

# Serverless LTI Learning Tools on AWS Lambda

---

Dr. Aneesha Bakharia  
Senior Developer & Development Team Leader  
John Zornig  
Director, UQx



THE UNIVERSITY  
OF QUEENSLAND  
AUSTRALIA



## UQX STATISTICS

---

- UQx formed in 2013
- Produced 53 MOOCs to date
- 4 Micromasters
- 2 million + registrations to date

# Current LTI Development

## ..... LTI Tools .....

- Mainly developed in either Php or Python (Django)
- Researching LTI deployment to AWS Lambda
- Want ability to revert back to server deployments if required
  - High no of requests (Cost?)
  - Additional functionality (larger memory)
  - Move to Azure?

# What is Serverless with AWS Lambda?

"without any permanent infrastructure."

- There still is a server - but it only has a 40 millisecond life cycle!
- Each request is given its own virtual HTTP "server"
- Simple functions can be implemented in Python or Node.js
- AWS handles the horizontal scaling automatically

# Solution: Zappa

## ZAPPA

### Deploy your WSGI apps on AWS Lambda

With Zappa, each request is given its own virtual HTTP "server" by Amazon API Gateway. AWS handles the horizontal scaling automatically, so no requests ever time out. After your app returns, the "server" dies.

- No more tedious web server configuration!
- No more paying for 24/7 server uptime!
- No more worrying about load balancing / scalability!
- No more worrying about web server security!

```
~/demo $ cd demo
~/demo $ ls
env          my_app.py      zappa_settings.json
~/demo $ source env/bin/activate
(env)~/demo $ cat my_app.py
from flask import Flask
app = Flask(__name__)

@app.route('/')
def hello():
    return 'Hello, from Zappa!\n'

if __name__ == '__main__':
    app.run()
(env)~/demo $ cat zappa_settings.json
{
  "dev": {
    "s3_bucket": "lmbda",
    "app_function": "my_app.app",
    "parameter_depth": 1
  }
}
(env)~/demo $ zappa deploy dev
Packaging project as zip...
Uploading zip (5.8MiB)...
44% | ██████████ | 2.68M/6.03M [00:03<00:04, 764Kit/s]
```

<https://www.zappa.io/>

# Zappa

Full documentation:

<https://github.com/Miserlou/Zappa>

## ..... Features .....

- Zappa works with any WSGI app
- Use Web frameworks (Django or Flask)
- Python 2.7 and 3.6
- Able to deploy and update

# What does Zappa do?

- package up your application + local virtual environment into a Lambda-compatible archive
- replace any dependencies with versions precompiled for Lambda
- set up the function handler and necessary WSGI Middleware
- upload the archive to S3
- create and manage the necessary Amazon IAM policies and roles
- register it as a new Lambda function
- create a new API Gateway resource
- create WSGI-compatible routes for it, link it to the new Lambda function

# Zappa & Flask & LTI

- Flask web framework is preferred over Django (smaller memory footprint)
- Use PyLTI Library  
<https://github.com/ucfcdl/pylti>
- Proof of concept  
[https://github.com/UQ-UQx/awslambda\\_lti\\_starter](https://github.com/UQ-UQx/awslambda_lti_starter)



# What about a database?

- Use AWS DynamoDB
- DynamoDB is a NoSQL database
- Key-value store based
- Great to insert json but not good for dynamic queries (Beware: AWS charges per query and no powerful query language like MongoDB)
- <https://read.acloud.guru/why-amazon-dynamodb-isnt-for-everyone-and-how-to-decide-when-it-s-for-you-aefc52ea9476>

# THANK YOU

Contact: [uqx@uq.edu.au](mailto:uqx@uq.edu.au)

