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.

Currently work at a VAR promoting cloud service, DevOps and Automation for customers

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

WHAT IS PLATFORM-AS-A-SERVICE

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
Simplify user experience.	Cover 80% of common use cases	Leverage standard tooling (`git`, `ssh`, `uwsgi`, `nginx`).	Sensible defaults for all features

Simple install ————— Install takes <2 min and you're up and running
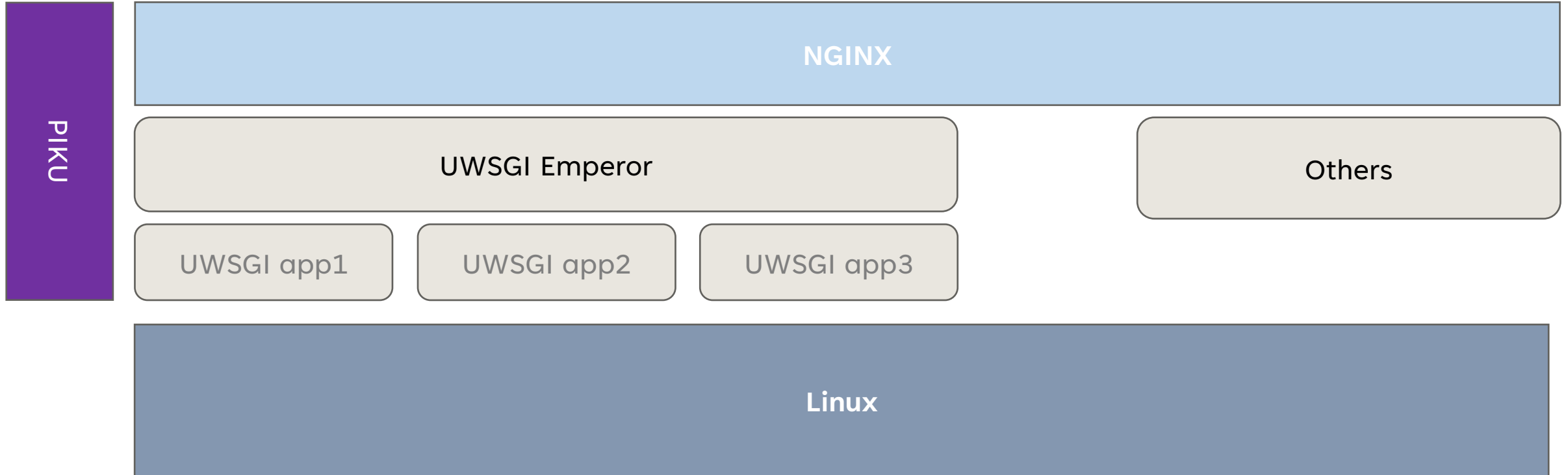
Automatic SSL ————— Let's Encrypt integration provides SSL certs when you push code and it generates self-signed ones just in case

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

PIKU FEATURES

PIKU ARCHITECTURE

PIKU

NGINX

UWSGI Emperor

Others

UWSGI app1

UWSGI app2

UWSGI app3

Linux

DEMO TIME

1. Install (<5m)

2. Setup GIT

3. Push Django app

4. SSL test

5. PIKU cli

Questions during are welcome

Piku Install on VM ──────────
```
curl https://piku.github.io/get | sh

./piku-bootstrap install

./piku-bootstrap install postgres.yml
```

Setup GIT on client ──────────
```
git clone https://github.com/jfmatth/piku-django-scale.git

pipenv shell

git remote add piku piku@server:appname
```

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

DEMO DETAILS

THANK YOU

John Matthew

John@compunique.com

https://www.github.com/jfmatth