AWS CDK with Python

Using Python to deploy Infrastructure-as-code

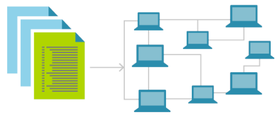
Dean Stringer dean@jumpflex.co.nz


AWS CDK & Python

- Why Infrastructure as Code (IaC)?
- Why AWS CDK?
- Why Python?
- Environment / setup
- Python app/stack demo
- Gotchas and lessons learned

Not covering…

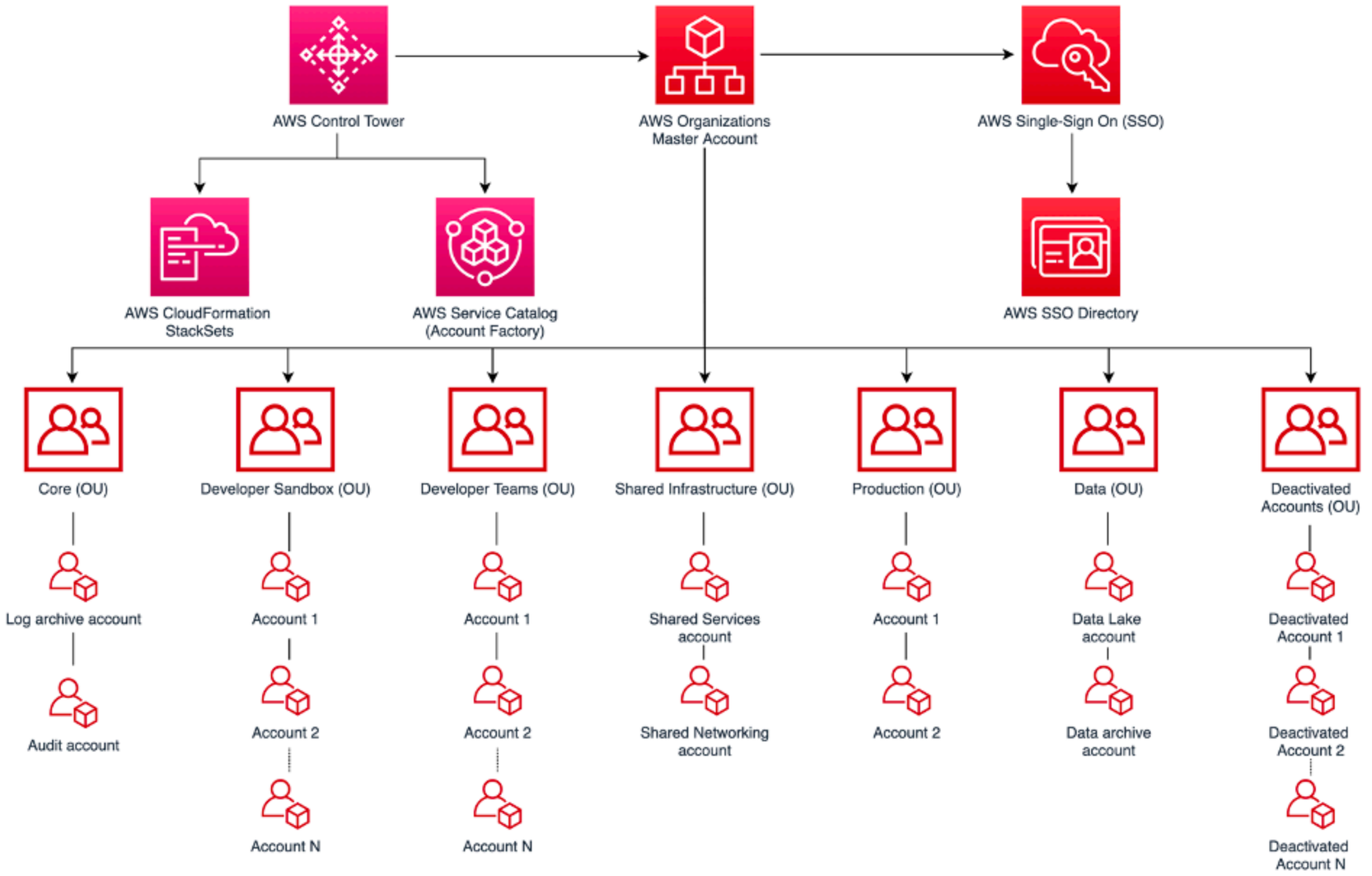
- Detail on AWS components
- Not live coding
- Not debating multi-Cloud or AWS vs xyz
- Install/use of Python, virtualenv or Node.js
- Detailed environment setup



