# Infrastructure Automation: Real Examples


Karl Gutwin

*Sr Scientific Consultant, BioTeam*

# We live in an automated world

*infrastructure automation is robotics for IT*

BIOTEAM
Enabling Science

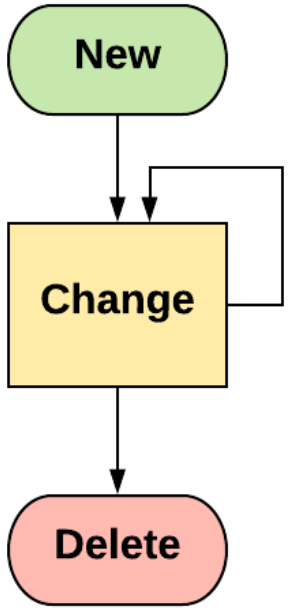
# Infrastructure as Code

Know the intent and implementation of your system

BIOTEAM
Enabling Science

# Automation is more than provisioning

- Managing changes
- Cleanup and deletion




BIOTEAM
Enabling Science

# Why not automation?

- Additional point of failure
- Too clever for its own good
- Code as a liability
- Sometimes additional costs

**These are all risks we currently manage within IT**

BIOTEAM
Enabling Science

# Many tools to choose from

*including*

- Ansible
- Docker
- Lambda
- CloudFormation

BIOTEAM
Enabling Science

# Examples

↳ github.com/bioteam/infrastructure-automation

# Ansible

Configure your server

# webserver.yml

```
---
- hosts: localhost
  tasks:
    - name: install nginx
      package: name=nginx state=present

    - name: set to run on boot
      service: name=nginx enabled=yes state=started
```

# database.yml

```
---
- hosts: localhost
  tasks:
    - name: install postgresql
      package: name=postgresql state=present

    - name: update config file
      lineinfile:
        path: /etc/postgresql/9.6/main/postgresql.conf
        regexp: '^synchronous_commit ='
        line: 'synchronous_commit = {{ pg_sync|default("on") }}'
      register: postgres_config

    - name: set to run on boot
      service: name=postgresql enabled=yes state=started

    - name: restart service
      service: name=postgresql state=restarted
      when: postgres_config.changed
```

```
ansible-playbook database.yml
```

```
 [WARNING]: provided hosts list is empty, only localhost is available. Note that the implicit localhost does
not match 'all'


PLAY [localhost] ***************************************************************************************

TASK [Gathering Facts] *********************************************************************************
ok: [localhost]

TASK [install postgresql] ******************************************************************************
ok: [localhost]

TASK [update config file] ******************************************************************************
ok: [localhost]

TASK [set to run on boot] ******************************************************************************
ok: [localhost]

TASK [restart service] *********************************************************************************
skipping: [localhost]

PLAY RECAP *********************************************************************************************
localhost                  : ok=4    changed=0    unreachable=0    failed=0

/demo/ansible#
```

# Get Started

↳ `pip install ansible`

↳ github.com/ansible/ansible-examples

↳ docs.ansible.com/ansible/latest/user_guide/playbooks_intro.htm

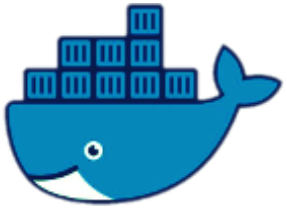
# Docker

Applications in Containers

# Dockerfile

```
FROM node

RUN git clone https://github.com/hakimel/reveal.js.git && \
    cd reveal.js && npm install

EXPOSE 8000
WORKDIR /reveal.js
CMD ["npm", "start"]
```

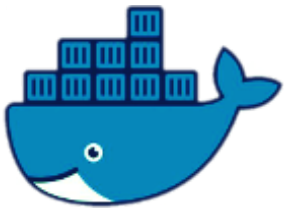
```
docker build -t my-image .
```



# docker-compose.yml

```
---
version: '3'
services:
  mediawiki:
    image: mediawiki
    restart: always
    ports: ['8080:80']
    links: ['database']
    volumes:
      - /var/www/html/images
#     - ./LocalSettings.php:/var/www/html/LocalSettings.php
  database:
    image: mariadb
    restart: always
    environment:
      MYSQL_DATABASE: my_wiki
      MYSQL_USER: wikiuser
      MYSQL_PASSWORD: example
      MYSQL_RANDOM_ROOT_PASSWORD: 'yes'
```

```
docker-compose up
```

