/ OSEDEA

Philippe Trépanier

Adaptive Serverless in Python with AWS Chalice

/OSEDEA

Montreal-based innovation and technology firm, founded in 2011

We design intelligent, high-performance applications, tailored to the specific needs of our customers

Our services

/ Software Engineering
/ UX/UI Design
/ Artificial Intelligence
/ Robotics

Technology, for us, is a means within our innovation process, not the ultimate goal

Today's talk is about...

- ... iterating quickly
- ... keeping an eye on medium-term problems
- ... pushing long-term problems far far away
- ... building an adaptive serverless application with Chalice

High-Level Todos

1. Context, Services, Technologies and Limitations
2. Setting up a base infrastructure
3. Building an Hello World app
4. Integrating app & infrastructure
5. Upgrading to a Polls app
6. Running multiple environments

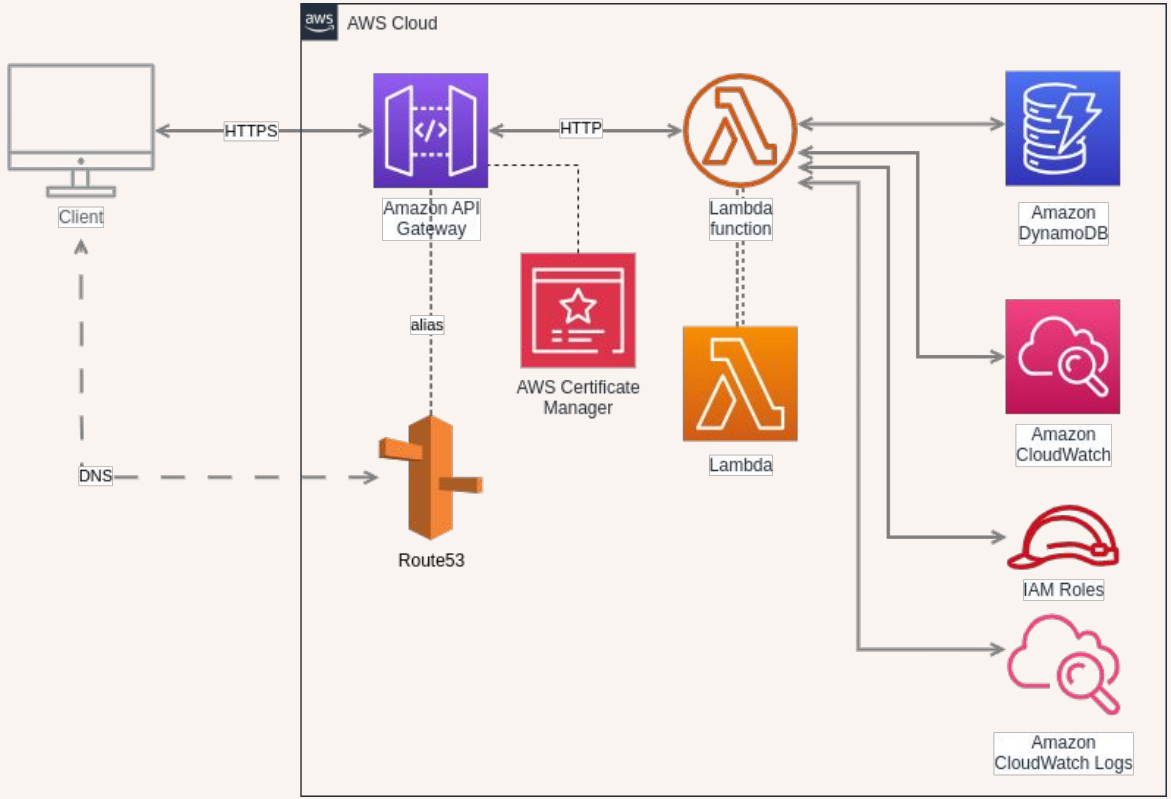
Adaptive Architecture

An adaptive system changes its behavior, structure or resources on demand.

Python Stuff

- Python 3.11: latest version compatible with Chalice
- Poetry: dependency management, environment management
- Chalice: AWS-built framework to simplify the building of Lambdas with Python
- Boto3: AWS SDK for Python, programmatically manage AWS services

AWS Services



🚀 Lambda's Stuff

- CPU Compute
- You're not managing the server
- You pay for what you use, scale-to-zero by default
- Hyperscaling-ish! *Almost* don't need to manage load
- Close relationship between infrastructure and software
- Function-as-a-Service
- All the code on one lambda □ mono-lambda!
- 🤡 CPU scales with Memory

Lambda Quotas

- Concurrent executions: 1,000 to Tens of thousands
- Storage for uploaded functions: 75 GB to Terabytes
- Function memory allocation: 128 MB to 10,240 MB **HARD**
- Function timeout: 15 minutes **HARD**
- Function environment variables: 4KB **HARD**
- Function concurrency scaling: 1,000/10 seconds/function **HARD**
- Maximum payload (request & response): 6MB **HARD**

AWS Lambda Pricing

At Default Execution Quota
Memory == 128
Execution time == 100ms
Concurrent users == 1000
Executions per user == 10

10,000 RPS Total
0.00000041USD/execution
0.0041USD/second
~129K USD/year for ~315B execs

~ 400 USD per Billion execs
~ 0.4 USD per Million execs

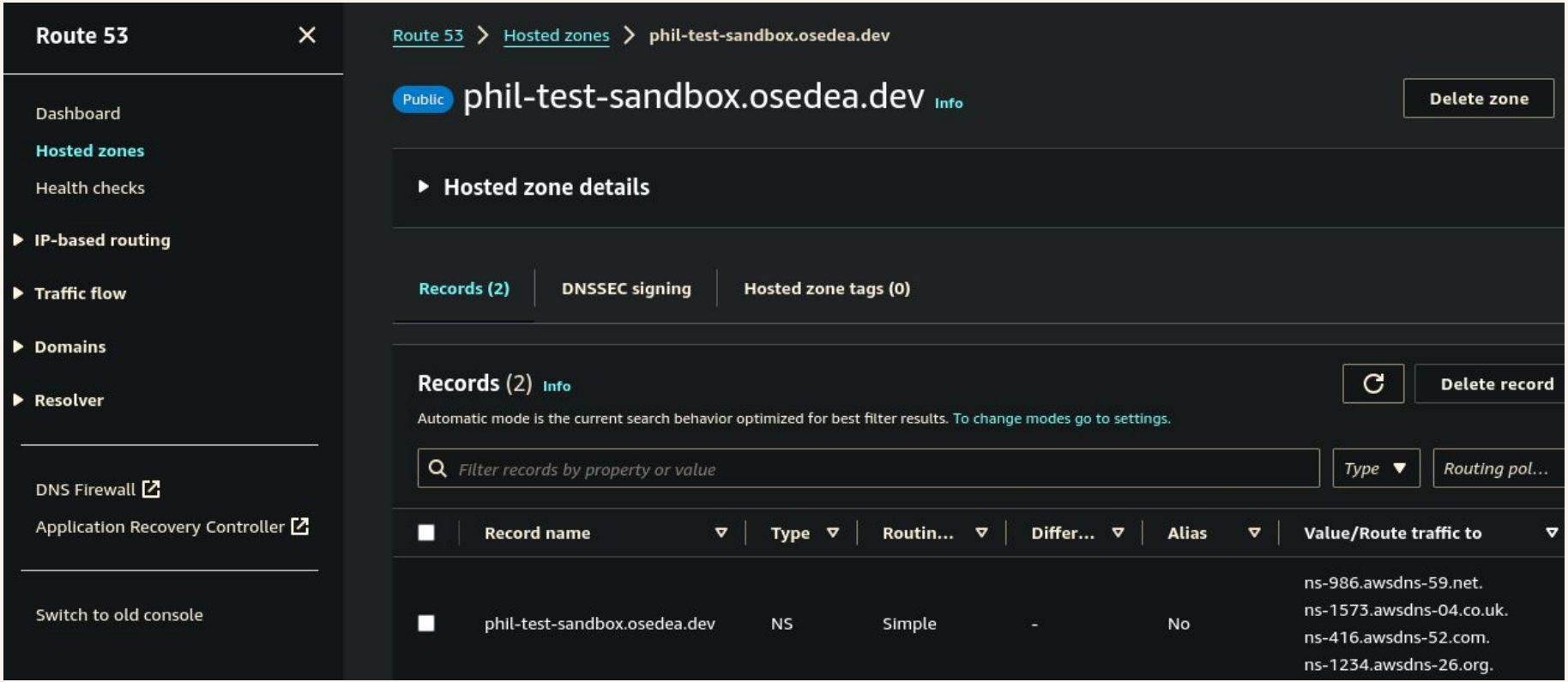
First 6 Billion GB-seconds / month	$0.0000166667 for every GB-second	$0.20 per 1M requests
Next 9 Billion GB-seconds / month	$0.000015 for every GB-second	$0.20 per 1M requests
Over 15 Billion GB-seconds / month	$0.0000133334 for every GB-second	$0.20 per 1M requests

