



DEV07

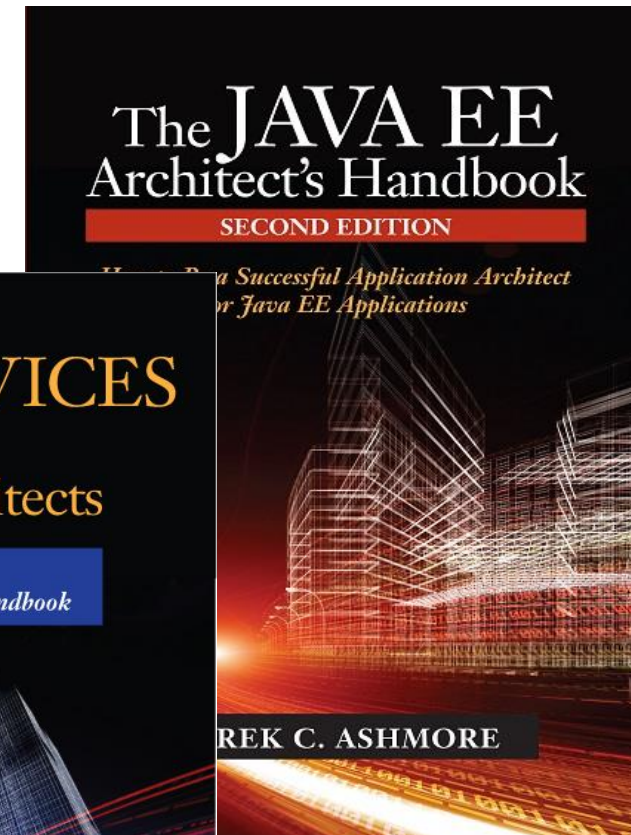
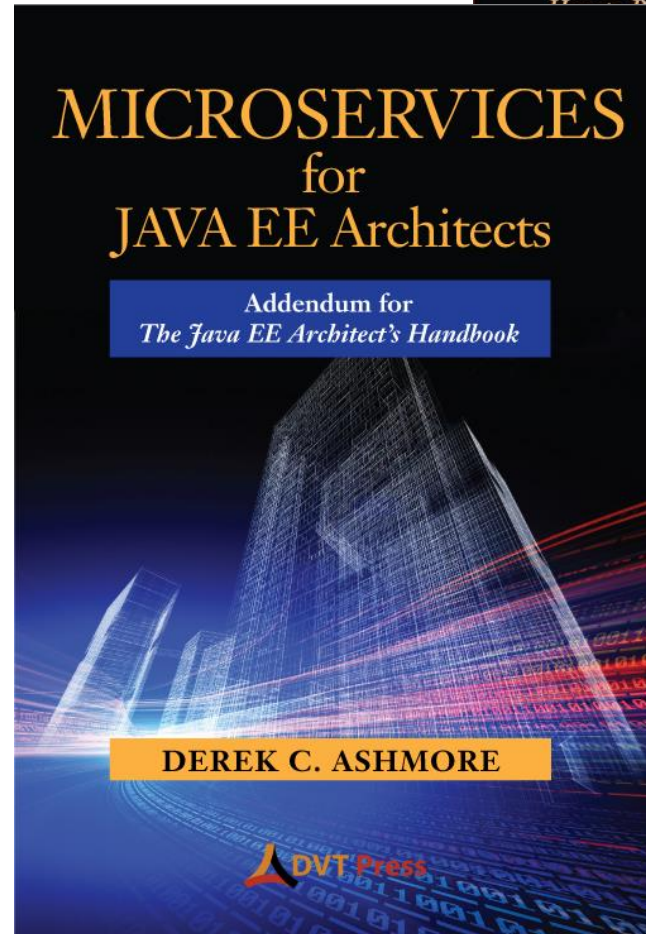
AWS Lambda: Best Practices and Common Mistakes

Derek C. Ashmore

Principle Cloud Architect, Asperitas Consulting

Who am I?

- Professional Geek since 1987
- Java/J2EE/Java EE since 1999
- AWS since 2010
- Specialties
 - Cloud Workshops
 - Cloud-native Applications
- Yes – I still code!



Discussion Resources

- This slide deck
 - <https://www.slideshare.net/derekashmore/presentations>
- Sample code on my Github
 - <https://github.com/Derek-Ashmore/>
- Slide deck has hyper-links!
 - Don't bother writing down URLs

What Makes a “Best Practice”?