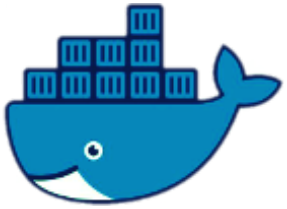


```
/demo# cd docker/mediawiki/
/demo/docker/mediawiki# docker-compose up
Starting mediawiki_database_1 ... done
Starting mediawiki_mediawiki_1 ... done
Attaching to mediawiki_database_1, mediawiki_mediawiki_1
database_1    | 2019-04-17 17:56:59 139955798349696 [Note] mysqld (mysqld 10.2.10-MariaDB-10.2.10+maria~jessie) s
tarting as process 1 ...
database_1    | 2019-04-17 17:56:59 139955798349696 [Note] InnoDB: Mutexes and rw_locks use GCC atomic builtins
database_1    | 2019-04-17 17:56:59 139955798349696 [Note] InnoDB: Uses event mutexes
database_1    | 2019-04-17 17:56:59 139955798349696 [Note] InnoDB: Compressed tables use zlib 1.2.8
database_1    | 2019-04-17 17:56:59 139955798349696 [Note] InnoDB: Using Linux native AIO
database_1    | 2019-04-17 17:56:59 139955798349696 [Note] InnoDB: Number of pools: 1
database_1    | 2019-04-17 17:56:59 139955798349696 [Note] InnoDB: Using SSE2 crc32 instructions
database_1    | 2019-04-17 17:56:59 139955798349696 [Note] InnoDB: Initializing buffer pool, total size = 256M, i
nstances = 1, chunk size = 128M
database_1    | 2019-04-17 17:56:59 139955798349696 [Note] InnoDB: Completed initialization of buffer pool
database_1    | 2019-04-17 17:56:59 139955062658816 [Note] InnoDB: If the mysqld execution user is authorized, pa
ge cleaner thread priority can be changed. See the man page of setpriority().
database_1    | 2019-04-17 17:56:59 139955798349696 [Note] InnoDB: Highest supported file format is Barracuda.
database_1    | 2019-04-17 17:56:59 139955798349696 [Note] InnoDB: 128 out of 128 rollback segments are active.
database_1    | 2019-04-17 17:56:59 139955798349696 [Note] InnoDB: Creating shared tablespace for temporary table
s
database_1    | 2019-04-17 17:56:59 139955798349696 [Note] InnoDB: Setting file './ibtmp1' size to 12 MB. Physica
lly writing the file full; Please wait ...
```

# Get Started

↳ www.docker.com/products/docker-desktop

↳ docs.docker.com/compose/gettingstarted/



# Lambda

Event-driven automation

# autotag.py

```
import boto3
ec2 = boto3.client('ec2')

def lambda_handler(event, context):
    principal_type = event['detail']['userIdentity']['type']
    if principal_type == 'IAMUser':
        username = event['detail']['userIdentity']['userName']
    else:
        username = event['detail']['userIdentity']['principalId']

    response = event['detail']['responseElements']
    instance_ids = [i['instanceId']
                    for i in response['instancesSet']['items']]

    ec2.create_tags(Resources=instance_ids,
                    Tags=[{"Key": "Owner", "Value": username}])
```

demo-autotag.sh

```
Launching EC2 instance ...
Instance ID: i-07ab4db1743f056cf

Watching instance state (Ctrl-C to abort) ...
Current time                 Instance ID         Launch time              State   Tags
-------------------------------------------------------------------------------------
Wed Apr 17 17:58:44 UTC 2019 i-07ab4db1743f056cf 2019-04-17T17:57:43.000Z running Owner karl
```

# Other ideas

- Manage DNS entries
- Integrate with a CMDB
- Automate application rollout
- Automatic health checks for critical applications
- Data pipelines and ETL

# Get Started

↳ docs.aws.amazon.com/AmazonCloudWatch/latest/events/CloudWatch-Events-Tutorials.html

