



# FIVE WAYS TO DEPLOY YOUR PYTHON WEB APP IN 2017

Andrew T. Baker  
PyCon 2017



# ABOUT ME



# TODAY



# Flask

web development,  
one drop at a time

[overview](#) // [docs](#) // [community](#) // [snippets](#) // [extensions](#)

*Flask is a microframework for Python based on Werkzeug, Jinja 2 and good intentions. And before you ask: It's [BSD licensed](#)!*

## Flask is Fun

Latest Version: [0.12](#)

```
from flask import Flask
app = Flask(__name__)

@app.route("/")
def hello():
    return "Hello World!"

if __name__ == "__main__":
    app.run()
```

## And Easy to Setup



# TECHNIQUE #1

# ngrok

# PROS

- Fast and easy
- Handy for demos
- And hacking on webhooks

# CONS

- Stops when you close your laptop
- Random domains
- **Definitely** doesn't scale





## TECHNIQUE #2

# Heroku

# PROS

- One app 24/7 for free
- Zero server management
- Add-ons ecosystem

# CONS

- Scaling is easy but gets pricey
- Harder server customization
- Some add-ons better than others







## TECHNIQUE #3

# “Serverless”

# PROS

- Economical for small to medium loads
- Good for “spikey” traffic
- Zero server config

# CONS

- Relatively new technique
- Less fun without Zappa
- Can be tricky to troubleshoot





## TECHNIQUE #4

# Virtual Machines

# PROS

- Full control
- Scales as much as your wallet
- Economical... if you're careful

# CONS

- More work for you
- There's a lot more to learn
- Harder to predict ultimate costs





## TECHNIQUE #5

# Docker

# PROS

- Helps with dev / prod parity
- Nice for “microservices”
- Impress your friends

# CONS

- Newest technique
- Works best when you go all-in
- Has its own learning curve





 Full Stack Python  
Guide to Deployments

by Matthew Makai



**THANK YOU**  
**@andrewtorkbaker**

