Cabinet Office

Sam Sharpe
Infrastructure Engineer
Government Digital Service
@samjsharpe

Used to work for Rackspace Hosting
Before that used to work for Imperial College
Before that got an Electronic Engineering degree at IC.

Building GOV.UK

Sam Sharpe GDS

GOV.UK is a (big) publishing website

Sam Sharpe GDS

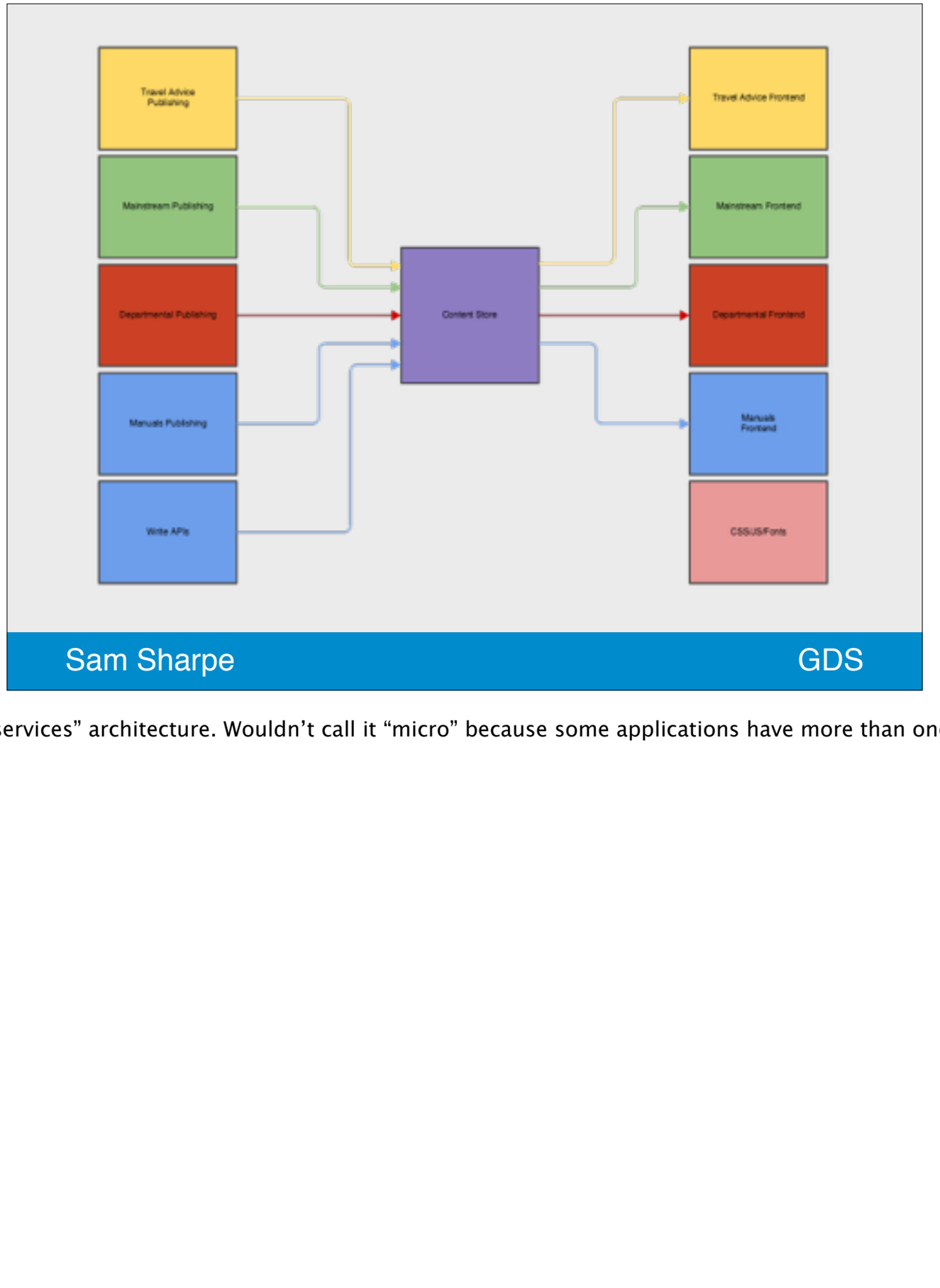
It “publicly” launched (Oct 2012) after 6 months work with the minimum functionality to support information for Citizens

A year later, we had the functionality needed for publishing new government information (and improved the other stuff all the time).