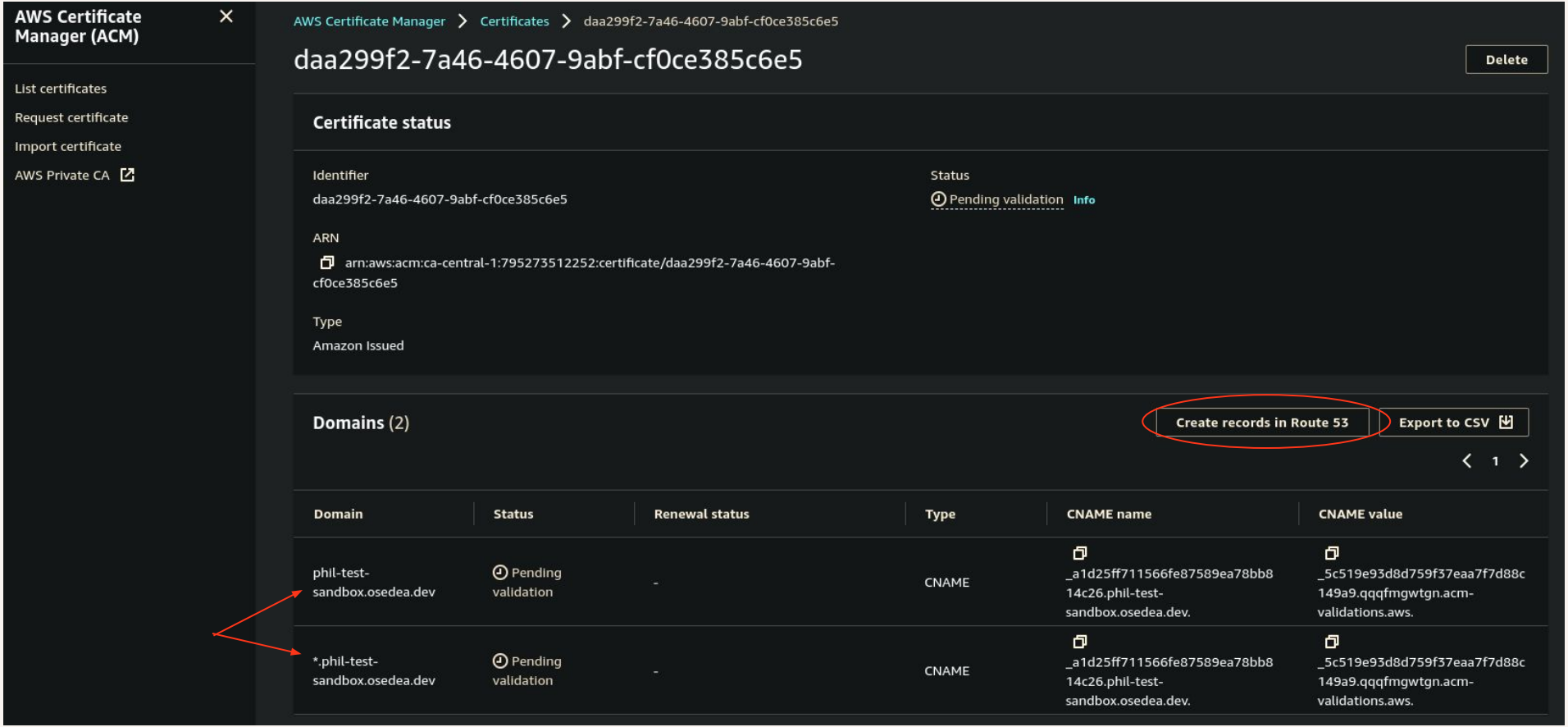
Memory (MB)	Price per 1ms
128	$0.0000000021
512	$0.0000000083
1024	$0.0000000167
1536	$0.0000000250
2048	$0.0000000333
3072	$0.0000000500
4096	$0.0000000667

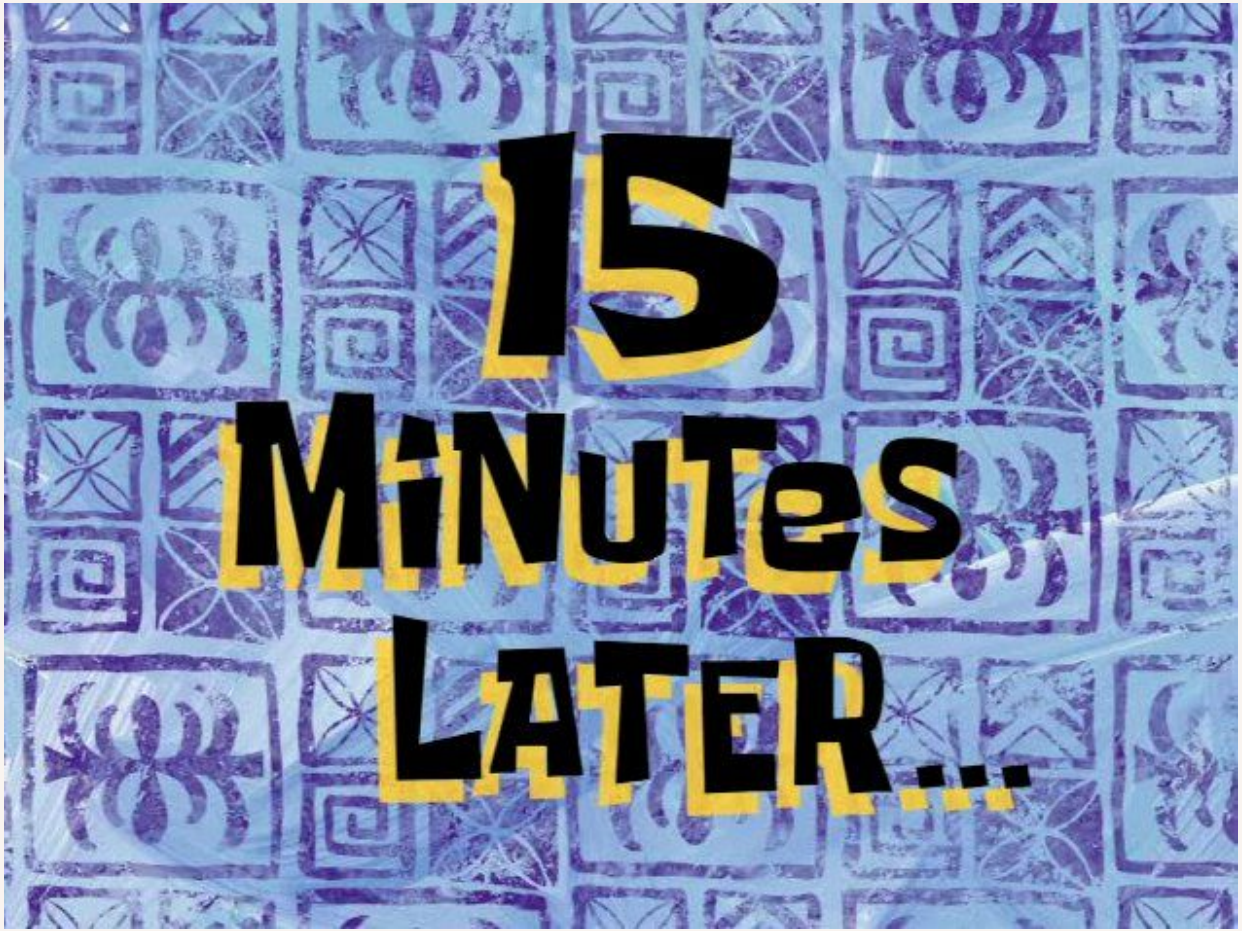
Setting Up A Base Infra

Base Infrastructure Todos

1. Add Domain to Route53
2. Generate SSL Certificate
3. Create a CLI User in IAM
4. Install and configure AWS CLI


Route 53
Dashboard
Hosted zones
Health checks
IP-based routing
Traffic flow
Domains
Resolver
DNS Firewall
Application Recovery Controller
Switch to old console
Route 53 > Hosted zones > phil-test-sandbox.osedea.dev
Public
phil-test-sandbox.osedea.dev Info
Delete zone
Hosted zone details
Records (2)
DNSSEC signing
Hosted zone tags (0)
Records (2) Info
Delete record
Automatic mode is the current search behavior optimized for best filter results. To change modes go to settings.
Filter records by property or value
Type
Routing pol...
Record name
Type
Routin...
Differ...
Alias
Value/Route traffic to
phil-test-sandbox.osedea.dev
NS
Simple
-
No
ns-986.awsdns-59.net.
ns-1573.awsdns-04.co.uk.
ns-416.awsdns-52.com.
ns-1234.awsdns-26.org.


AWS Certificate Manager (ACM)
List certificates
Request certificate
Import certificate
AWS Private CA
AWS Certificate Manager > Certificates > daa299f2-7a46-4607-9abf-cf0ce385c6e5
daa299f2-7a46-4607-9abf-cf0ce385c6e5
Delete
Certificate status
Identifier
daa299f2-7a46-4607-9abf-cf0ce385c6e5
Status
Pending validation Info
ARN
arn:aws:acm:ca-central-1:795273512252:certificate/daa299f2-7a46-4607-9abf-cf0ce385c6e5
Type
Amazon Issued
Domains (2)
Create records in Route 53
Export to CSV
1
Domain
Status
Renewal status
Type
CNAME name
CNAME value
phil-test-sandbox.osedea.dev
Pending validation
-
CNAME
_a1d25ff711566fe87589ea78bb814c26.phil-test-sandbox.osedea.dev.
_5c519e93d8d759f37eaa7f7d88c149a9.qqqfmgwtgn.acm-validations.aws.
*.phil-test-sandbox.osedea.dev
Pending validation
-
CNAME
_a1d25ff711566fe87589ea78bb814c26.phil-test-sandbox.osedea.dev.
_5c519e93d8d759f37eaa7f7d88c149a9.qqqfmgwtgn.acm-validations.aws.


15
MINUTES
LATER...

Cool 👍


phil-test-
sandbox.osedea.dev
Success
*.phil-test-
sandbox.osedea.dev
Success

chalice-deploy Info

Delete

Summary

ARN
arn:aws:iam::795273512252:user/chalice-deploy

Console access
Disabled

Access key 1
AKIA3SKP6YU6KW2CB3JV - Active
Used today. Created today.

Created
February 20, 2024, 14:02 (UTC-05:00)

Last console sign-in
-

Access key 2
Create access key

Permissions | Groups | Tags | Security credentials | Access Advisor

Permissions policies (5)

Permissions are defined by policies attached to the user directly or through groups.

