droplr

DevOps in Droplr

Antoni Orfin <antoni@droplr.com>
Solutions Architect @ Droplr

What is Droplr?

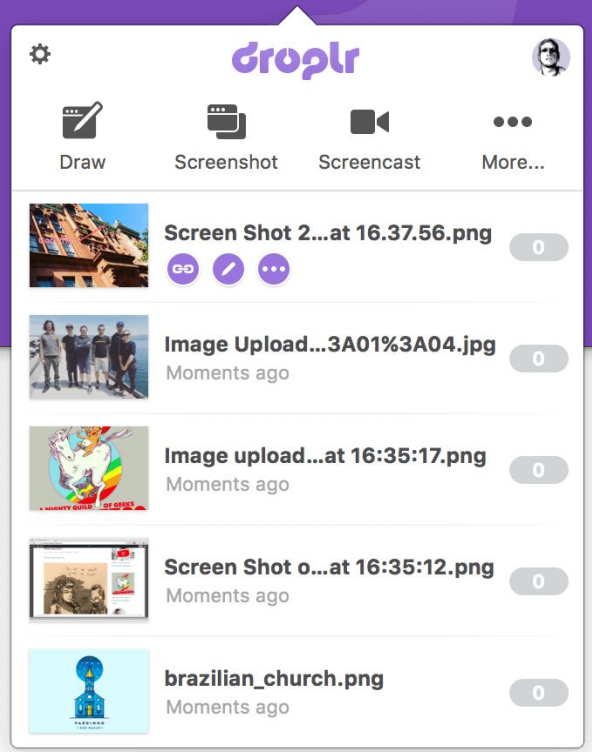
File sharing and screen capture tool
(with gif/video recording)

Used by...

NIKE YAHOO! intel TED Adobe PIXAR cisco TARGET

Droplr in numbers:

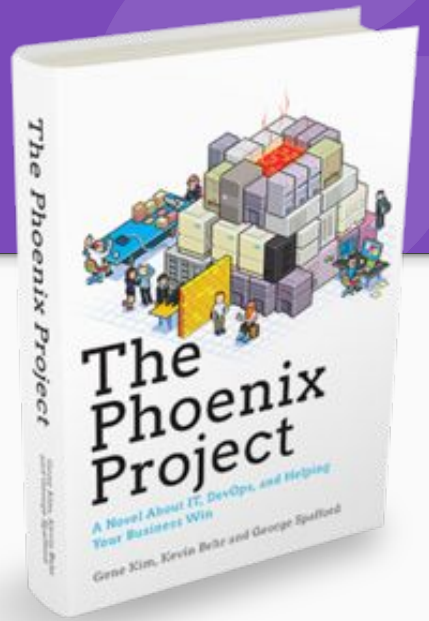
+ 450k users
+ 50M files (70 TB) in AWS S3



Role of DevOps

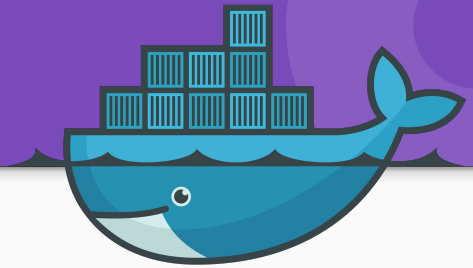
Make the company **competitive** on the market, by using right technologies and best software development practices.

+ Scalability
+ Stability, Maintainability
+ Quality
    + **of the business?**



The Phoenix Project: A Novel About IT, DevOps, and Helping Your Business Win
Gene Kim, Kevin Behr, George Spafford

Technologies overview



amazon web services™

Microsoft Azure

node JS

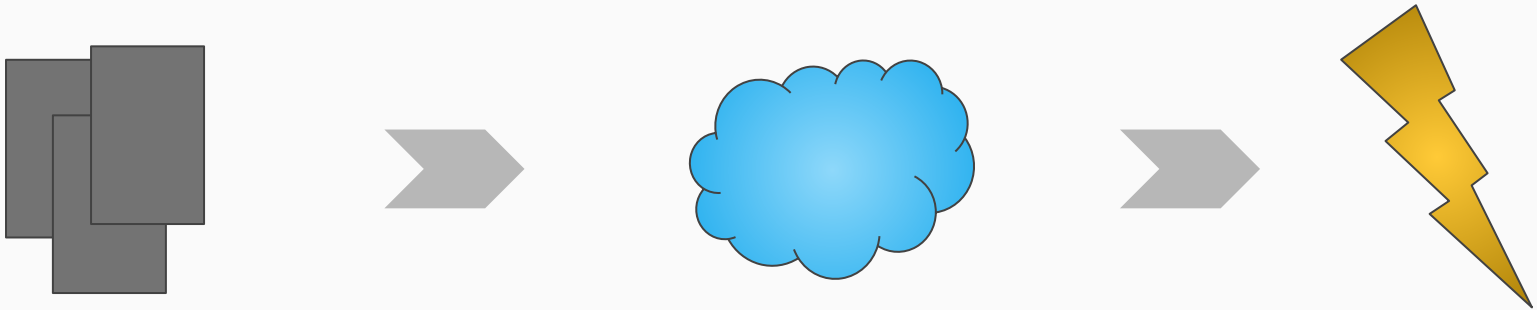
RAILS

mongoDB.

elastic

redis

Infrastructure **evolution**



Bare metal

IaaS
PaaS

FaaS

Infrastructure **in AWS**

Services we use:

1. Route53 (Cloud DNS)
2. VPC (Virtual Private Cloud)
3. Amazon EC2 (with autoscaling)
4. Amazon ELB (Elastic Load Balancer)
5. Amazon ECS (EC2 Container Service)
6. Amazon CloudWatch (Monitoring, Statistics)
7. **And going more into Amazon Lambda (FaaS)**

Infrastructure **in AWS**

Pros

+ Reduced operations tasks
+ Autoscaling
+ Many regions to run your platform

Cons

- Steep learning curve
- Very easy to build it wrong...

