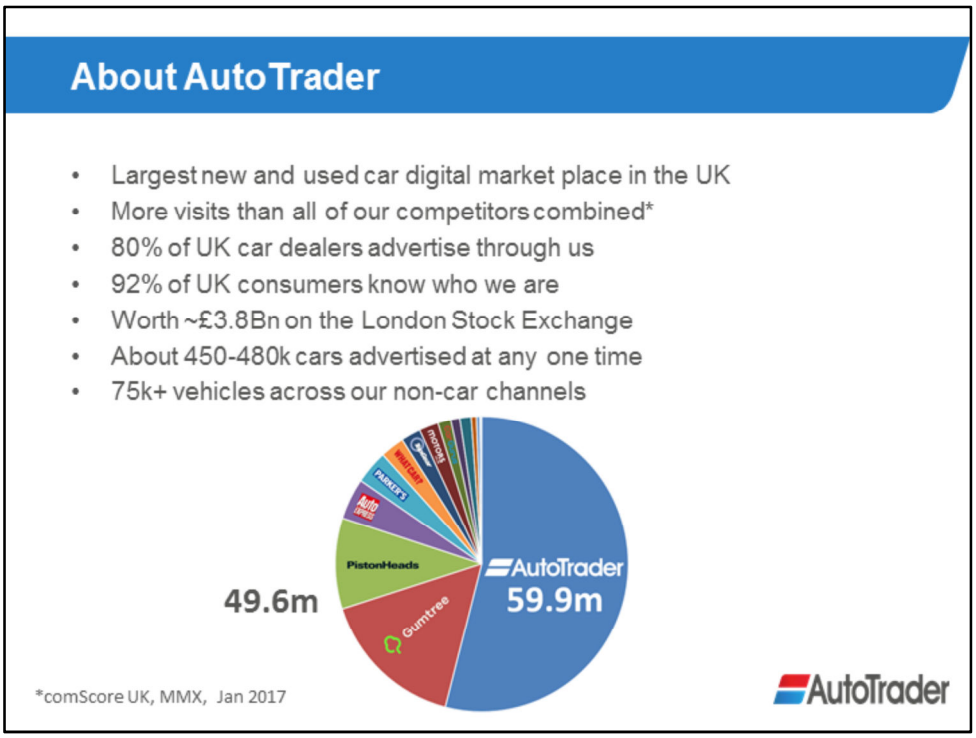

Today I am going to talk about who I am and who Autotrader and what Autotrader has achieved in the last 4 years with agile & continuous delivery
Crash course in what continuous intergration & continuous delivery is
Finally what tools are available to help you follow a similar approach as developers

## About Me

- Operations Engineer for AutoTrader
  - Been here for 3.5 years
- Previously worked for M247
  - Samer taught me the basics
- Odd fact is my arm span is greater than my height
- I don't have as much skill on a mountain bike as I thought

AutoTrader

I am an operations Engineer at AutoTrader and been here for 3.5 years.
Previously I worked at M247 in Trafford Park as a 2nd / 3rd line support and also doing basic network stuff
Odd fact is my arm span is wider than my height (this is very uncommon)
Shortly after my talk in October went to Scotland for a long weekend mountain biking with some friends from work and on day two went over the bars and broke my scayfoid




•Largest new and used car digital market place in the UK
•We were 6x larger than our nearest competitor
•80& of UK car dealers advertise through us
•92% of UK consumers know who we are
•Worth ~£3.9Bn on the London Stock Exchange
•About 450k+ cars advertised at any one time

33200 vans
17200 bikes
5200 motorhomes
3200 caravans
7200 truck
7300 farm
2200 plant