Why IaC?

- Simplify details of deployments and config
- Automate deployments driven by code abstractions
- Reproducible environments at scale
- Reuse properties and iterate over common resource objects
- Drive architectures using data re workload perf
- Deploy test systems at prod workload scale
- Create custom constructs to suit your workloads
- Use a language you prefer in a way familiar to Dev-ops/Developers
- Code is source of truth : rendered template is simply an artefact

e.g. Multi-Account Strategy



AWS CDK

About the CDK

- Runtime written in TS and used in CLI via Node
- Develop your app/stack in language of choice: JS, TS, Python, Java, C#/.Net
- Any OS that supports Python and Node.js
- JSii used to provide language bindings for the other languages
- 'Construct' modules as abstractions for most cloud asset types
- Create 'Stacks' using constructs or extending them
- CLI generates Cfn templates and deploys Stack sets
- Open source. Many end user contributions.

CDK vs...

- Terraform and Serverless have numerous provider targets
- but.. CDK v2 plugin supports some other providers
- Cfn has complete coverage of AWS feature surface
- but … CDK constructs typically higher level (L2/L3) (although can use L1 Cfn constructs)
- Terraform uses json like static Config
- CDK Imperative. Stateful. Directly maps to output (Cfn)
- where .. Server-less & Terraform more Declarative

CDK vs...


```
> pip install boto3
```

```
import boto3

for i in ec2.instances.all():
    if i.state['Name'] == 'stopped':
        i.start()
```

Gist: Create a Lamda function:

https://gist.github.com/steinwaywhw/9d64db15518099c1f26f254ee35c4217#file-main-py

Why Python?

