PIKU

"THE TINIEST PAAS YOU'VE EVER SEEN"

HTTPS://GITHUB.COM/PIKU/PIKU

John Flynn Matthew

AGENDA

Introduction

What / Why a Platform-as-a-Service

Why PIKU?

Features of PIKU

Architecture

Hands on

INTRODUCTION

Hello, my name is John Matthew.

Been in the IT industry for 44 yrs. (yikes). I love programming.

Started with dBase, FoxPro, Delphi. Now I spend my spare time in Python and Django.

I discovered PIKU and want to share my experience and excitement for it

WHAT IS PLATFORM-AS-A-SERVICE

Heroku

"Heroku is a cloud platform that lets companies build, deliver, monitor and scale apps — we're the fastest way to go from idea to URL, bypassing all those infrastructure headaches.”

Google (App engine)

"Platform as a Service, also known as PaaS, is an environment that offers a flexible, scalable cloud platform to develop, deploy, run, and manage apps.

…

PaaS provides everything developers need for application development without the headaches of updating the operating system and development tools or maintaining hardware."

Microsoft (App Service)

"Platform as a service (PaaS) is a complete development and deployment environment in the cloud, with resources that enable you to deliver everything from simple cloud-based apps to sophisticated, cloud-enabled enterprise applications."

John

```
$>git push
```

Deploy Fast

`$>git push piku` is all you need to get new code to your server and deployed

Inexpensive

PIKU runs on a simple $5 DO droplet for as many apps as you like

Simple

For simple to moderate projects, I don't need containers, Kubernetes, Lamba, etc. I just want a simple NGINX + WSGI + Python app environment

Scalable

I can scale my app with one command

WHY PIKU?

PIKU'S CORE VALUES

Must run on low end devices

Accessible to hobbyists and K-12 schools

~1500 lines readable code

Few (single?) dependencies

12-Factor

Cover 80% of common use cases

Leverage standard tooling (`git`, `ssh`, `uwsgi`, `nginx`).

Sensible defaults for all features

PIKU FEATURES

Simple install — Install takes <2 min and you're up and running

Automatic SSL — Let's Encrypt integration provides SSL certs when you push code, and it generates self-signed ones just in case

`Procfile` — `Procfile` functionality from Heroku is utilized in PIKU.

Simple CLI — Uses SSH for pushing code and deployment

Provides a `piku` command line interface for managing your application: Logs, Scale, Restarts, environment

Static Sites

Ability to use NGINX as a Static Site only, i.e. HUGO, Pelican

Supported

Although PIKU is open source, the responses on GH from the creators and others, is impressive

Flexible

The ENV file can modify NGINX behavior, UWSGI, custom commands, etc.

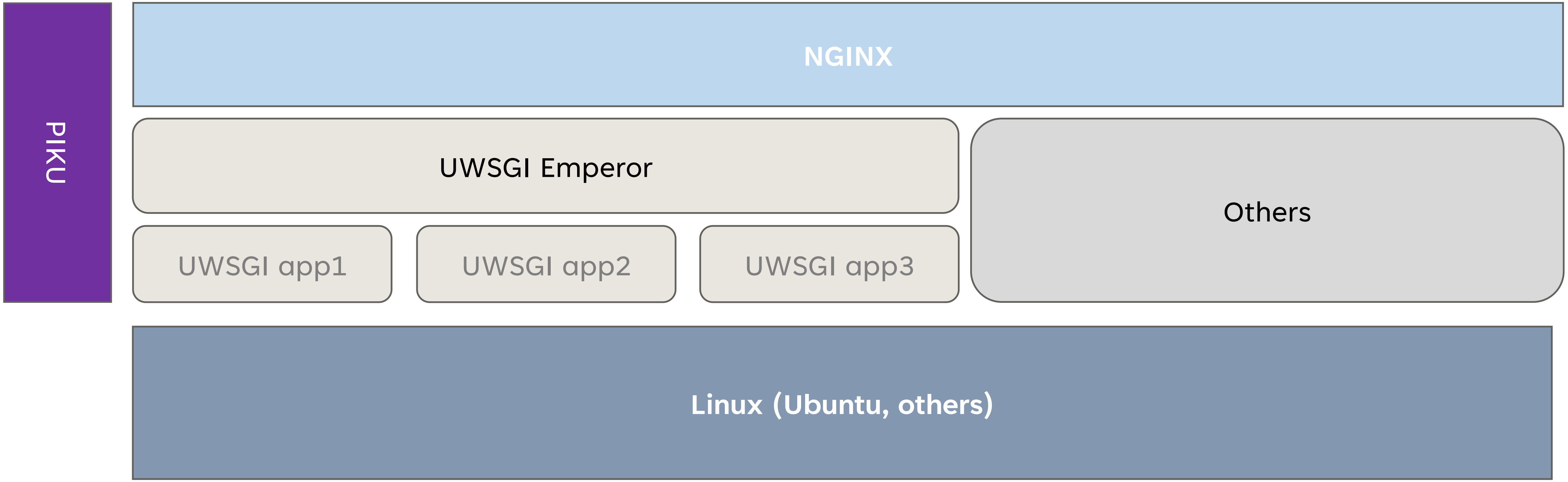
Bare bones

At the end of the day, it’s just NGINX and UWSGI bundled into a simple python script, with some SSH magic thrown in.

You’ll see in the DEMO ☺

PIKU FEATURES (CONT.)

PIKU ARCHITECTURE



DEMO TIME

1. Install (<5m)
2. Setup GIT
3. Push Django app
4. SSL test
5. PIKU cli

PLEASE – Ask questions during the demo

DEMO DETAILS

Piku Install on VM

```
curl https://piku.github.io/get | sh
./piku-bootstrap install
```

Clone Repo and follow README

```
git clone https://github.com/jfmatth/piku-django-scale.git
...
```

Configure and Push Django app

```
git push piku
piku config:add NGINX_SERVER_NAME=<fqdn>
```

SSL and CLI

HTTPS automatic once NGINX_SERVER_NAME is set

PIKU CLI examples

THANK YOU

John Matthew

John@compunique.com

https://www.github.com/jfmatth