Gratuitous office photos

## Our Product Estate

2013/14

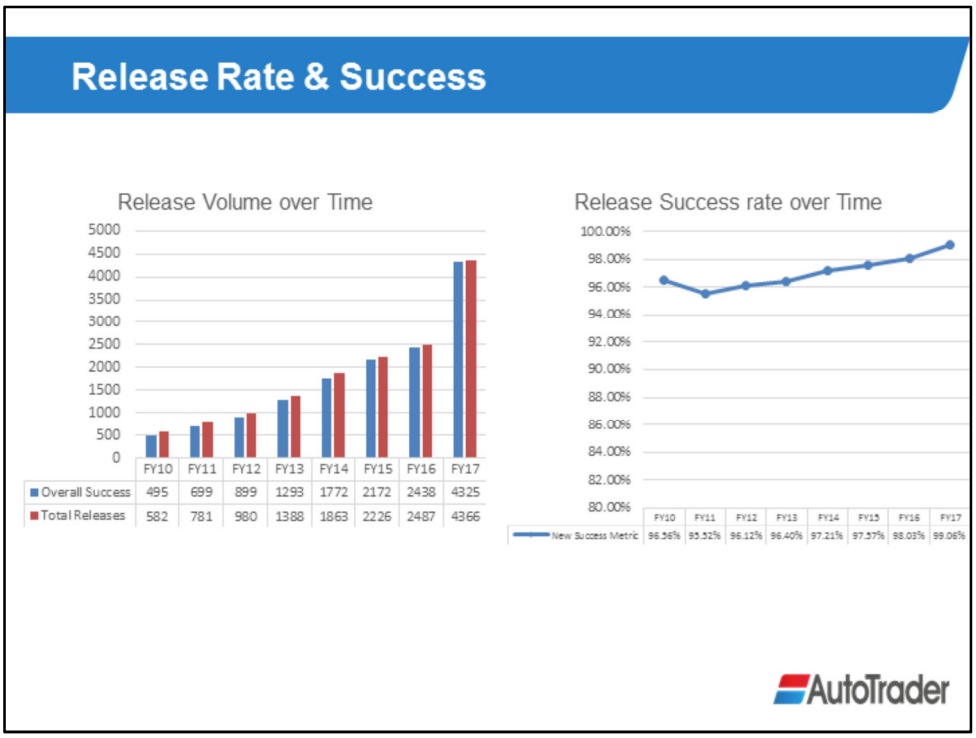
- 40 Deployments per week
- Deployed by Operations to live
- Deployed to live manually using legacy Perl scripts
- 75 Apps
- 350 Go Pipelines

- 2016/17
- 80+ Deployments per week
- Deployed by Devs to live
- Deployed to live automagically using GO CI pipelines
- 195 Apps
- 2400+ Go Pipelines

AutoTrader

Late 2012 / Start of 2013 restructured into tribes and squads to start our agile journey
40 deployments a week, deployed using some legacy Perl scripts by 2 ops engineers each day
We had about 75 applications in live, many of them old monoliths
350 Go pipelines mostly handling dev & qa and pushing an rpm into the live repo

So for FY17 we were doing 80+ often hitting 100 a week with 99% of them being triggered by the devs into live all using the same pipelines that were used in dev & qa
195 apps in live many of the old monoliths either gone or been split into microservices.




In FY17 we started counting releases related to our new cloud platform and so this caused an increase but we also handed over “full” control of releasing to live to the development squads. With this rapid changes in number of deployments we have been able to be more successful with our releases as fewer stories were being included each time which meant the risks were lower. The metric for an unsuccessful release is if live customer traffic has been sent to it, so it may be a new defect was introduced or it completely didn’t work but if picked up in the PIT testing and backed out it wasn’t counted as failed.
FY17 we finished the year on 99.06%


We all know what it is, lets not talk about it!


So what is Continuous Integration (CI) and Continuous Delivery (CD)
For one it’s not a dog in the back of a Discovery

## Continuous Integration (CI)

- Code is stored in shared repository (Git, SVN)
- Committed multiple times per day (typically)
- Each check in is then verified with automated builds
- Fast feed back loop
- Build & deployment process to be consistent
- Dev/QA environments to be consistent
    - OS
    - Package versions
    - Hardware can be different for each environment requirements