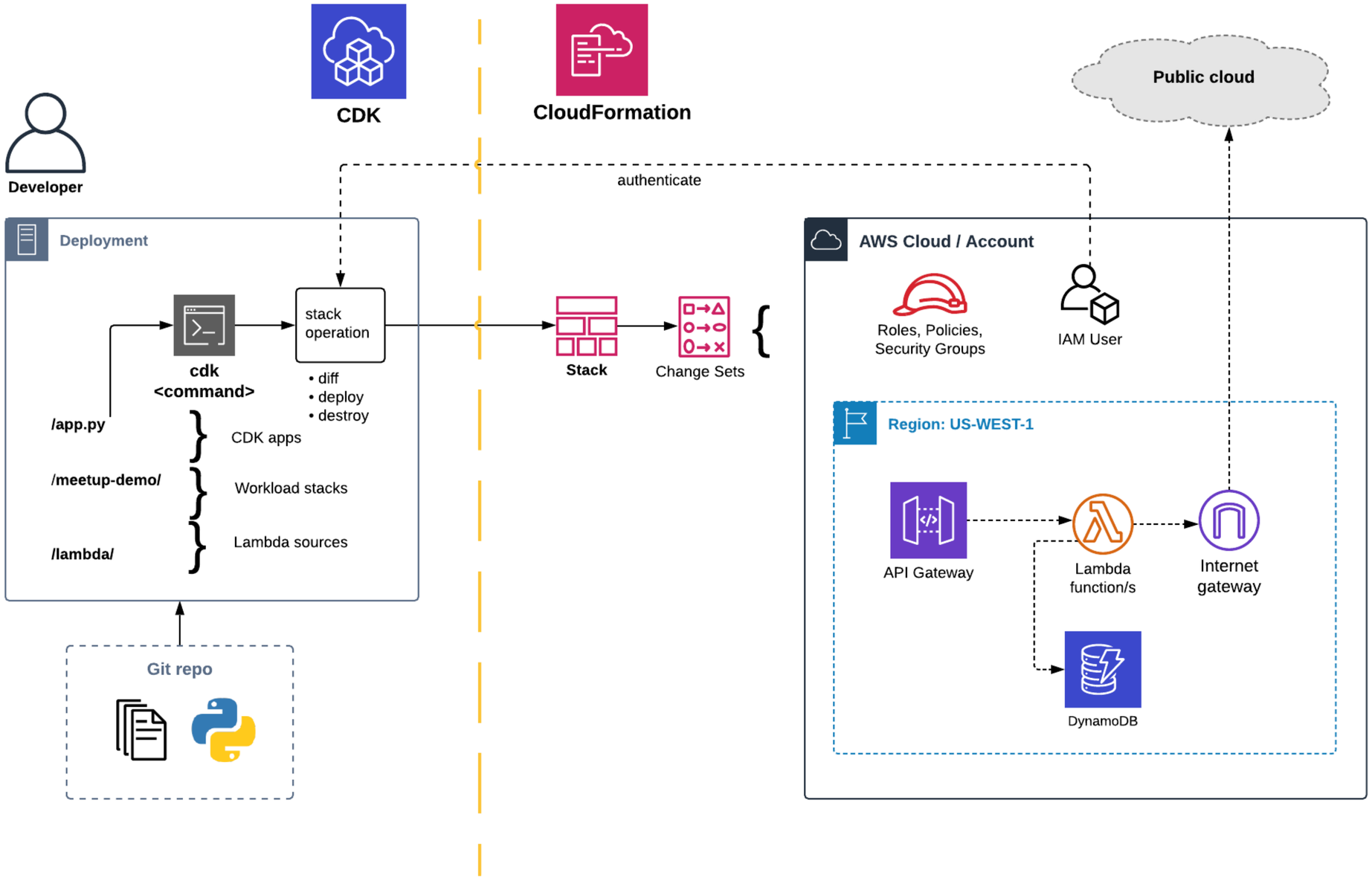
- First class support in the CDK API
- Most demos and tuts are in TS
- However, decent Python API docs page (Sphinx)
  https://docs.aws.amazon.com/cdk/api/latest/python/
- Lambda functions often written in Python using Boto3
- Because you can…

Environment / setup

- AWS CLI
- AWS Account and IAM User
- Node.js (v10.13+)
- AWS CDK Toolkit
- Git
- Python (v3.6+) & pip
- Any editor (e.g. VSCode + ms-python.python plugin)