A year after that we had all the stuff needed to move most of the content from other Government websites (and the tools to do it).

GOV.UK is Composed of 60+ different custom applications for publishing and display.

In March 2014 it had:
- 100,000 items of content (webpages)
- 250GB of attachments (downloads from webpages)
- 20+ databases (ranging from 6MB to 6GB in size)



Simplified outline of the GOV.UK “mini-services” architecture. Wouldn’t call it “micro” because some applications have more than one function.

Government moves extremely fast and changes direction often

Sam Sharpe GDS

GOV.UK is always going to be incomplete. Government is changing all the time. What’s important is that we can go from an idea to being useful in as short a time as possible.

Reasons we need a pipeline:

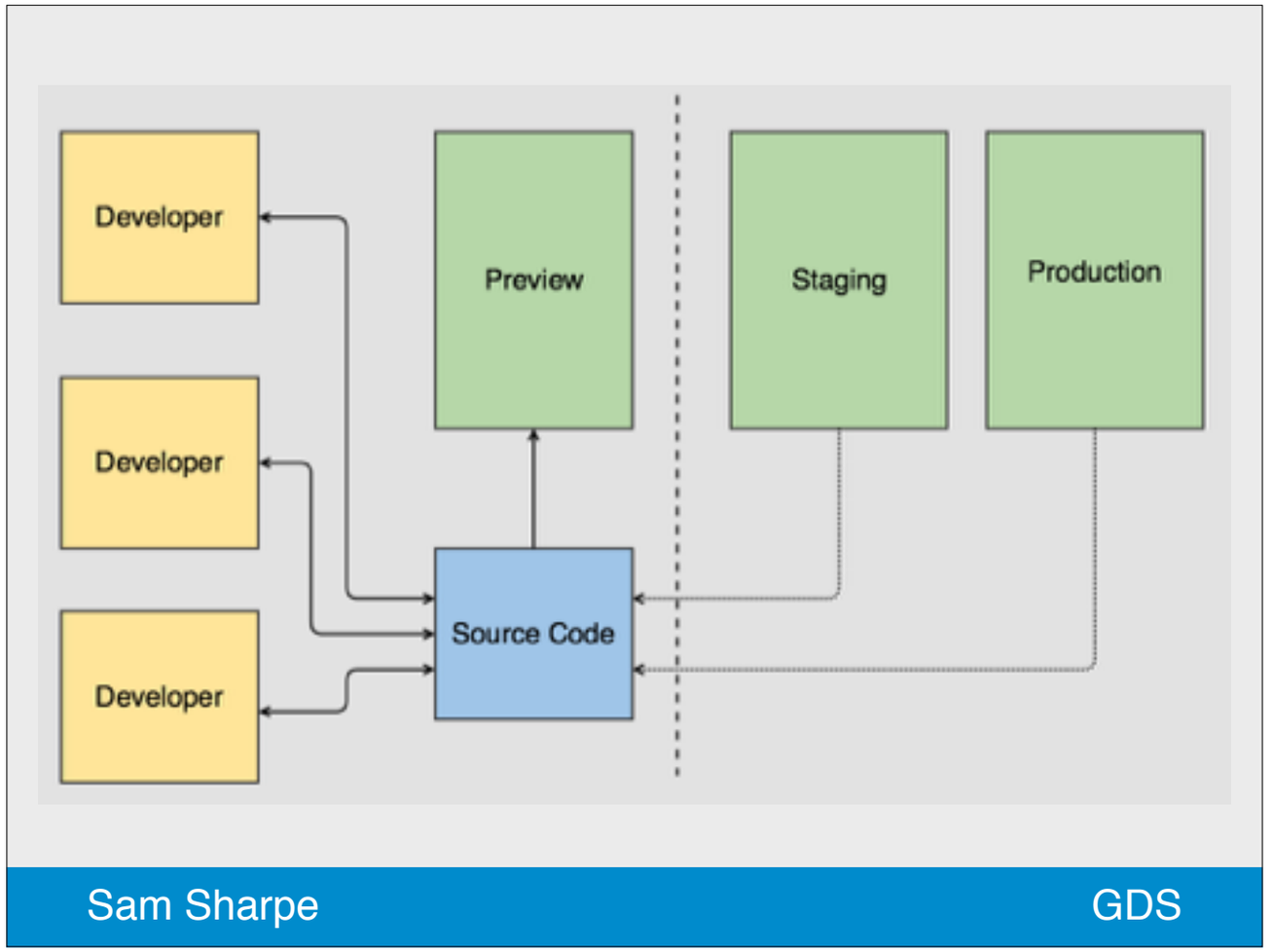
- Lots of developers
- Cross-functional working
- New developers join regularly