AutoTrader

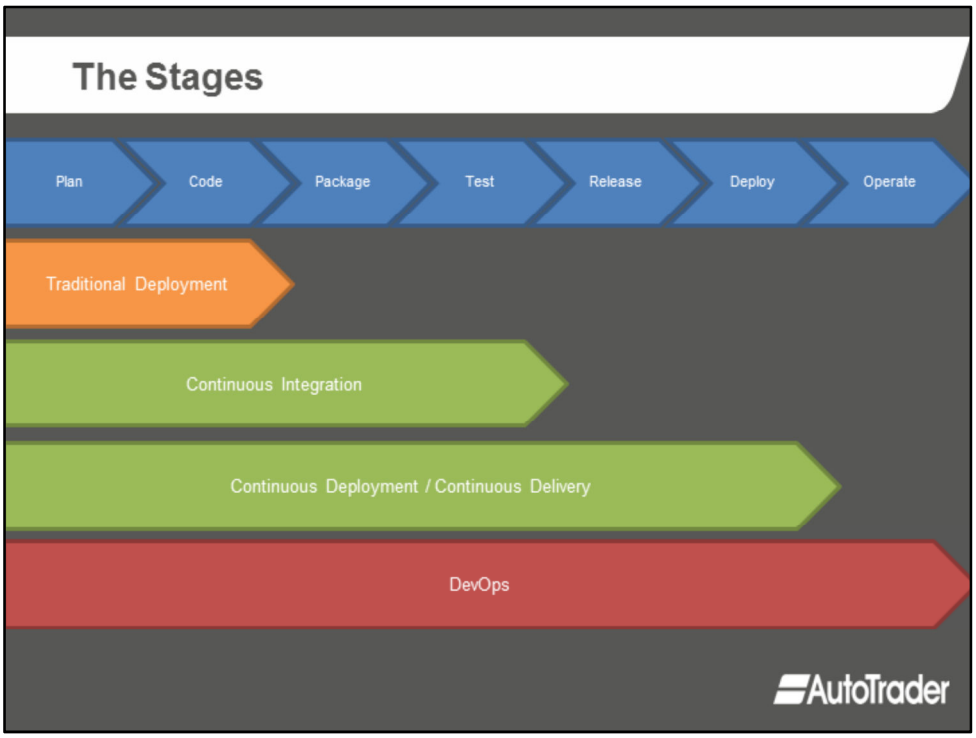
Code gets stored in something like GIT or SVN
You will commit multiple times a day to it
Each check-in will then get checked with an automated build process like GO, Jenkins
This then gives you a fast feedback loop as to if you have a working build or not
Your build and deployment process should be consistent in your non production environments
Dev and QA environments should be consistent as to OS choice and version and the software packages deployed.
However your “hardware” can be different (to within reason) so that it matches the requirement of the environment.

## Continuous Delivery (CD)

- An extension of CI
- Deployed to live automatically
- Post integration testing done automatically
- Build & deployment process to be consistent across all environments
- Dev/QA/Staging/Live environments to all be consistent
  - OS
  - Package versions
  - Hardware can be different for each environment requirements

AutoTrader

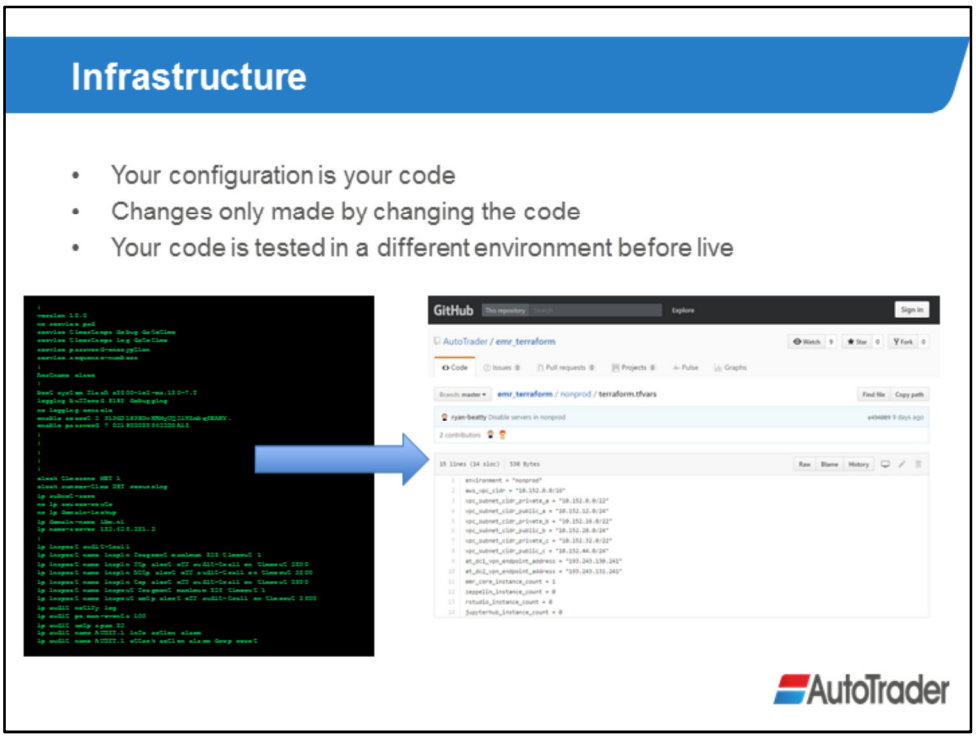
This is continuing what you already do with CI
Your deployments to live a done automatically
The running of your PIT testing happens automatically
The build and deployment process now has to be consistent all the way into live
Each environment has to have been configured the same way, this doesn't mean that the specs are the same, we use a 1 2 3 model typically. With Dev & QA servers being of a lower spec than live.