Remove | Add permissions

Search

Filter by Type: All types

1

Policy name	Type	Attached via
AmazonAPIGatewayAdministrator	AWS managed	Directly
AmazonDynamoDBFullAccess	AWS managed	Directly
AmazonRoute53FullAccess	AWS managed	Directly
AWSLambda_FullAccess	AWS managed	Directly
IAMFullAccess	AWS managed	Directly

Setup and Configure AWS CLI v2

```
➜  ~ curl "https://awscli.amazonaws.com/awscli-exe-linux-x86_64.zip" -o "awscliv2.zip"
unzip awscliv2.zip
sudo ./aws/install
```

```
➜  chalice-project git:(main) ✗ aws configure
AWS Access Key ID [None]: [redacted]
AWS Secret Access Key [None]: [redacted]
Default region name [None]: ca-central-1
Default output format [None]:
```

```
➜  ~ aws route53 list-resource-record-sets --hosted-zone-id Z01822541IR2XNWD4KDBG | cat
{
    "ResourceRecordSets": [
        {
            "Name": "phil-test-sandbox.osedea.dev.",
            "Type": "NS",
            "TTL": 172800,
            "ResourceRecords": [
                {
                    "Value": "ns-986.awsdns-59.net."
                },
                {
                    "Value": "ns-1573.awsdns-04.co.uk."
                },
                {
                    "Value": "ns-416.awsdns-52.com."
```

Building An Hello World App

Project Setup Todos

1. Setup Python & Poetry
2. Init a Git repo
3. Create a Hello World Chalice project
4. Run our Hello World project locally
5. Deploy and run our Hello World project on AWS

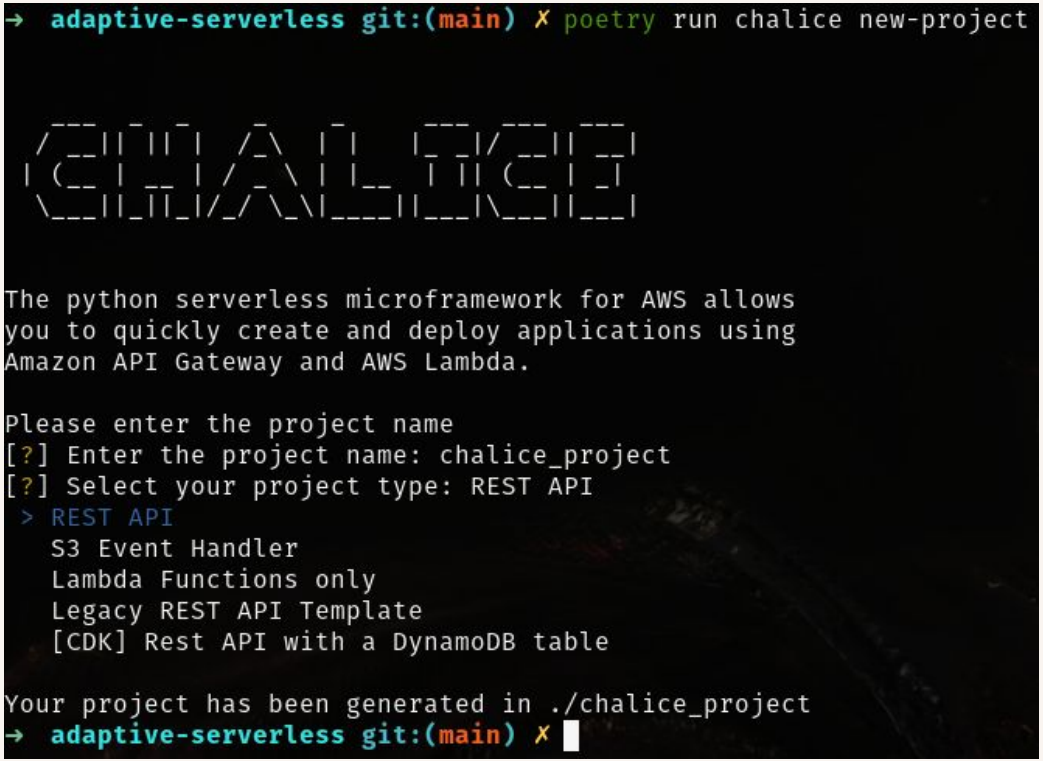
```
➜  adaptive-serverless which python3.11
/home/phil/.local/bin/python3.11
```

```
➜  adaptive-serverless python3.11 -m pip install -q poetry
```

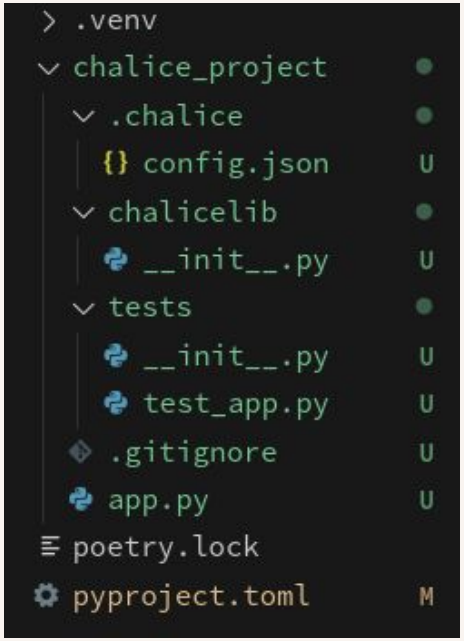
```
➜  adaptive-serverless python3.11 -m poetry init

This command will guide you through creating your pyproject.toml config.

Package name [adaptive-serverless]:
```

```
➜  adaptive-serverless poetry add -q --group dev chalice
➜  adaptive-serverless poetry add -q boto3
```

```
➜  adaptive-serverless git init .
Initialized empty Git repository in /home/phil/Projects/adaptive-serverless/.git/
➜  adaptive-serverless git:(main) ✗ git remote add origin git@github.com:philtrep/confoo2024
➜  adaptive-serverless git:(main) ✗ git add -A
➜  adaptive-serverless git:(main) ✗ git commit -m "Python Setup"
gi[main (root-commit) 785d183] Python Setup
 2 files changed, 380 insertions(+)
 create mode 100644 poetry.lock
 create mode 100644 pyproject.toml
➜  adaptive-serverless git:(main) git push -u origin HEAD
```


→ adaptive-serverless git:(main) ✗ poetry run chalice new-project
The python serverless microframework for AWS allows
you to quickly create and deploy applications using
Amazon API Gateway and AWS Lambda.
Please enter the project name
[?] Enter the project name: chalice_project
[?] Select your project type: REST API
> REST API
S3 Event Handler
Lambda Functions only
Legacy REST API Template
[CDK] Rest API with a DynamoDB table
Your project has been generated in ./chalice_project
→ adaptive-serverless git:(main) ✗

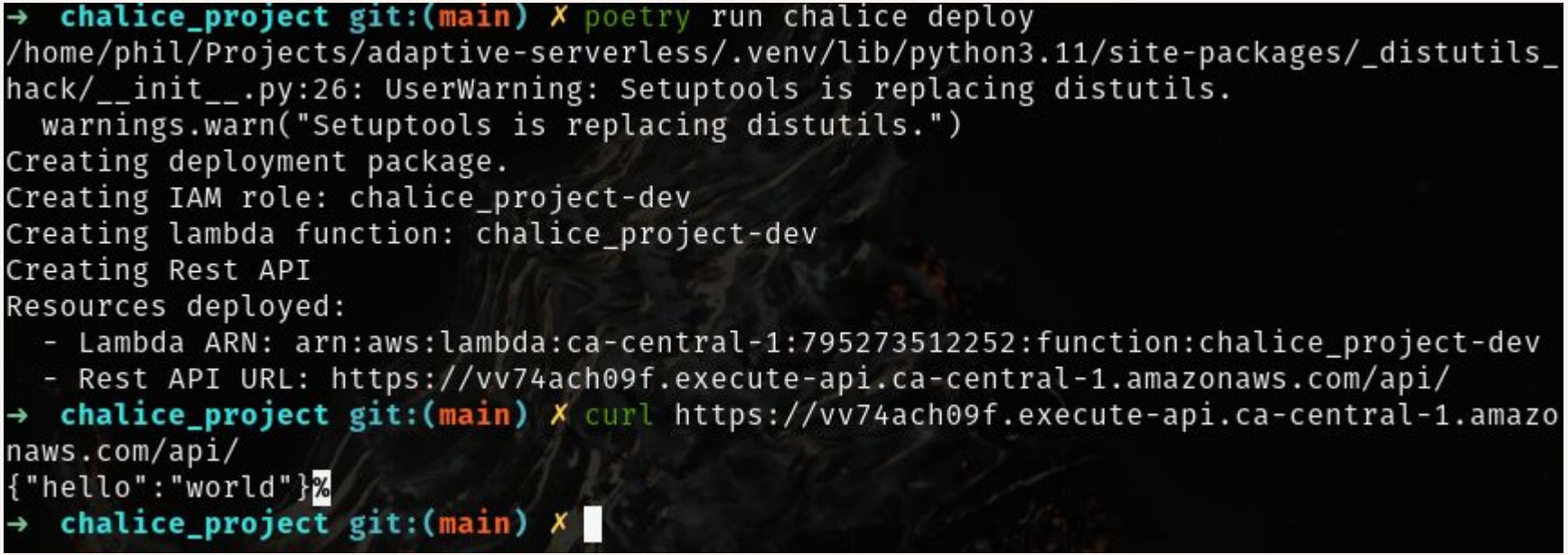

EXPLORER
ADAPTIVE-SERVE…
.venv
chalice_project
.chalice
config.json U
chalicelib
__init__.py U
tests
__init__.py U
test_app.py U
.gitignore U
app.py U
poetry.lock
pyproject.toml M