250+ developers and writers over the past 3 years. Everybody works on more than one application. It’s a “mini-services” architecture.

The GOV.UK Pipeline

Sam Sharpe GDS

Simplified (but not by much) view of the GOV.UK deployment pipeline to introduce some terms.



Top right are the environments we are deploying to
Bottom is source code
Left is the Developers and their laptops

Developing locally

- Scalable to many developers
- Can run some or all of the applications as needed
- Mechanisms exist to get Production data

Everyone has a laptop
Some Contractors bring their own (so we need to be OS agnostic)
Everyone has their own favourite development environment
Solution: Have a development VM that can run a copy of GOV.UK
- Uses the same config management as the real infrastructure
- Replicated the various parts of the stack in one VM
- Allows consistent development across various developers
- Gets developers up and running ASAP.

Using a development VM

Sam Sharpe GDS

Quick demo to show a development VM

Test while you work for fast feedback

Sam Sharpe GDS

Most GOV.UK applications are publicly hosted (github.com/alphagov). There's a lot to be said for hosting code publicly if you can as it gives you access to a lot of free testing tools.

We commonly run Travis against branches, sometimes only running a subset of the tests (non-integration), so every time a branch is pushed, we can see feedback on whether the tests still pass.

Good if they integrate with your sourcecode hosting, because then you can annotate code review requests.

Once code is reviewed and accepted, we merge to the master branch. At this point we move onto our own infrastructure.

Post-merge test and deploy

- GitHub triggers CI (Jenkins)
- CI runs tests
- CI triggers a Preview deployment
- CI *may* run further tests against deployed application