We have our deployment stages
Traditional deployment just gets you your planning and code
CI will get it packaged and tested
CD will give you the automated release and deployment
DevOps combines everything earlier and you are responsible for ruining it, monitoring it, cleaning up after it.

Remember though that:
DevOps is not simply combining Development & Operations teams
DevOps is not a separate team
DevOps is not a tool
DevOps is not a one-size-fits-all strategy
DevOps is not automation


What about infrastructure?
Quick of hands at to who actually commits there configs to SVN or GIT?




So in reality nothing really changes here compared with developers
Your config is your code rather than an application
You should only be making your changes by committing a change to your source control
You  then test this in a different environment

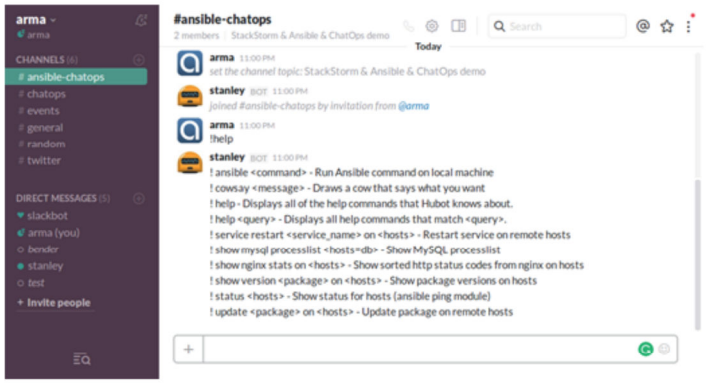

Cloud Only?
AutoTrader

NO!

AutoTrader

## Nope

- Networks is fashionably late to the party
- Use the same principles & tools as sysadmin / developers
- Brings benefits of things like chatops




AutoTrader

It's just that we are fashionably late to the party,
You just need to adopt the same tools and priciples that your dev and sys admin colleagues have being using
You also start getting other benefits like chat ops