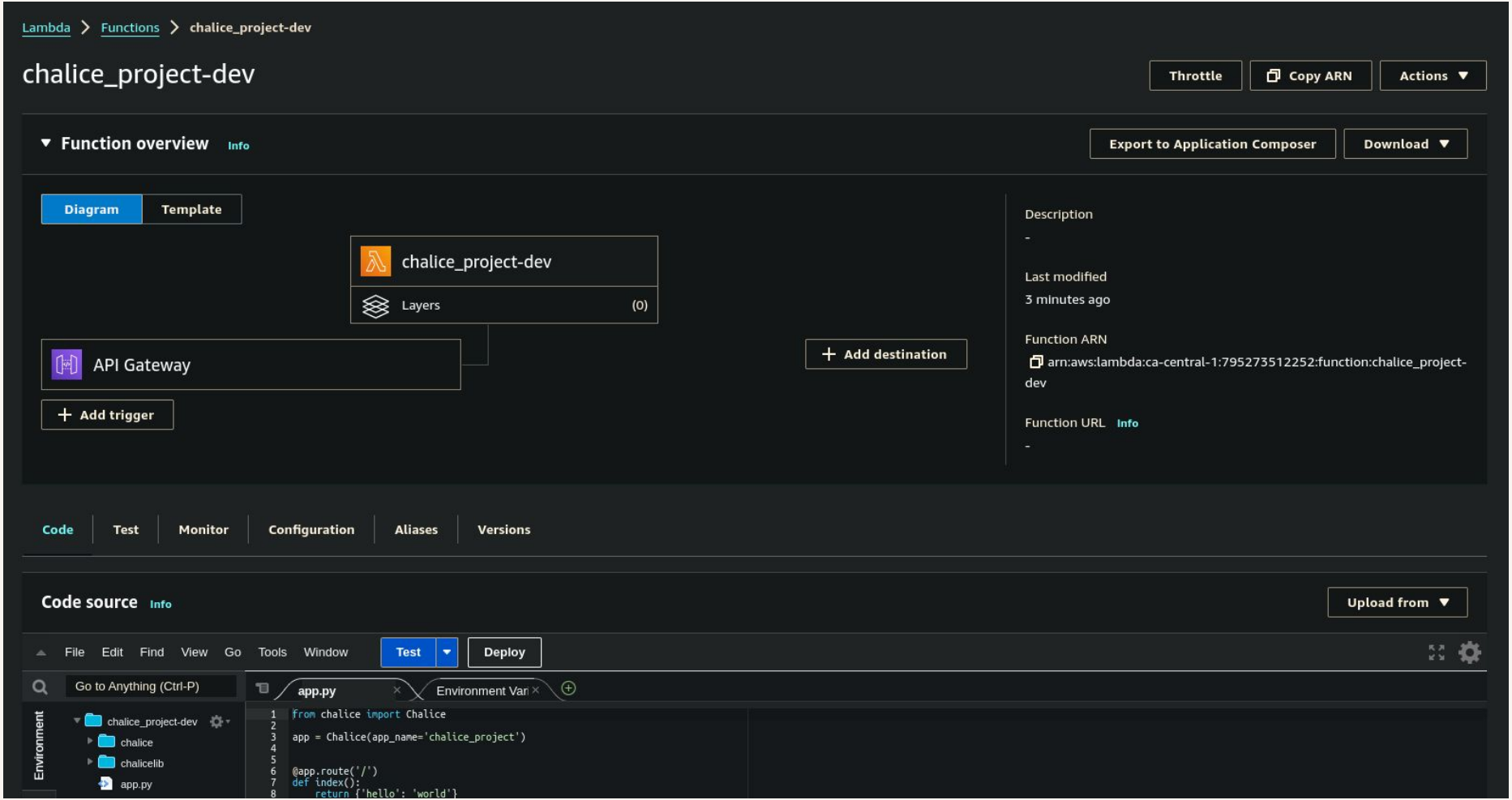



```
chalice_project > app.py > ...
1   from chalice import Chalice
2
3   app = Chalice(app_name='chalice_project')
4
5
6   @app.route('/')
7   def index():
8       return {'hello': 'world'}
9
```

```
➜  adaptive-serverless git:(main) ✗ cd chalice_project
➜  chalice_project git:(main) ✗ poetry run chalice local
Serving on http://127.0.0.1:8000
Restarting local dev server.
Serving on http://127.0.0.1:8000
127.0.0.1 - - [20/Feb/2024 18:54:15] "GET / HTTP/1.1" 200 -
```

```
➜  ~ curl http://127.0.0.1:8000/
{"hello":"world"}%
➜  ~
```


➜ chalice_project git:(main) ✗ poetry run chalice deploy
/home/phil/Projects/adaptive-serverless/.venv/lib/python3.11/site-packages/_distutils_
hack/__init__.py:26: UserWarning: Setuptools is replacing distutils.
warnings.warn("Setuptools is replacing distutils.")
Creating deployment package.
Creating IAM role: chalice_project-dev
Creating lambda function: chalice_project-dev
Creating Rest API
Resources deployed:
- Lambda ARN: arn:aws:lambda:ca-central-1:795273512252:function:chalice_project-dev
- Rest API URL: https://vv74ach09f.execute-api.ca-central-1.amazonaws.com/api/
➜ chalice_project git:(main) ✗ curl https://vv74ach09f.execute-api.ca-central-1.amazo
naws.com/api/
{"hello":"world"}%
➜ chalice_project git:(main) ✗


Lambda > Functions > chalice_project-dev
chalice_project-dev
Throttle
Copy ARN
Actions
Function overview Info
Export to Application Composer
Download
Diagram
Template
chalice_project-dev
Layers (0)
API Gateway
Add trigger
Add destination
Description
-
Last modified
3 minutes ago
Function ARN
arn:aws:lambda:ca-central-1:795273512252:function:chalice_project-dev
Function URL Info
-
Code
Test
Monitor
Configuration
Aliases
Versions
Code source Info
Upload from
File Edit Find View Go Tools Window
Test
Deploy
Go to Anything (Ctrl-P)
app.py
Environment Vari
Environment
chalice_project-dev
chalice
chalicelib
app.py
1 from chalice import Chalice
2
3 app = Chalice(app_name='chalice_project')
4
5
6 @app.route('/')
7 def index():
8 return {'hello': 'world'}

Recent invocations

#	Timestamp	RequestId	LogStream	DurationInMS	BilledDurationInMS	MemorySetInMB	MemoryUsedInMB
▶ 1	2024-02-20T23:56:23.196Z	dceaee25-be59-4bf1-8111-d5d2c643df2a	2024/02/20/[$LATEST]b15d8b44743047efbd49dd2c26b1823a	1.3	2.0	128.0	45.0

CloudWatch > Log groups > /aws/lambda/chalice_project-dev > 2024/02/20/[$LATEST]b15d8b44743047efbd49dd2c26b1823a

Log events

Actions ▼ | Start tailing | Create metric filter

You can use the filter bar below to search for and match terms, phrases, or values in your log events. Learn more about filter patterns

Filter events | Clear | 1m | 30m | 1h | 12h | Custom (3h) | Local timezone ▼ | Display ▼

Timestamp	Message
	No more records within selected time range *Retry*
2024-02-20T18:55:57.264-05:00	INIT_START Runtime Version: python:3.11.v27 Runtime Version ARN: arn:aws:lambda:ca-central-1::runtime:aaa0628b7c27cf9a6da9328d61875e549fca5396…
2024-02-20T18:55:57.397-05:00	START RequestId: af389565-4679-4e17-946b-36728d16a853 Version: $LATEST
2024-02-20T18:55:57.407-05:00	END RequestId: af389565-4679-4e17-946b-36728d16a853
2024-02-20T18:55:57.408-05:00	REPORT RequestId: af389565-4679-4e17-946b-36728d16a853 Duration: 10.88 ms Billed Duration: 11 ms Memory Size: 128 MB Max Memory Used: 45 MB In…
2024-02-20T18:56:23.195-05:00	START RequestId: dceaee25-be59-4bf1-8111-d5d2c643df2a Version: $LATEST
2024-02-20T18:56:23.196-05:00	END RequestId: dceaee25-be59-4bf1-8111-d5d2c643df2a
2024-02-20T18:56:23.196-05:00	REPORT RequestId: dceaee25-be59-4bf1-8111-d5d2c643df2a Duration: 1.30 ms Billed Duration: 2 ms Memory Size: 128 MB Max Memory Used: 45 MB