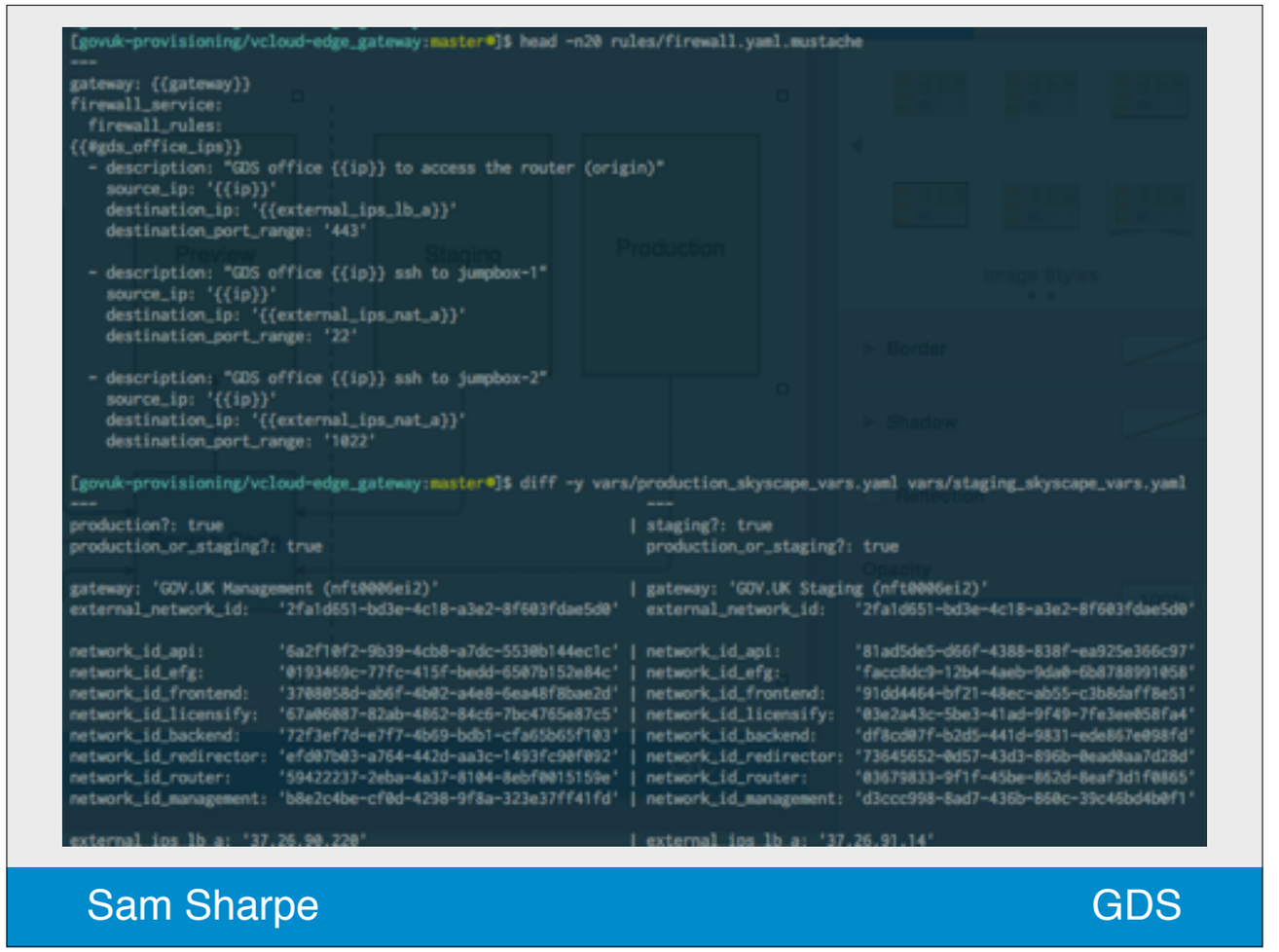
We have our own CI farm because it's easier to run integration tests on an environment you fully control (you can load data, run multiple applications).

It's currently Jenkins, but we've experimented recently with replacing it with Drone and/or Travis Enterprise.

Environments

- Preview - showing changes
- Staging - testing deploys
- Production - the real thing

They are all (roughly) the same:
- No point deploying to something different to Production and expecting it to work.
- They are all in the cloud
    - So we can easily expand/create/destroy them
    - Because owning hardware isn't a good economic proposition for Government
- Staging has a copy of yesterday's production data and is identical to Production in every way. We replay Production traffic against Staging 24/7.
- Preview has a slightly modified (we remove unpublished content) copy of yesterday's Production data.