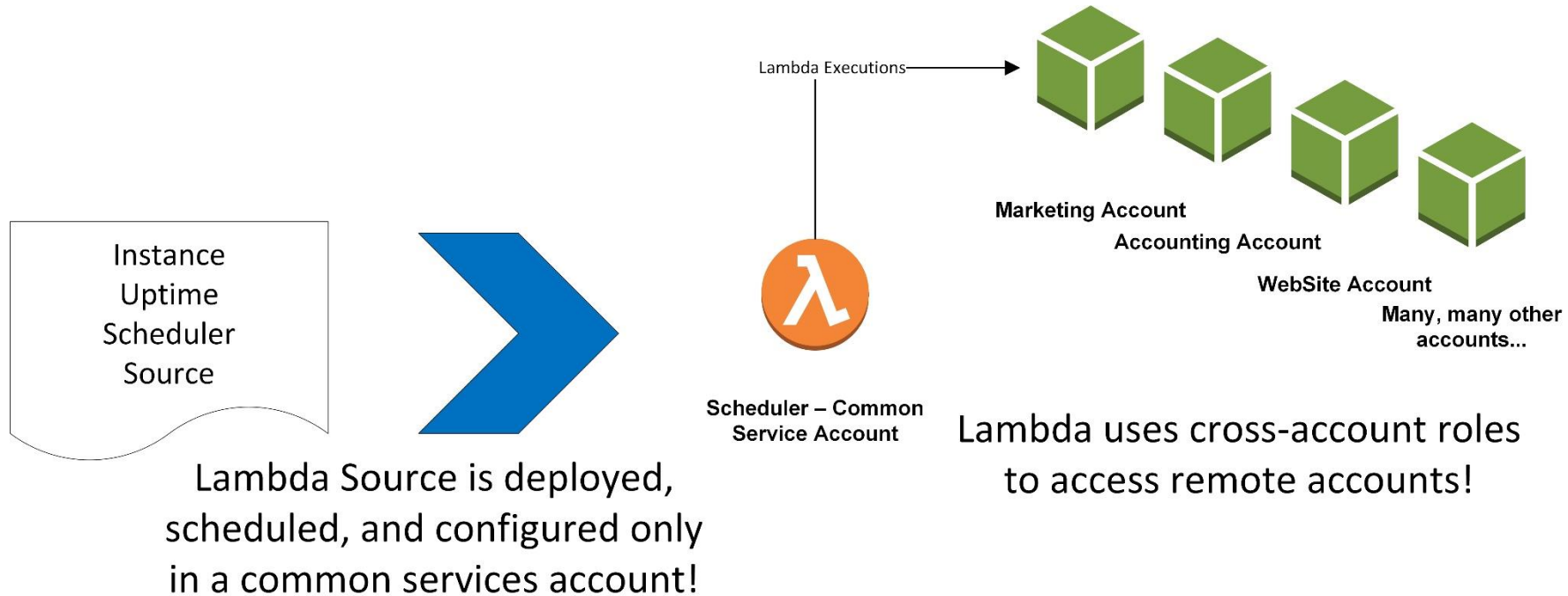
- Makes Support Easier
 - Cuts time to diagnose bugs
 - Makes it easier for others to fix/enhance what you wrote
- Increases Reuse
 - Don’t Repeat Yourself (DRY)
 - Easier to deploy fixes and enhancements
 - Can be used in multiple contexts

Lambda Installs Everywhere!



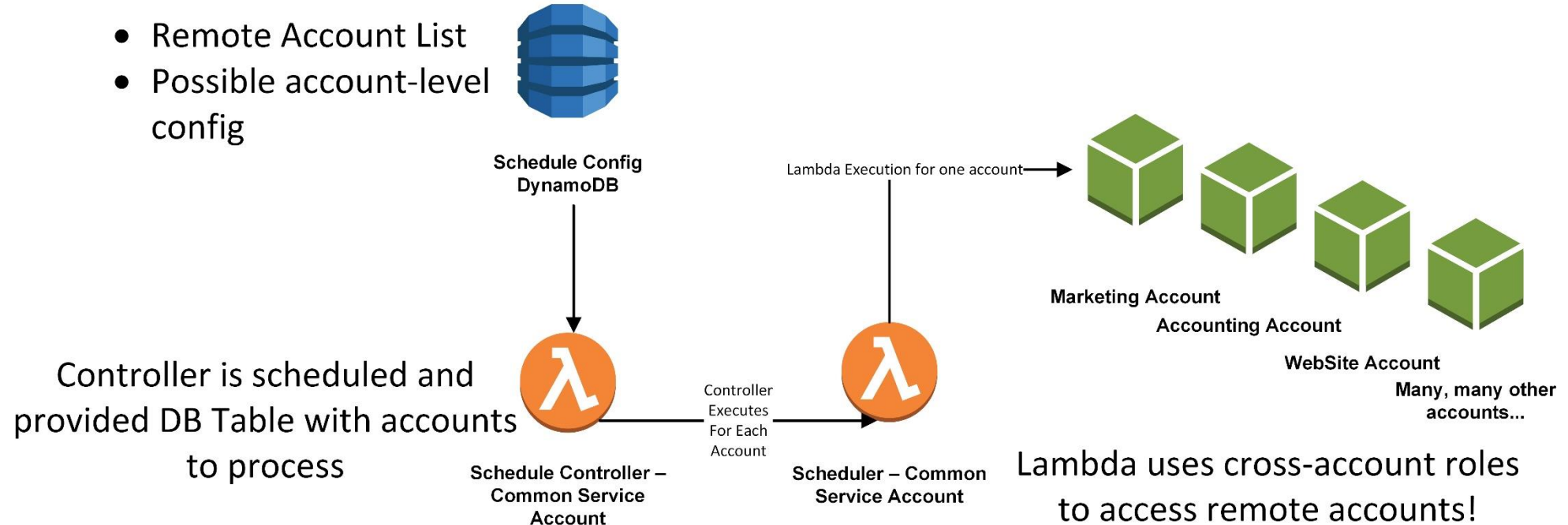
- Changes / Bug Fixes need to be deployed everywhere
- Solving with automation solves the wrong problem!