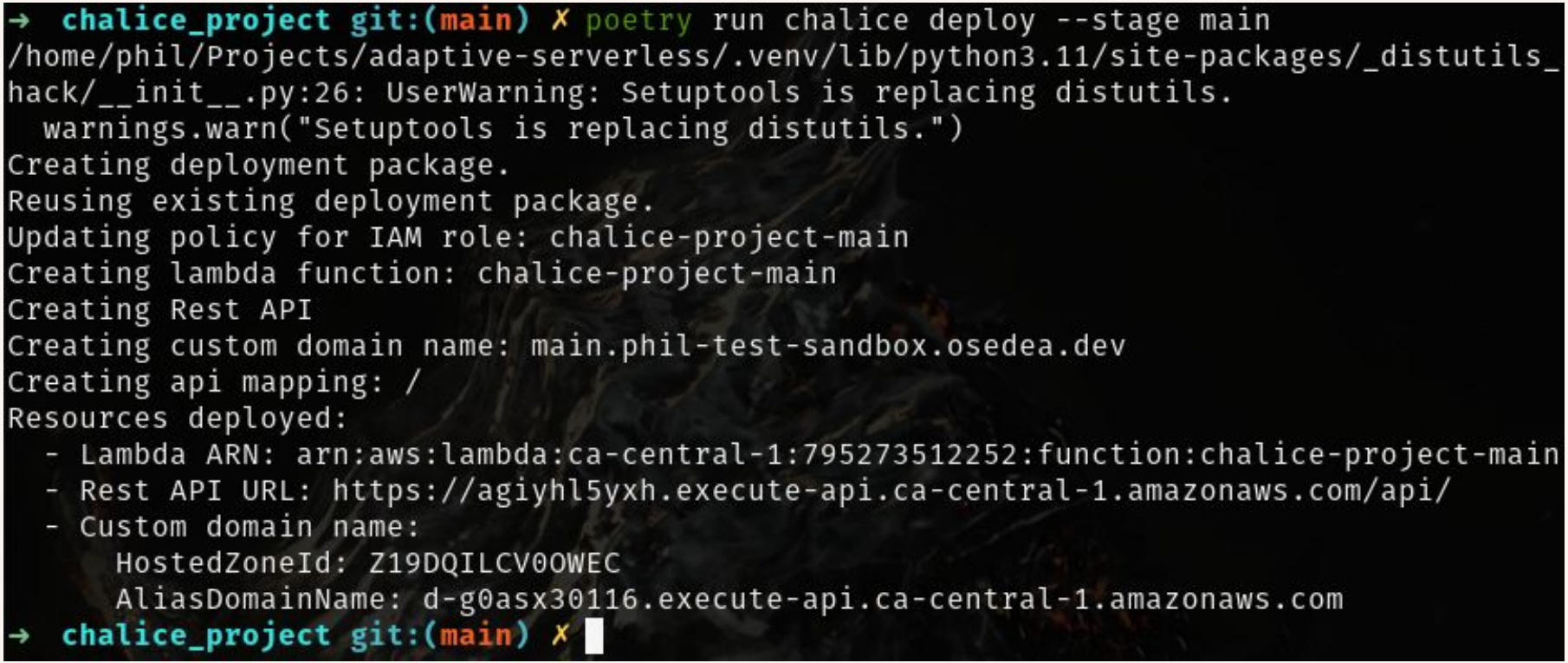
Integrating App & Infra

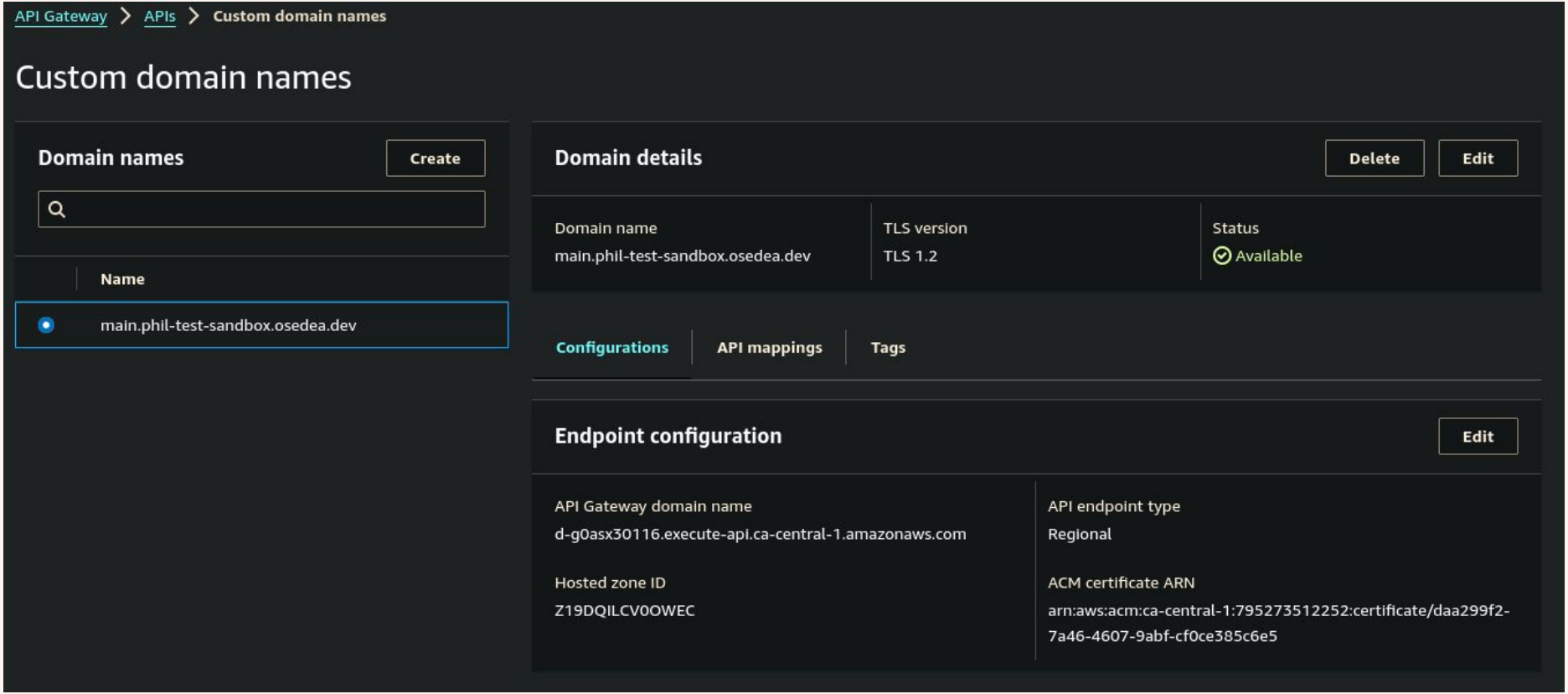
📋 Integrate Infra & Project Todos

1. Set up a custom domain name in the Chalice config
2. Deploy Chalice with the custom domain name
3. Add an A record to Route53 point to our Hello World app
4. Test that the subdomain points to the app


{} config.json U
chalice_project > .chalice > {} config.json > ...
{
"version": "2.0",
"app_name": "chalice_project",
"stages": {
"dev": {
"api_gateway_stage": "api"
}
}
}


{} config.json U
chalice_project > .chalice > {} config.json > ...
{
"version": "2.0",
"app_name": "chalice_project",
"stages": {
"main": {
"api_gateway_stage": "api",
"api_gateway_endpoint_type": "REGIONAL",
"api_gateway_custom_domain": {
"domain_name": "main.phil-test-sandbox.osedea.dev",
"tls_version": "TLS_1_2",
"certificate_arn":
"arn:aws:acm:ca-central-1:795273512252:certificate/
daa299f2-7a46-4607-9abf-cf0ce385c6e5"
}
}
}
}


➜ chalice_project git:(main) ✗ poetry run chalice deploy --stage main
/home/phil/Projects/adaptive-serverless/.venv/lib/python3.11/site-packages/_distutils_hack/__init__.py:26: UserWarning: Setuptools is replacing distutils.
warnings.warn("Setuptools is replacing distutils.")
Creating deployment package.
Reusing existing deployment package.
Updating policy for IAM role: chalice-project-main
Creating lambda function: chalice-project-main
Creating Rest API
Creating custom domain name: main.phil-test-sandbox.osedea.dev
Creating api mapping: /
Resources deployed:
- Lambda ARN: arn:aws:lambda:ca-central-1:795273512252:function:chalice-project-main
- Rest API URL: https://agiyhl5yxh.execute-api.ca-central-1.amazonaws.com/api/
- Custom domain name:
HostedZoneId: Z19DQILCV0OWEC
AliasDomainName: d-g0asx30116.execute-api.ca-central-1.amazonaws.com
➜ chalice_project git:(main) ✗


API Gateway > APIs > Custom domain names
Custom domain names
Domain names
Create
Name
main.phil-test-sandbox.osedea.dev
Domain details
Delete
Edit
Domain name
main.phil-test-sandbox.osedea.dev
TLS version
TLS 1.2
Status
Available
Configurations
API mappings
Tags
Endpoint configuration
Edit
API Gateway domain name
d-g0asx30116.execute-api.ca-central-1.amazonaws.com
Hosted zone ID
Z19DQILCV0OWEC
API endpoint type
Regional
ACM certificate ARN
arn:aws:acm:ca-central-1:795273512252:certificate/daa299f2-7a46-4607-9abf-cf0ce385c6e5

Route 53 > Hosted zones > phil-test-sandbox.osedea.dev > Create record

Create record Info

Quick create record

Switch to wizard

▼ Record 1

Delete

Record name | Info

main .phil-test-sandbox.osedea.dev

Keep blank to create a record for the root domain.

Record type | Info

A – Routes traffic to an IPv4 address and some AWS resources

Alias

Route traffic to | Info

Alias to API Gateway API

Canada (Central)

d-g0asx30116.execute-api.ca-central-1.amazonaws.com

Use: "d-g0asx30116.execute-api.ca-central-1.amazonaws.com"

d-g0asx30116.execute-api.ca-central-1.amazonaws.com

Simple routing

Yes

Add another record

Cancel Create records

Cool 👍

```
➜  chalice_project git:(main) ✗ curl https://main.phil-test-sandbox.osedea.dev
{"hello":"world"}%
```

<10ms execution time

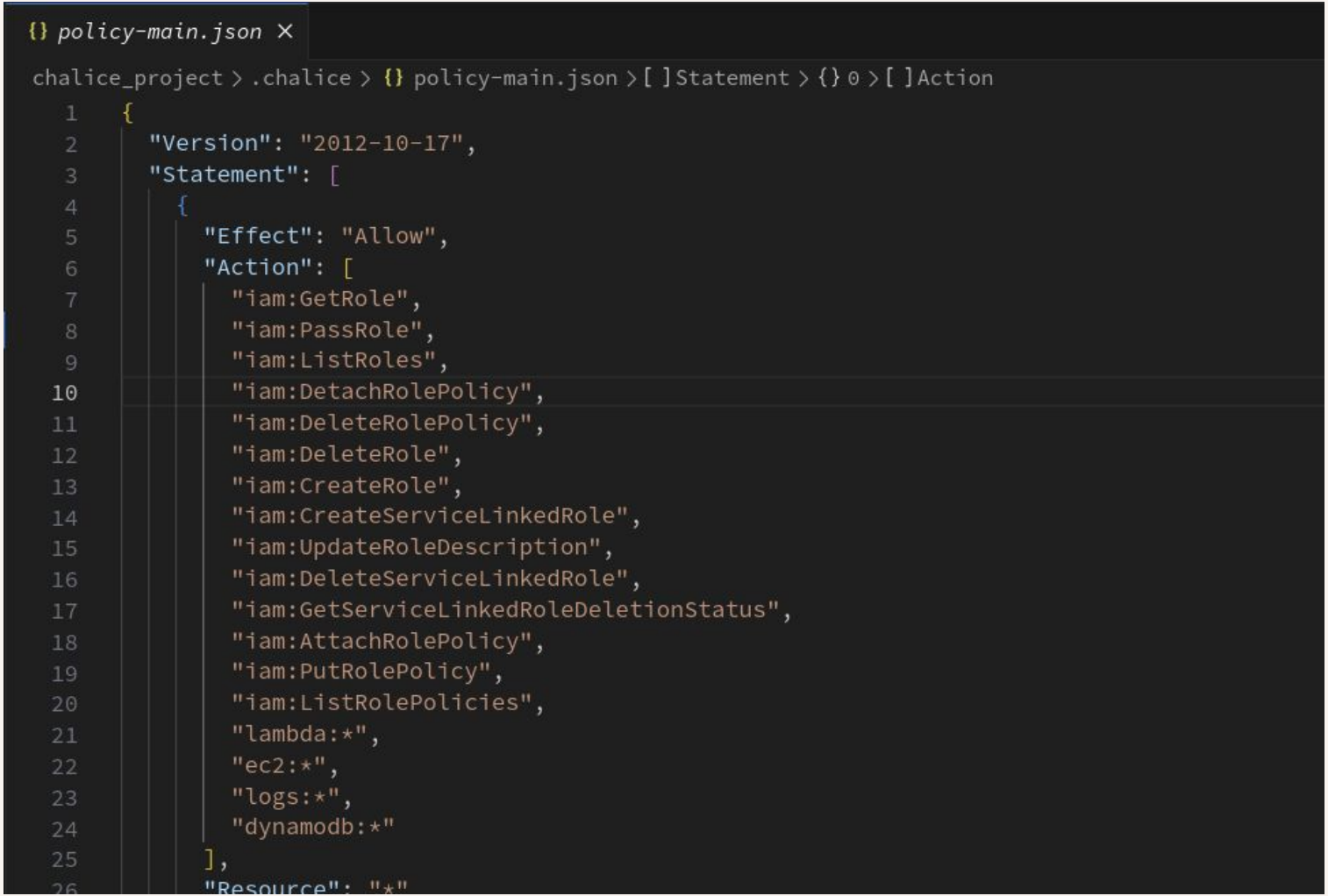
Upgrading To A Polls App

Polls App should allows users to

- Create a new poll with options
- Vote for an option on a poll
- View a poll, it’s options and vote counts