Sam Sharpe GDS

- Having environments the same is easy in The Cloud and really important if you are doing it at scale.
    - We have full configuration management of our Network, VMs, Firewalls and Operating System
- Same firewall rules - common ruleset between all environments
- Difference in variables only
- We wrote our own provisioning tool which relies on Fog to enable this on VMWare

```
[govuk-provisioning/vcloud-launcher:master●]$ diff -y production_skyscape staging_skyscape | head -n 322 | tail -n 35
- name: backend-lb-2                                    - name: backend-lb-2
  vdc_name: GOV.UK Backend (IL2-PROD-ENHANCED)            vdc_name: GOV.UK Backend (IL2-PROD-ENHANCED)
  catalog_name: packer                                    catalog_name: packer
  vapp_template_name: ubuntu_precise64_20141023           vapp_template_name: ubuntu_precise64_20141023
  vm:                                                     vm:
    hardware_config:                                        hardware_config:
      memory: '4096'                                          memory: '4096'
      cpu: '2'                                                cpu: '2'
    network_connections:                                    network_connections:
    - name: Backend                                         - name: Backend
      ip_address: 10.3.0.102                                  ip_address: 10.3.0.102
    bootstrap:                                              bootstrap:
      script_path: vcloud-launcher/preamble/preamble.sh.erb   script_path: vcloud-launcher/preamble/preamble.sh.erb
      vars:                                                   vars:
        master_ip: 10.0.0.5                                     master_ip: 10.0.0.5
    storage_profile: 4-3-62-ENHANCED-Storage2             |   storage_profile: 4-3-63-ENHANCED-Storage2
- name: datainsight-1                                   <
  vdc_name: GOV.UK Backend (IL2-PROD-ENHANCED)          <
  catalog_name: packer                                  <
  vapp_template_name: ubuntu_precise64_20141023         <
  vm:                                                   <
    hardware_config:                                    <
      memory: '2048'                                    <
      cpu: '1'                                          <
    network_connections:                                <
    - name: Backend                                     <
      ip_address: 10.3.0.30                             <
    bootstrap:                                          <
      script_path: vcloud-launcher/preamble/preamble.sh.erb <
      vars:                                             <
        master_ip: 10.0.0.5                             <
    storage_profile: 4-3-63-ENHANCED-Any                <
- name: elasticsearch-1                                   - name: elasticsearch-1
  vdc_name: GOV.UK Backend (IL2-PROD-ENHANCED)            vdc_name: GOV.UK Backend (IL2-PROD-ENHANCED)
  catalog_name: packer                                    catalog_name: packer
```

Sam Sharpe GDS

- Same machines in Production and Staging
- Easy to see differences - note diff storage profile
- Note extra machine!!!! in Production