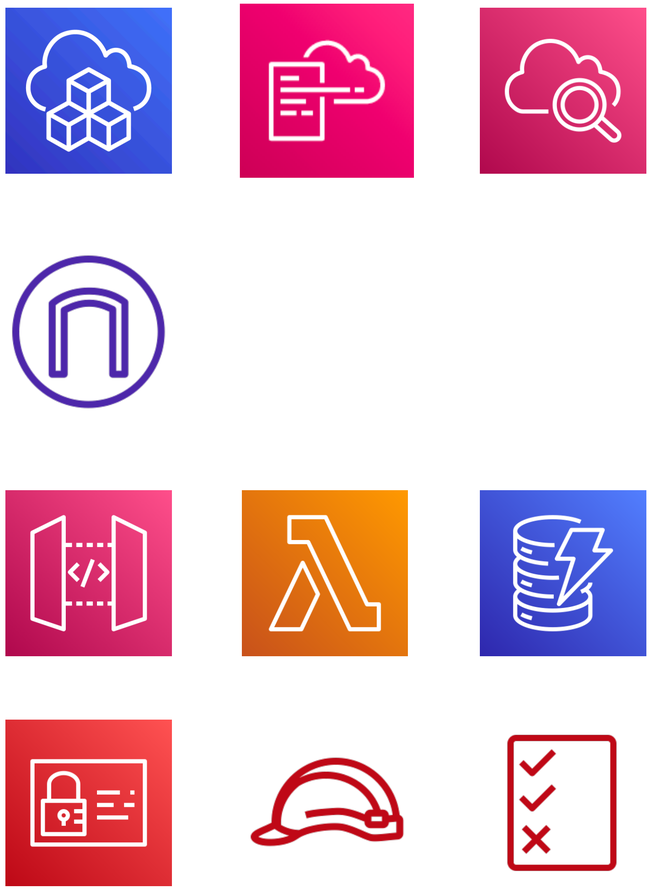
Demo Stack

Demo: Deployment and Assets



Demo: AWS Components

- CDK: Cloud Formation & Stack Sets, Cloud Watch
- VPC: Internet Gateway
- API Gateway, Lambda Functions, DynamoDB
- Identity & Access Mngt. (IAM) & Policies / Permissions



Anatomy of a CDK App

```
from aws_cdk import core as cdk
from aws_cdk import core

from meetup_demo.meetup_demo_stack import MeetupDemoStack

app = core.App()

MeetupDemoStack(app, "MeetupDemoStack",
    env=core.Environment(account='309865535433', region='us-west-1'),
    )

app.synth()
```

Anatomy of a CDK Stack

```python
from aws_cdk import (
    aws_lambda,
    core,
)

class MeetupDemoStack(core.Stack):

    def __init__(self, scope: core.Construct, construct_id: str, **kwargs) -> None:
        super().__init__(scope, construct_id, **kwargs)

        # Create a Lambda from local source
        lambda_func = aws_lambda.Function(
            self, "MeetupLambda",
            code=aws_lambda.Code.asset('lambda'),
            handler="lambda-handler.handler",
            timeout=core.Duration.seconds(300),
            runtime=aws_lambda.Runtime.PYTHON_3_7,
        )
```

Development life-cycle

Initial setup…

```
npm install -g cdk
aws configure
cdk init app ––language python
cdk bootstrap
```

Code your stack/s in Python, then …

```
pip install -r requirements.txt
cdk ls (or diff)
cdk deploy
```

Repeat till done, tear-down if required …

```
cdk destroy
```

Walkthrough…

- Lambda only
- Lambda + API
- Lambda + API + DynamoDB

CDK Gotchas

- Handling multiple apps w shared args/resources
- Triple check your AWS and .env crews before deploy/destroy calls
- If it LOOKS like a lot of new assets are being created you may be on the wrong account
- Double-check any permissions/role changes shown during deploy
- If a Stack fails don’t panic - rare that they don’t recovers
- Need SOME sort of bundling (unless only using @aws-cdk libs)
- Be sure selected correct right region in UI
- Check versions when adding new @aws-cdk libs to requirements.txt

Err .. more Gotchas

- If CLI call connection times out will continue in background
- Deadlocks can occur if you remove a referenced element
- DONT delete /cdk.out/
- Installing CDK locally in the venv (version drift)
- Although open-source outputs are Cfn specific / proprietary
- Construct libs are all in one big mono-repo

Resources

- Installing, updating, and uninstalling the AWS CLI v2
  https://docs.aws.amazon.com/cli/latest/userguide/install-cliv2.html
- Python CDK API docs
  https://docs.aws.amazon.com/cdk/api/latest/python/
- aws-cdk examples (including Python)
  https://github.com/aws-samples/aws-cdk-examples/tree/master/python
- CDK Python workshop (from AWS
  https://cdkworkshop.com/30-python.html
- JSii : Framework for building language bindings between JS and others
  https://github.com/aws/jsii