# CloudFormation

AWS infrastructure templates

# bucket.yml

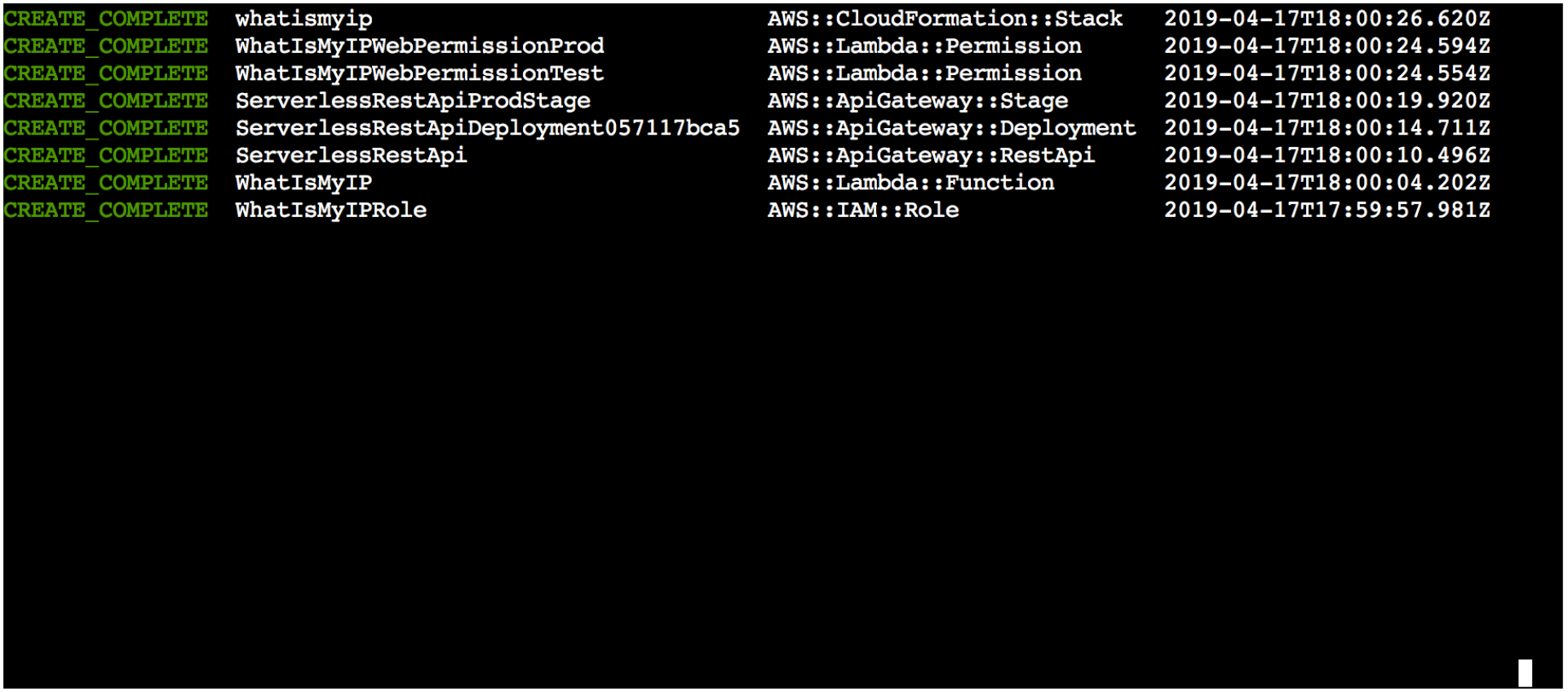
```yaml
---
AWSTemplateFormatVersion: "2010-09-09"
Parameters:
  BucketName:
    Type: String
    Default: my-awesome-bucket

Resources:
  StaticBucket:
    Type: "AWS::S3::Bucket"
    Properties:
      BucketName: !Ref BucketName
```

# whatismyip.yml

```yaml
AWSTemplateFormatVersion: "2010-09-09"
Transform: "AWS::Serverless-2016-10-31"
Resources:
  WhatIsMyIP:
    Type: "AWS::Serverless::Function"
    Properties:
      Handler: index.handler
      Runtime: python3.7
      InlineCode: |
        import json
        def handler(event, context):
          body = {'ip': event['requestContext']['identity']['sourceIp']}
          return {'statusCode': '200', 'body': json.dumps(body)}
      Events:
        Web:
          Type: Api
          Properties:
            Path: /
            Method: get
```

# aws cloudformation create-stack

```
--stack-name whatismyip --template-body file://whatismyip.yml
    --capabilities CAPABILITY_AUTO_EXPAND CAPABILITY_IAM
```


CREATE_COMPLETE whatismyip AWS::CloudFormation::Stack 2019-04-17T18:00:26.620Z
CREATE_COMPLETE WhatIsMyIPWebPermissionProd AWS::Lambda::Permission 2019-04-17T18:00:24.594Z
CREATE_COMPLETE WhatIsMyIPWebPermissionTest AWS::Lambda::Permission 2019-04-17T18:00:24.554Z
CREATE_COMPLETE ServerlessRestApiProdStage AWS::ApiGateway::Stage 2019-04-17T18:00:19.920Z
CREATE_COMPLETE ServerlessRestApiDeployment057117bca5 AWS::ApiGateway::Deployment 2019-04-17T18:00:14.711Z
CREATE_COMPLETE ServerlessRestApi AWS::ApiGateway::RestApi 2019-04-17T18:00:10.496Z
CREATE_COMPLETE WhatIsMyIP AWS::Lambda::Function 2019-04-17T18:00:04.202Z
CREATE_COMPLETE WhatIsMyIPRole AWS::IAM::Role 2019-04-17T17:59:57.981Z

```
                "StackDriftStatus": "NOT_CHECKED"
            },
            "Tags": [],
            "Outputs": [
                {
                    "OutputKey": "URL",
                    "OutputValue": "https://n9zlep7bs6.execute-api.us-east-1.amazonaws.com/Prod/"
                }
            ],
            "EnableTerminationProtection": false,
            "CreationTime": "2019-04-17T17:59:35.900Z",
            "Capabilities": [
                "CAPABILITY_AUTO_EXPAND",
                "CAPABILITY_IAM"
            ],
            "StackName": "whatismyip",
            "NotificationARNs": [],
            "StackStatus": "CREATE_COMPLETE",
            "DisableRollback": false,
            "RollbackConfiguration": {}
        }
    ]
}
/demo/cloudformation# curl https://n9zlep7bs6.execute-api.us-east-1.amazonaws.com/Prod/
{"ip": "76.191.45.114"}/demo/cloudformation#
```

# Get Started

↳ docs.aws.amazon.com/AWSCloudFormation/latest/UserGuide/gettingstarted.templatebasics.html

↳ aws.amazon.com/serverless/sam

# Thanks!


bioteam.net