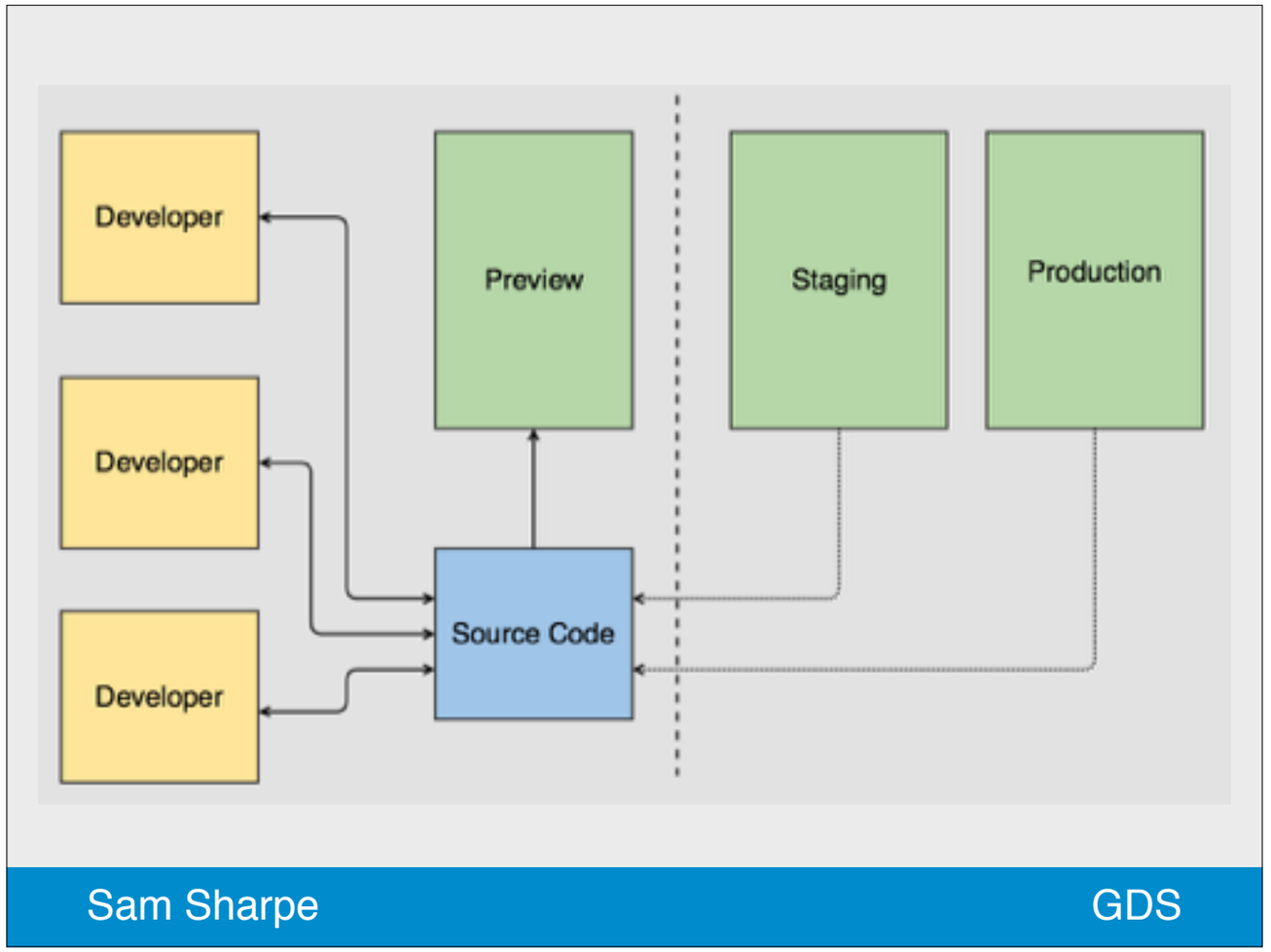
```
[gds/vagrant-govuk:master●]$ ls
total 24
-rw-r--r--  1 sam   2.4K 29 Nov 14:43 README.md
-rw-r--r--  1 sam   2.8K 29 Nov 14:43 Vagrantfile
-rw-r--r--  1 sam   1.5K 29 Nov 14:43 load_nodes.rb
[gds/vagrant-govuk:master●]$ head load_nodes.rb
require 'yaml'

# Load node definitions from the vcloud-launcher YAML in the
# govuk-provisioning repo parallel to this.
def load_nodes
  yaml_dir = File.expand_path(
    "../../govuk-provisioning/vcloud-launcher/preview_carrenza/",
    __FILE__
  )
  yaml_local = File.expand_path("../nodes.local.yaml", __FILE__)
[gds/vagrant-govuk:master●]$ vagrant status
Current machine states:

api-1.api                                 not created (virtualbox)
api-2.api                                 not created (virtualbox)
api-lb-1.api                              not created (virtualbox)
api-lb-2.api                              not created (virtualbox)
api-mongo-1.api                           not created (virtualbox)
```

Sam Sharpe GDS

- Even have a full vagrant setup based on the same definitions so if you have a really big laptop, you can test all 40 node types.



Back to the diagram of our pipeline to talk about deployment

- All deployment is controlled via a Jenkins instance within that particular environment
- Production/Staging deploys are manually triggered

Jenkins is used as a job runner with a web interface.
- Ensures repeatable deploys
- Ensures that all applications are deployed the same way and we don't get missed steps
- Can run smoke tests after the deploy and chain steps
- Logs the deployment in our registry of deployments

Capistrano is actually used at the backend as a deployment framework - we use that for consistency, even for non-Ruby applications like Go and Python.

Staging/Production deploys are manually triggered because the management environment isn't on the internet and because we control the number of people who can set off a deployment.

Questions?

Cabinet Office

Sam Sharpe
Infrastructure Engineer
Government Digital Service
@samjsharpe