SHIP IT

EASY DEPLOYMENTS WITH JENKINS , FABRIC, AND PIP

DAN TRACY - PYCON 2015

WHO AM I?

- Dan Tracy
- @djt5019 on most things
- Backend Dev @ AWeber
- I like DevOps'y stuff
- I like deployment stuff

WHAT'S THIS ABOUT?

Deploying should be easy as pushing a button

Reverting should be easy as pushing a button

Bad deployment processes impede development and innovation

Good deployment processes foster development and innovation

Code isn’t useful until it’s being used.

HYPOTHETICAL SITUATION

So you have a new feature ready for production?

You have a bug-fix that's needed in production 10 minutes ago?

All that's left is to deploy it!

SHIP IT!

WHAT'S THE WORST THAT CAN HAPPEN?


RENA
MONRONIA

DO YOU HAVE...

- More than one way to deploy code across teams?
- More than one way to deploy code amongst a team?
- More than one way to deploy a single app?!?

OUR OLD DEPLOYMENT SETUP

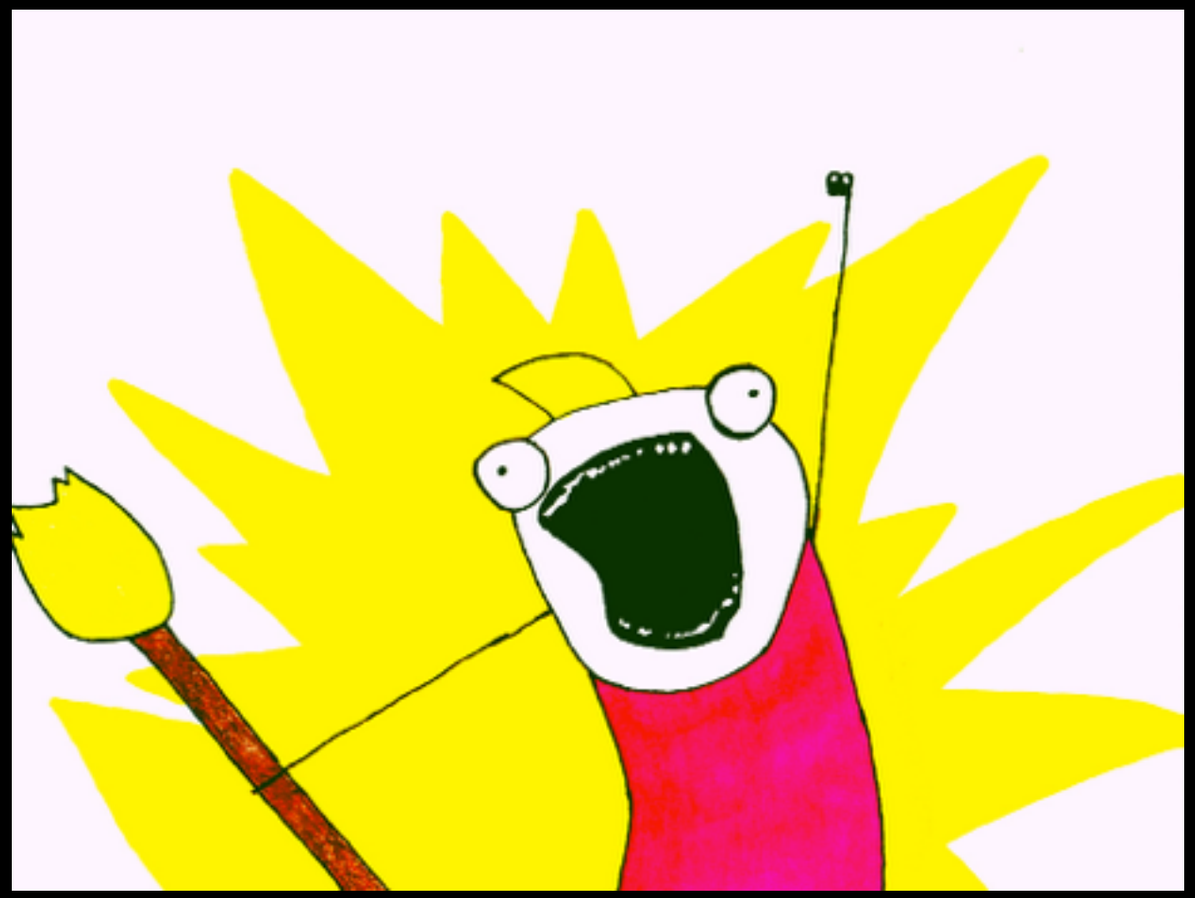
- Debian packaging
- SCP un-versioned tarballs
- Chef installing packages
- PIP installations

CHOOSE ONE AND

ITERATE

MAINTAIN ONE STANDARD WAY

WE PICKED PYTHON BASED PACKING



PYTHON PACKAGING

- Python & Setuptools to build packages
- PEP 440 versioned packages (setupext-gitversion)
- Pip to install Python packages
- Jenkins for CI and package building
- Chef for initial deploy and service discovery
- Fabric for gluing the pieces together

HOW DOES IT WORK?

Devs merge commits to master. Github webhooks notify project specific Jenkins job

Webhook notification

Jenkins job runs tests, builds a package, uploads it to an internal PyPI server

Upload internal package

Project specific Jenkins job triggers generic deployment job

Parameterized job trigger

Generic deployment job queries Chef server via Fabric

Chef node search

SSH into each node and upgrade the Python package

SSH into node

pip install -U

Job restarts service which loads new code

kill -HUP <pid>

Test the new code

Control returns to the project specific job

Return status of deployment

Notify monitoring that a deploy happened, include version information and timestamp.

Webhook notification

THAT'S ALL THERE IS TO IT

TAKEAWAYS

- Choose one deployment strategy and iterate on it
- Code can be deployed in parallel
- Services should be bounced sequentially
- Centralized deployment job makes for easy rollbacks
- Other codebases can use similar deployment setup

SOME SHORTCOMINGS...

- Fixing old code deployment is **very** painful.
- Strongly coupled with Chef for service discovery.
- Jenkins not the greatest continuous deployment server.
- Initial hacky usage of Fabric in early stages.

NEXT STEPS

- All aboard the Docker hype-train.
- RunDeck instead of Jenkins as a continuous delivery server?
- Trend monitoring using something like SkyLine/Riemann.
- Using Consul/Etcd/ZK for service discovery in place of Chef
- More sophisticated deployments; partial rollouts, rollouts to a “canary”.

TL;DR

- Good deployments can help innovation
- Bad deployments can hurt it
- Deployment pipelines are never “done” or “finished”
- Deploy quickly; Deploy often
- Iterate on what works

Thank You!

Github: djt5019

Twitter: @djt5019