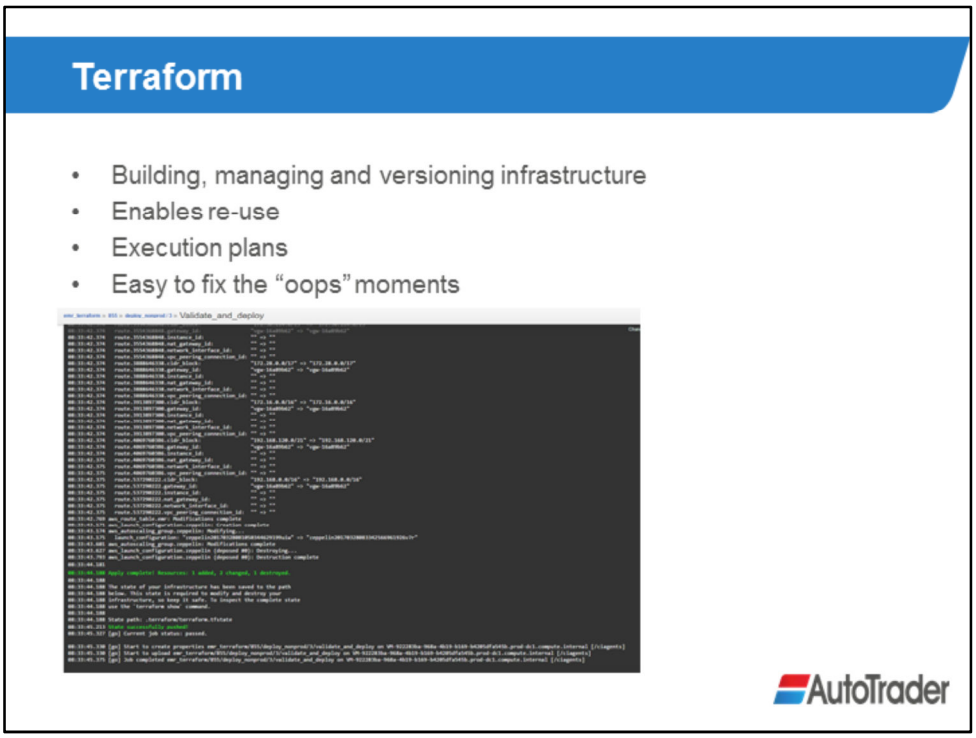
## What are your options?

- NAPALM
  - Network Automation and Programmability Abstraction Layer with Multivendor support
  - Python library
  - Set of functions to interact with different router vendor devices via unified API
  - https://github.com/napalm-automation/napalm
- OpenConfig
  - Written in YANG
  - Not an API but about models
  - http://www.openconfig.net

AutoTrader

I have no use or knowledge on these but please if anyone wants to find out I am sure a great presentation will come out the other side of it.
Network Automation and program-abilility abstraction layer with multi-vendor support
It's only a python library  so easy to use
Give you a set of functions to interact with different router vendor devices via the unified API (Cisco, Juniper, Fortinet,  Mikrotik, Palo Alto +more)
Github

OpenConfig
It's not an API but about how that data should be represented




Terraform by HashiCorp which we have just started using for a project we are hosting in AWS
Allows you to build, manage and version your infrastructurere
"object orientated" so you can re-use parts of the configuration without having to copy and paste
Tells you what it's about to do so you don't get a nasty surprise