One Lambda Install for All!



- Lambda centrally defined
- Bugfixes in one place
- Sample in Python [here](#)

For workloads over 5 min



- Executor that invokes lambda asynchronously for each account
- Sample in Python [here](#)

Which Error would you rather fix?

Choice 1

ClientError: An error occurred (AccessDenied) when calling the AssumeRole operation: Not authorized to perform sts:AssumeRole

Choice 2

```
ClientError: (u'An error occurred (AccessDenied) when calling the AssumeRole operation: Not authorized to perform sts:AssumeRole',
{
    u 'Assumed_Role': u 'arn:aws:iam::1234:role/Derek2Adminsitrator'
}, {
    'aws_request_id': '5071870f-b390-11e8-9aa1-5f12fffb48c7',
    'log_stream_name': '2018/09/08/[$LATEST]00284953fdae4f2dab8d25a7b7f69e56',
    'invoked_function_arn': 'arn:aws:lambda:us-east-1:517214143524:function:crossAccountListRoles',
    'client_context': None,
    'log_group_name': '/aws/lambda/crossAccountListRoles',
    'function_name': 'crossAccountListRoles',
    'function_version': '$LATEST',
```


Report Inputs on Exception

```
def crossAccountHandler(event, context):  
    try:  
        if 'Assumed_Role' in event:  
            listRolesInDifferentAccount(event['Assumed_Role'])  
        else:  
            raise Exception('Assumed_Role not provided as argument')  
    except Exception as e:  
        e.args += (event, vars(context))  
        raise
```

- Place a Try / Catch in your handler
 - [Python](#) Example
 - [Java](#) Example

Check Arguments Up Front

```
def crossAccountHandler(event, context):  
    try:  
        if 'Assumed_Role' in event:  
            listRolesInDifferentAccount(event['Assumed_Role'])  
        else:  
            raise Exception('Assumed_Role not provided as argument')  
    except Exception as e:  
        e.args += (event, vars(context))  
        raise
```

- Check your arguments with a clear error message
 - [Python](#) Example
 - [Java](#) Example

Separate Lambda from Business Logic

```
def crossAccountHandler(event, context):  
    try:  
        if 'Assumed_Role' in event:  
            listRolesInDifferentAccount(event['Assumed_Role'])  
        else:  
            raise Exception('Assumed_Role not provided as argument')  
    except Exception as e:  
        e.args += (event, vars(context))  
        raise
```

- Make business logic reusable
 - Other applications
 - On-premise
- Easier to locally develop and debug
 - Lambda is thin!

Use Configuration Injection

- No environment specifics hardcoded in the Lambda deployment
 - Not in the source
 - Not in the zip bundle
- Use Environment Variables on the Lambda Definition
 - No un-encrypted secrets (e.g. database password)
 - Use KMS to encrypt/decrypt
 - Digital vault (Hashicorp Vault, CyberArk, etc.)
- Use Arguments in the triggering event
 - No un-encrypted secrets
- Anti-Example
 - Splunk forwarding Lambda with hard-coded Splunk channels

Specify Lambda Source Repo

- Explicitly put the source repository name in the Lambda comments
 - In most organizations, the repository name isn't obvious
 - Others changing your code need it
 - You don't want source control to be out of date

Further Reading

- This slide deck
 - <https://www.slideshare.net/derekashmore/presentations>
- AWS Lambda Reading List
 - <http://www.derekashmore.com/2016/04/aws-lambda-reading-list.html>
- Amazon's Published Best Practice List
 - <https://docs.aws.amazon.com/lambda/latest/dg/best-practices.html>

Questions?

- Derek Ashmore:
 - Blog: www.derekashmore.com
 - LinkedIn: www.linkedin.com/in/derekashmore
 - Connect Invites from attendees welcome
 - Twitter: https://twitter.com/Derek_Ashmore
 - GitHub: <https://github.com/Derek-Ashmore>
 - Book: <http://dvtpress.com/>

***Please complete the Session
Survey in the Summit Mobile App***