Infrastructure **as Code**

Goals:

1. Automatically set-up our infrastructure in the different geographic region.
2. Easily create new environments with the same configuration as production.
3. Make the infrastructure easy to maintain.
4. **...Continuous Deployment?**

Infrastructure **as Code**

1. Set-up infrastructure

Terraform - definition of our infrastructure (e.g. AWS configuration) in declarative code.

2. Run deploy commands

Ansible - configuration management, deployments (e.g. set up MongoDB).

Infrastructure **as Code**

1. Terraform

- Supports multiple providers (AWS, Azure)
- Creates readable plan of deployment (list of created/destroyed/changed resources)
- Compose infrastructure in modules
- Keeps state of the infrastructure in file

- *CloudFormation? :-(*

```
49
50  resource "aws_alb_target_group" "default" {
51    name     = "${var.tags["Env"]}-${var.tags["App"]}-defau
52    port     = 80
53    protocol = "HTTP"
54    vpc_id   = "${var.vpc}"
55
56    tags = "${merge(var.default_tags, var.tags)}"
57  }
58
59  resource "aws_alb_target_group_attachment" "default" {
60    target_group_arn = "${aws_alb_target_group.default.arn}
61    target_id        = "${aws_instance.master.id}"
62    port             = 8080
63  }
64
65  # Jenkins master instance
66  resource "aws_instance" "master" {
67    ami = "${var.instance_ami}"
68    instance_type = "${var.instance_type_master}"
69    key_name = "${var.instance_key_name}"
70
71    vpc_security_group_ids = ["${aws_security_group.ec.id}"
72    subnet_id = "${var.subnets[0]}"
73
74    # We need to provision python2.7 on machines as Ubuntu
75    provisioner "remote-exec" {
76      script = "../provisioners/setup_ubuntu.sh"
77
78      connection {
79        type     = "ssh"
80        user     = "${var.instance_ssh_user}"
```

Infrastructure **as Code**

1. Terraform directories structure

```
/env-prod
  /db.tf
  /api.tf # just includes api module
  /terraform.tfstate   # state file
/modules
  /api
  /db
  /jenkins
```

```
49
50  resource "aws_alb_target_group" "default" {
51    name     = "${var.tags["Env"]}-${var.tags["App"]}-defau
52    port     = 80
53    protocol = "HTTP"
54    vpc_id   = "${var.vpc}"
55
56    tags = "${merge(var.default_tags, var.tags)}"
57  }
58
59  resource "aws_alb_target_group_attachment" "default" {
60    target_group_arn = "${aws_alb_target_group.default.arn}
61    target_id        = "${aws_instance.master.id}"
62    port             = 8080
63  }
64
65  # Jenkins master instance
66  resource "aws_instance" "master" {
67    ami = "${var.instance_ami}"
68    instance_type = "${var.instance_type_master}"
69    key_name = "${var.instance_key_name}"
70
71    vpc_security_group_ids = ["${aws_security_group.ec.id}"
72    subnet_id = "${var.subnets[0]}"
73
74    # We need to provision python2.7 on machines as Ubuntu
75    provisioner "remote-exec" {
76      script = "../provisioners/setup_ubuntu.sh"
77
78      connection {
79        type     = "ssh"
80        user     = "${var.instance_ssh_user}"
```

Infrastructure **as Code**

2. Ansible

- Basically - a list of commands that will be run via SSH on the server to set-up services (declarative?)
- Small learning curve
- Server has *roles* (e.g. mongodb, nginx)

```
---
- name: add jenkins key to apt-key
  apt_key:
    url: "{{ jenkins_key_url }}"
    state: present

- name: add jenkins repository
  apt_repository:
    repo: 'deb {{ jenkins_repository }}'
    state: present

- name: ensure jenkins is installed
  apt: pkg=jenkins state=present update_cache=yes

- name: download jenkins-client
  get_url:
    dest: "{{ jenkins_client_dest }}"
    url: "{{ jenkins_client_url }}"

- name: download jenkins plugins
  with_items: "{{ jenkins_plugins }}"
  get_url:
    dest: "{{ jenkins_home }}/plugins/{{ item }}.jpi"
    url: "https://updates.jenkins-ci.org/latest/{{ item }}.
    owner: jenkins
    group: jenkins
    mode: 0644
  notify: restart jenkins

- name: generate SSH key
  shell: ssh-keygen -b 4096 -t rsa -C "{{ jenkins_ssh_key_e
  become: True
  become_user: jenkins
  args:
```

Continuous Deployment

Build Docker images

Create new ECS definition

Run functional tests

Run performance tests

Deploy stage

Deploy production

BrowserStack

Se

APACHE JMeter™

Future of Droplr

Long-term goals

1. Reducing complexity of the platform
   → quicker to develop new features

2. Trying serverless architectures
   → easier to maintain

3. Setting up platform in new geo-regions
   → faster for end-users

Thanks

Time for discussion :-)

- Do you use Automation Tools and Continuous Deployment?
- What’s your opinion about Cloud?
- …