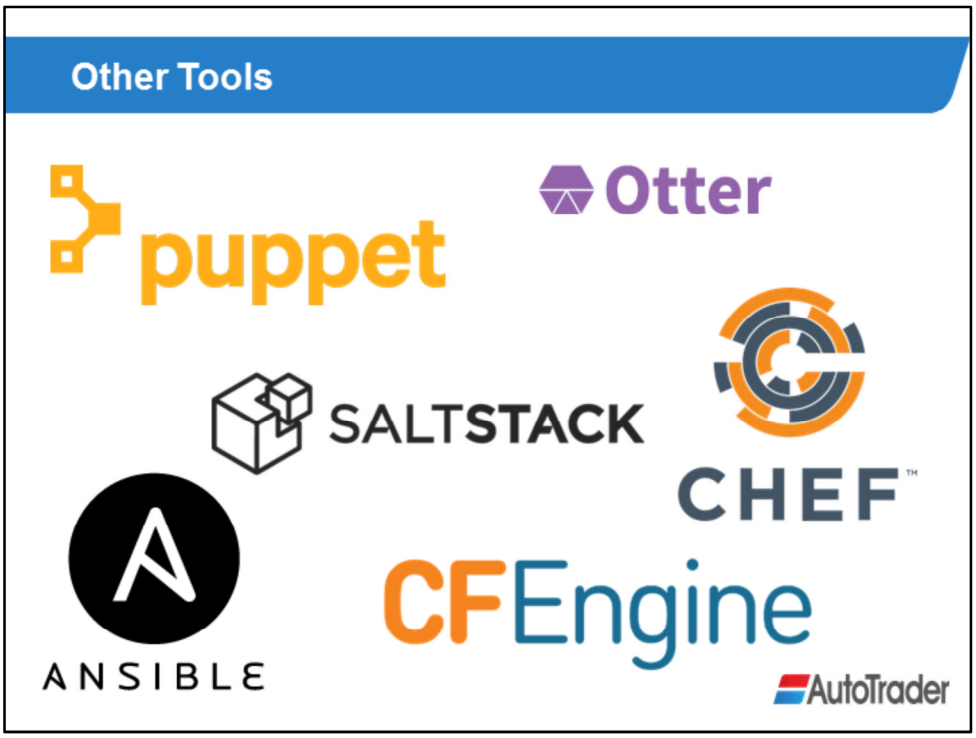
## Quote on Terraform

Terraform has enabled us to roll out complete network config at the touch of a button with full version control. We want to keep the same network setup in all environments and Terraform allows us to roll out changes to all environments at the same time, ensuring we have identical network configuration.

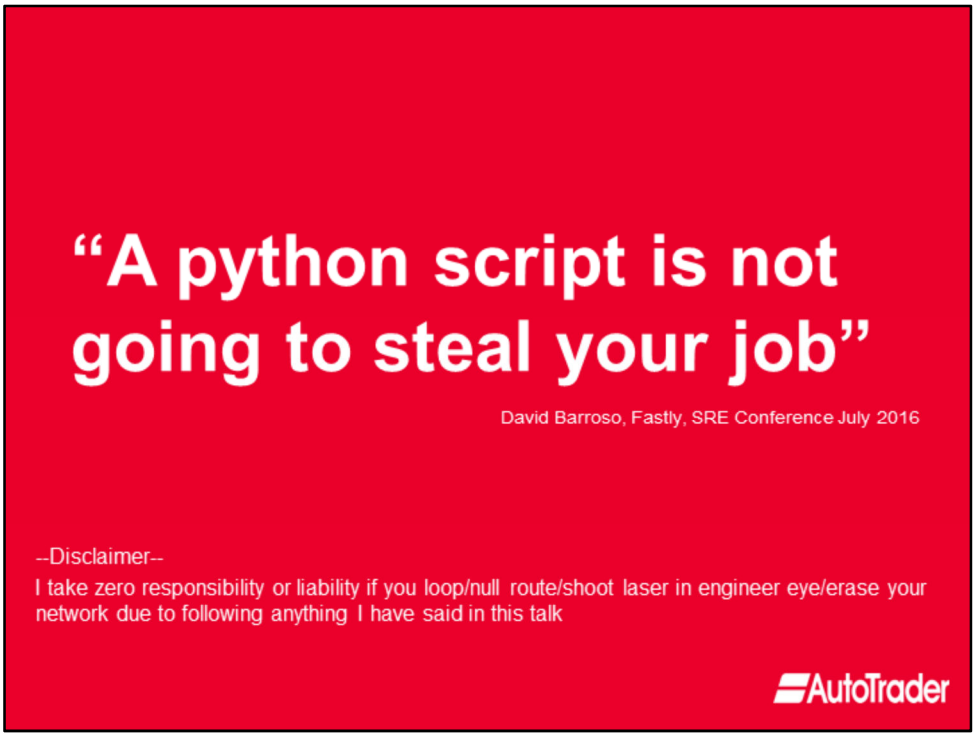
Chris Mckean - Senior Network Engineer

AutoTrader

Chris one of the Network engineers was working on the project where we were using AWS to host some big data stuff. So was his first time doing anything like this.




I won't go into these tools but most of them you have probably heard of or used. NAPALM will integrate without too much difficulty into thins like Ansible




Last notes from me, a python script is not going to steal your job, however it will probably make it a lot easier.
Also I take no liability if you have listened to what I have said and you end up breaking your network!

## References

- Past, Present, and Future of Network Operations
  - David Barroso, Fastly
  - https://srecon16europe.sched.com/event/7VkB/past-present-and-future-of-network-operations
- AutoTrader… from continuous integration to continuous delivery
  - Mark Crossfield
  - http://markcrossfield.co.uk/2014-10-22-devops-manchester-auto-trader-from-continuous-integration-to-continuous-delivery.html

AutoTrader

So there are two presentations that a reasonable amount of this is based on. One is a presentation from SRE (Site Reliability Engineering) Con in Dublin last year.
The other is from my colleague Mark Crossfield who went into a lot of detail 3 years ago when we were “half way through” the journey.