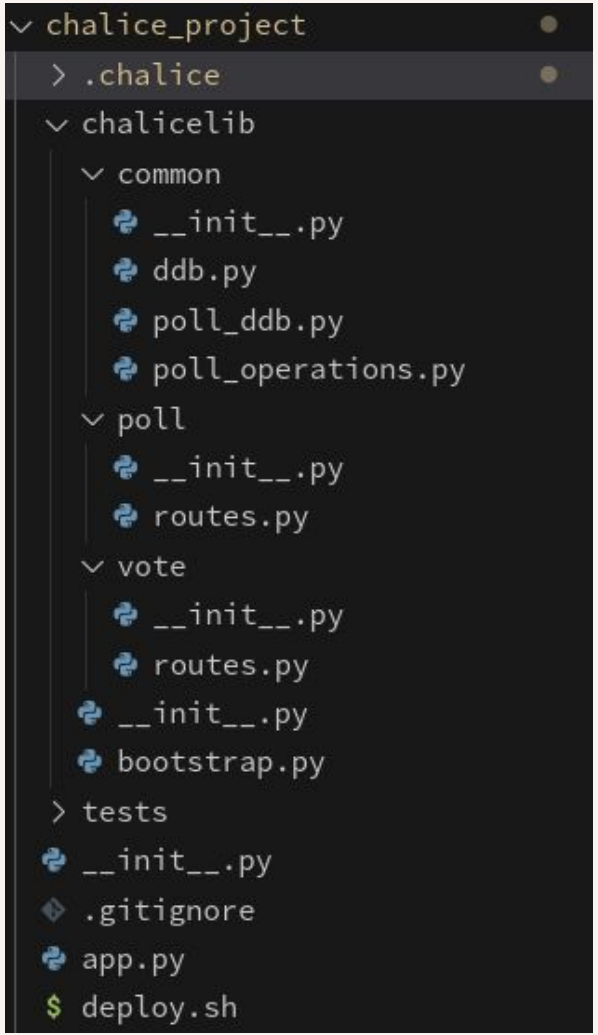
Creating a Polls app

1. Give more permissions to the app
2. Add endpoints for managing polls
3. Add endpoint for votes
4. Creating a DynamoDB table
5. Setting up operations to DynamoDB
6. Creating a (small) deploy Script
7. Deploying our new Polls app
8. Manually testing the app’s features
9. Taking a look at performance


{} policy-main.json
chalice_project > .chalice > {} policy-main.json > [ ] Statement > {} 0 > [ ] Action
{
"Version": "2012-10-17",
"Statement": [
{
"Effect": "Allow",
"Action": [
"iam:GetRole",
"iam:PassRole",
"iam:ListRoles",
"iam:DetachRolePolicy",
"iam:DeleteRolePolicy",
"iam:DeleteRole",
"iam:CreateRole",
"iam:CreateServiceLinkedRole",
"iam:UpdateRoleDescription",
"iam:DeleteServiceLinkedRole",
"iam:GetServiceLinkedRoleDeletionStatus",
"iam:AttachRolePolicy",
"iam:PutRolePolicy",
"iam:ListRolePolicies",
"lambda:*",
"ec2:*",
"logs:*",
"dynamodb:*"
],
"Resource": "*"


{} config.json
chalice_project > .chalice > {} config.json > ...
{
"version": "2.0",
"app_name": "chalice-project",
"automatic_layer": true,
"stages": {
"main": {
"api_gateway_stage": "api",
"autogen_policy": false, # policy-main.json
"api_gateway_endpoint_type": "REGIONAL",
"api_gateway_custom_domain": {
"domain_name": "main.phil-test-sandbox.osedea.dev",
"tls_version": "TLS_1_2",
"certificate_arn": "arn:aws:acm:ca-central-1:795273512252:certificate/
daa299f2-7a46-4607-9abf-cf0ce385c6e5"
}
}
}
}

- App (entrypoint)
- Bootstrap (instantiate chalice)
- Common Library (shared w/ services)
- Poll Service
- Vote Service
- Deploy Script



app.py

chalice_project > app.py

```python
from chalicelib.bootstrap import app
import chalicelib.poll.routes
import chalicelib.vote.routes

```

we use *chalicelib* as the top level package for our application

Chalice needs *app* in it's entrypoint module to auto-wire

bootstrap.py

chalice_project > chalicelib > bootstrap.py > ...

```python
from chalice import Chalice

app = Chalice(app_name='chalice_project')
```

We want to reuse it across our services

routes.py

chalice_project > chalicelib > poll > routes.py > create_poll_route

```python
from urllib.parse import unquote
from chalice import Response
from chalicelib.bootstrap import app
from chalicelib.common.poll_operations import retrieve_poll, add_poll

@app.route('/poll/{poll_name}', methods=['GET'])
def get_poll_route(poll_name: str):
    unquoted_name = unquote(poll_name)
    result = retrieve_poll(poll_name=unquoted_name)
    if result is None:
        return Response(status_code=404, body=f'Poll \'{unquoted_name}\' does not exist.')
    return Response(status_code=200, body=result)


@app.route('/poll', methods=['POST'])
def create_poll_route():
    poll_request_body: dict = app.current_request.json_body
    unquoted_name = unquote(poll_request_body['name'])
    add_poll(
        poll_name=unquoted_name,
        poll_options=poll_request_body['options']
    )
    return Response(status_code=200, body=retrieve_poll(poll_name=unquoted_name))
```

chalice_project > chalicelib > vote > routes.py > ...

```python
from urllib.parse import unquote
from chalice import Response
from chalicelib.bootstrap import app
from chalicelib.common.poll_operations import retrieve_poll, vote


@app.route('/vote/{poll_name}/{poll_option}', methods=['POST'])
def get_poll_route(poll_name: str, poll_option: str):
    unquoted_name = unquote(poll_name)
    unquoted_option = unquote(poll_option)
    vote(poll_name=unquoted_name, option_name=unquoted_option)
    poll_result = retrieve_poll(poll_name=unquoted_name)
    return Response(status_code=200, body=f'''
You've voted for \'{unquoted_option}\' in the \'{unquoted_name}\' poll!
============================================
{poll_result}''')

```

chalice_project > chalicelib > common > poll_operations.py > …

```python
from datetime import datetime, timedelta
from typing import List, Optional
from chalicelib.common.poll_ddb import create_poll, get_poll, increment_vote


def retrieve_poll(poll_name: str) -> Optional[str]:
    poll_opts = get_poll(name=poll_name)
    if poll_opts is None:
        return None
    name, options, close_at = poll_opts
    options_text: str = '\n'.join([f'| {option_name.ljust(32, " ")} | {str(option_votes).ljust(7, " ")} |'
    for option_name, option_votes in options])
    close_at_datetime: datetime = datetime.fromisoformat(close_at)
    is_closed: bool = datetime.now() > close_at_datetime

    return f'''
Poll {name} {'is closed ❌' if is_closed else 'is still open 🕒 until '+close_at_datetime.strftime("%c")}
+---------------------------------+---------+
|             Options             |  Votes  |
+---------------------------------+---------+
{options_text}
+---------------------------------+---------+
'''

def add_poll(poll_name: str, poll_options: List[str]) -> None:
    close_at_datetime: datetime = datetime.now() + timedelta(minutes=30)
    create_poll(name=poll_name, options=[(option, 0,) for option in poll_options],
    close_at=close_at_datetime.isoformat())


def vote(poll_name: str, option_name: str) -> None:
    increment_vote(poll_name=poll_name, option_name=option_name)
```

chalice_project > chalicelib > common > poll_ddb.py > create_table

```python
from typing import List, Optional, Tuple
import logging
from boto3.dynamodb.conditions import Key
from chalicelib.common.ddb import get_ddb_client, get_ddb_resource

def _get_table_name() -> str:
    return 'polls'

def does_table_exist(table_name: str) -> bool:
    return table_name in get_ddb_client().list_tables()['TableNames']

def get_or_create_table():
    if not does_table_exist(_get_table_name()):
        create_table()
    return get_ddb_resource().Table(_get_table_name())
```

chalice_project > chalicelib > common > poll_ddb.py > create_table

```python
def create_table():
    client = get_ddb_client()
    table_name = _get_table_name()
    get_ddb_client().create_table(
        TableName=_get_table_name(),
        KeySchema=[
            {
                'AttributeName': 'name',
                'KeyType': 'HASH'
            },
            {
                'AttributeName': 'option',
                'KeyType': 'RANGE'
            }
        ],
        AttributeDefinitions=[
            {
                'AttributeName': 'name',
                'AttributeType': 'S'
            },
            {
                'AttributeName': 'option',
                'AttributeType': 'S'
            }
        ],
        BillingMode='PAY_PER_REQUEST'
    )
    logging.info('Wait for table %s to be created.', table_name)
    waiter = client.get_waiter('table_exists')
    waiter.wait(TableName=table_name)
    logging.info('Created table %s.', table_name)
```

How data is partitioned

How data is sorted

string

string

no provisioning, auto-scale

SQL Equivalents

INSERT INTO polls(name, option, close_at, votes) VALUES (...);

SELECT * FROM polls WHERE name = ...;

UPDATE polls SET votes = votes + 1 WHERE name = ... AND option = ...;

chalice_project > chalicelib > common > poll_ddb.py > create_table

```python
def create_poll(name: str, options: List[Tuple[str, int]], close_at: str):
    with get_or_create_table().batch_writer() as batch:
        for option_name, option_votes in options:
            batch.put_item(
                Item={
                    'name': name,
                    'option': option_name,
                    'close_at': close_at,
                    'votes': option_votes
                }
            )

def get_poll(name: str) -> Optional[Tuple[str, List[Tuple[str, int]], str]]:
    query_items = get_or_create_table().query(KeyConditionExpression=Key('name').eq(name))['Items']
    if len(query_items) == 0:
        return None
    options = [(query_item['option'], query_item['votes'],) for query_item in query_items]
    close_at = query_items[0]['close_at']
    return name, options, close_at

def increment_vote(poll_name: str, option_name: str) -> None:
    get_ddb_resource().Table(_get_table_name()).update_item(
        Key = {
            'name': poll_name,
            'option': option_name
        },
        ExpressionAttributeNames={
            '#votes': 'votes'
        },
        ExpressionAttributeValues={
            ':value': 1
        },
        UpdateExpression="SET #votes = #votes + :value",
    )
```


$ deploy.sh
chalice_project > $ deploy.sh
1 #!/usr/bin/env bash
2
3 poetry export > requirements.txt
4 poetry run chalice deploy --stage main

```
➜  chalice_project git:(main) ✗ ./deploy.sh
/home/phil/Projects/adaptive-serverless/.venv/lib/python3.11/site-packages/_distutils
_hack/__init__.py:26: UserWarning: Setuptools is replacing distutils.
  warnings.warn("Setuptools is replacing distutils.")
Creating shared layer deployment package.
  Reusing existing shared layer deployment package.
Creating app deployment package.
Updating lambda layer: chalice-project-main-managed-layer
Updating policy for IAM role: chalice-project-main-api_handler
Updating lambda function: chalice-project-main
Updating rest API
Updating custom domain name: main.phil-test-sandbox.osedea.dev
Resources deployed:
  - Lambda Layer ARN: arn:aws:lambda:ca-central-1:795273512252:layer:chalice-project-
main-managed-layer:8
  - Lambda ARN: arn:aws:lambda:ca-central-1:795273512252:function:chalice-project-mai
n
  - Rest API URL: https://agiyhl5yxh.execute-api.ca-central-1.amazonaws.com/api/
  - Custom domain name:
      HostedZoneId: Z19DQILCV0OWEC
      AliasDomainName: d-g0asx30116.execute-api.ca-central-1.amazonaws.com
➜  chalice_project git:(main) ✗
```

Cool...? 👀

```
➜  ~ curl -w 'Total: %{time_total}s\n' \
  --header "Content-Type: application/json" \
  --request POST \
  --data '{"name": "Best Cookie", "options": ["Chocolate Chip", "Oreo", "Prefer not to Answer"]}' \
  https://main.phil-test-sandbox.osedea.dev/poll

Poll Best Cookie is still open 🕒 until Wed Feb 21 20:20:27 2024
+---------------------------------+---------+
|             Options             |  Votes  |
+---------------------------------+---------+
| Chocolate Chip                  | 0       |
| Oreo                            | 0       |
| Prefer not to Answer            | 0       |
+---------------------------------+---------+
Total: 23.951536s
```

😱 DB + COLD START


→ chalice_project git:(main) ✗ curl -w 'Total: %{time_total}s\n' \
--header "Content-Type: application/json" \
--request POST \
--data '{"name": "Best Sundae", "options": ["Chocolate Chip", "Oreo", "Prefer not to Answer"]}' \
https://main.phil-test-sandbox.osedea.dev/poll
Poll Best Sundae is still open 🕒 until Thu Feb 22 00:55:57 2024
Options	Votes
Chocolate Chip	0
Oreo	0
Prefer not to Answer	0
Total: 3.623283s
COLD START
→ chalice_project git:(main) ✗ curl -w 'Total: %{time_total}s\n' \
--header "Content-Type: application/json" \
--request POST \
--data '{"name": "Best Milkshake", "options": ["Chocolate Chip", "Oreo", "Prefer not to Answer"]}' \
https://main.phil-test-sandbox.osedea.dev/poll
Poll Best Milkshake is still open 🕒 until Thu Feb 22 00:56:09 2024
Options	Votes
Chocolate Chip	0
Oreo	0
Prefer not to Answer	0
Total: 0.300457s
WARM START
👍
→ chalice_project git:(main) ✗

```
→  ~ curl -w 'Total: %{time_total}s\n' --request POST \
"https://main.phil-test-sandbox.osedea.dev/vote/Best%20Cookie/Chocolate%20Chip"

You've voted for 'Chocolate Chip' in the 'Best Cookie' poll!
============================================

Poll Best Cookie is still open 🕒 until Wed Feb 21 20:20:27 2024
+-----------------------------------+---------+
|              Options              |  Votes  |
+-----------------------------------+---------+
| Chocolate Chip                    | 3       |
| Oreo                              | 0       |
| Prefer not to Answer              | 0       |
+-----------------------------------+---------+
Total: 0.253090s
```

```
→  ~ curl -w 'Total: %{time_total}s\n' --request GET \
"https://main.phil-test-sandbox.osedea.dev/poll/Best%20Cookie/"

Poll Best Cookie is still open 🕒 until Wed Feb 21 20:20:27 2024
+-----------------------------------+---------+
|              Options              |  Votes  |
+-----------------------------------+---------+
| Chocolate Chip                    | 3       |
| Oreo                              | 0       |
| Prefer not to Answer              | 0       |
+-----------------------------------+---------+
Total: 0.207341s
```

```
➜  ~ curl -s -I -w 'Total: %{time_total}s\n' --request POST \
"https://main.phil-test-sandbox.osedea.dev/vote/Best%20Cookie/Chocolate%20Chip" | tail -n 1
Total: 0.234462s
➜  ~ curl -s -I -w 'Total: %{time_total}s\n' --request POST \
"https://main.phil-test-sandbox.osedea.dev/vote/Best%20Cookie/Chocolate%20Chip" | tail -n 1
Total: 0.248901s
➜  ~ curl -s -I -w 'Total: %{time_total}s\n' --request POST \
"https://main.phil-test-sandbox.osedea.dev/vote/Best%20Cookie/Chocolate%20Chip" | tail -n 1
Total: 0.218777s
➜  ~ curl -s -I -w 'Total: %{time_total}s\n' --request POST \
"https://main.phil-test-sandbox.osedea.dev/vote/Best%20Cookie/Chocolate%20Chip" | tail -n 1
Total: 0.239959s
```

DurationInMS	BilledDurationInMS	MemorySetInMB	MemoryUsedInMB
87.17	88.0	128.0	77.0
65.66	66.0	128.0	77.0
110.96	111.0	128.0	77.0
107.83	108.0	128.0	77.0

Running Multiple Environments

Running Multiple Envs

1. Create Chalice config template
2. Add environment variable for env naming
3. Modify code to account for env naming
4. Supercharge deploy script
5. Deploy new branch to new env
6. Test new env

```
→ chalice_project git:(main) mv .chalice/policy-main.json .chalice/policy.template.json
→ chalice_project git:(main) ✗ mv .chalice/config.json .chalice/config.template.json
```

chalice_project > .chalice > {} config.template.json > …

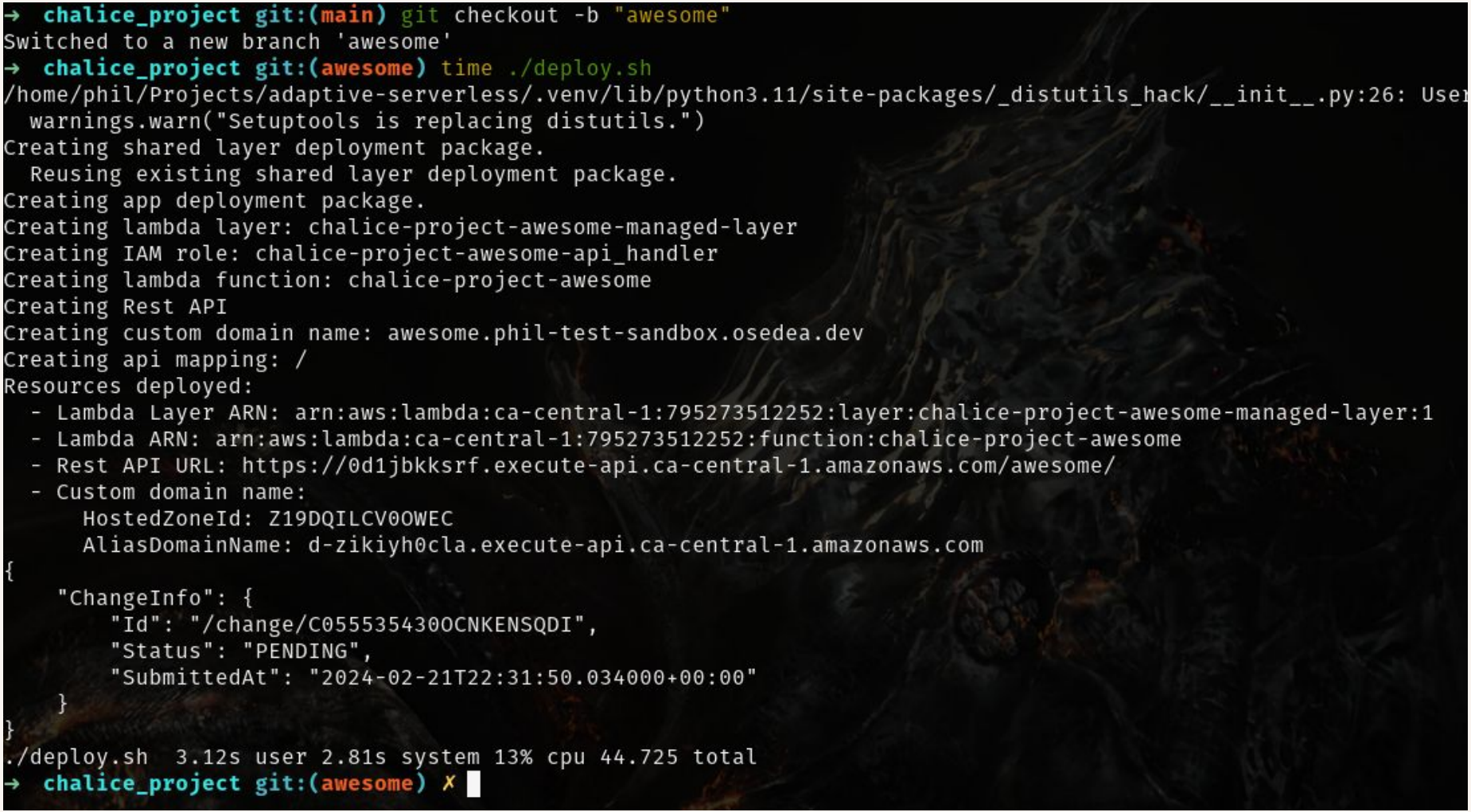
```
{
  "version": "2.0",
  "app_name": "chalice-project",
  "automatic_layer": true,
  "stages": {
    "__ENV_NAME__": {
      "api_gateway_stage": "__ENV_NAME__",
      "autogen_policy": false,
      "api_gateway_endpoint_type": "REGIONAL",
      "api_gateway_custom_domain": {
        "domain_name": "__ENV_NAME__.phil-test-sandbox.osedea.dev",
        "tls_version": "TLS_1_2",
        "certificate_arn": "arn:aws:acm:ca-central-1:795273512252:certificate/
        daa299f2-7a46-4607-9abf-cf0ce385c6e5"
      }
    }
  },
  "environment_variables": {
    "ENVIRONMENT_NAME": "__ENV_NAME__"
  }
}
```

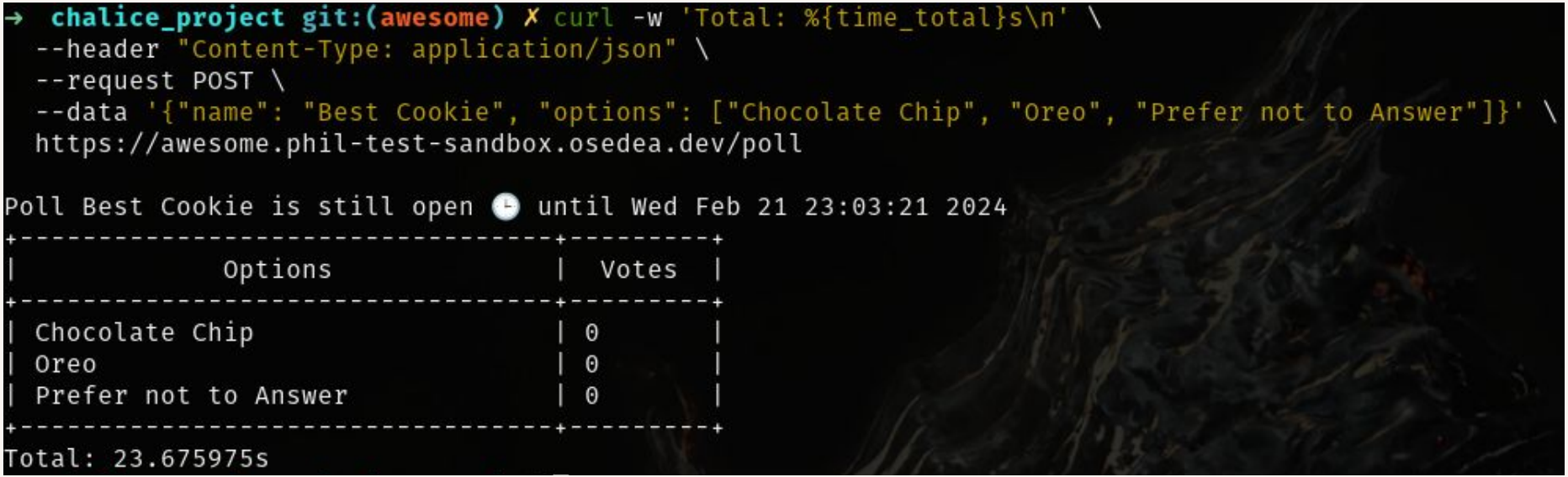
chalice_project > chalicelib > common > poll_ddb.py > create_table

```python
from typing import List, Optional, Tuple
import logging
from os import environ
from boto3.dynamodb.conditions import Key
from chalicelib.common.ddb import get_ddb_client, get_ddb_resource

def _get_table_name() -> str:
    env_name: str = environ.get('ENVIRONMENT_NAME', 'default')
    return f'{env_name}__polls'
```

chalice_project > $ deploy.sh

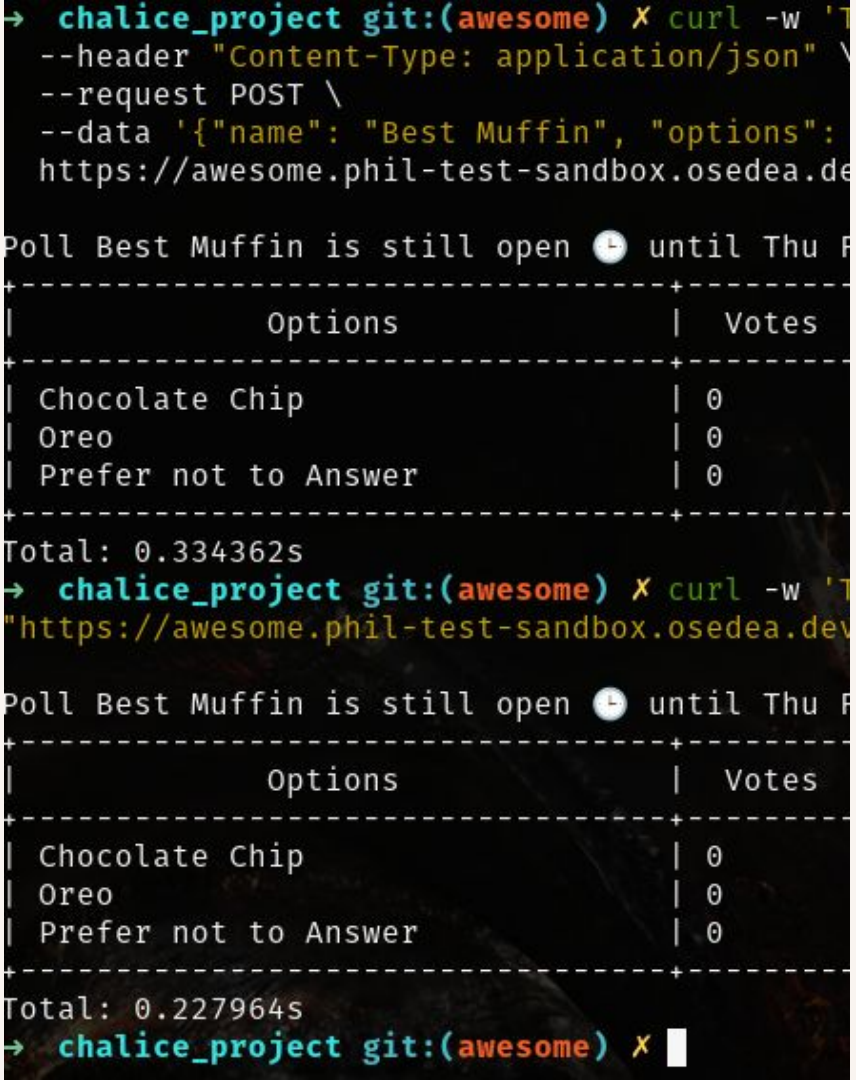
```
#!/usr/bin/env bash
set -e

# get the current branch name
ENV_NAME="$(git rev-parse --abbrev-ref HEAD)"

# prepare config & requirements
cat .chalice/config.template.json | sed 's/__ENV_NAME__/'$ENV_NAME'/' > .chalice/config.json
cp .chalice/policy.template.json .chalice/policy-$ENV_NAME.json
poetry export > requirements.txt

# deploy chalice
poetry run chalice deploy --stage $ENV_NAME

# grab values from AWS
DNS_ZONEID=$(aws route53 list-hosted-zones --output json | jq '.HostedZones[0].Id' | sed -r 's/^"\/hostedzone\/(.*)"$/\1/')
API_GATEWAY_ZONEID=$(cat .chalice/deployed/$ENV_NAME.json | jq '.resources[] | select(.name=="api_gateway_custom_domain").hosted_zone_id' | sed 's/"//g')
API_GATEWAY_DNS_NAME=$(cat .chalice/deployed/$ENV_NAME.json | jq '.resources[] | select(.name=="api_gateway_custom_domain").alias_domain_name' | sed 's/"//g')
DNS_CURRENT_RECORD_SET=$(aws route53 list-resource-record-sets --hosted-zone-id $DNS_ZONEID | jq '.ResourceRecordSets[] | select((.Name=="'$ENV_NAME'.phil-test-sandbox.osedea.dev.") and (.Type=="A"))')

# update or create the A record
DNS_RECORD_CHANGE_ACTION="UPSERT"
if [[ -z $DNS_CURRENT_RECORD_SET ]]; then
    DNS_RECORD_CHANGE_ACTION="CREATE"
fi

aws route53 change-resource-record-sets --no-cli-pager \
  --hosted-zone-id $DNS_ZONEID \
  --change-batch '{"Changes":[{"Action":"'$DNS_RECORD_CHANGE_ACTION'","ResourceRecordSet":{"Name":"'$ENV_NAME'.phil-test-sandbox.osedea.dev.","Type":"A","AliasTarget": {"HostedZoneId":"'$API_GATEWAY_ZONEID'", "DNSName": "'$API_GATEWAY_DNS_NAME'.","EvaluateTargetHealth": true}}}]}'
```

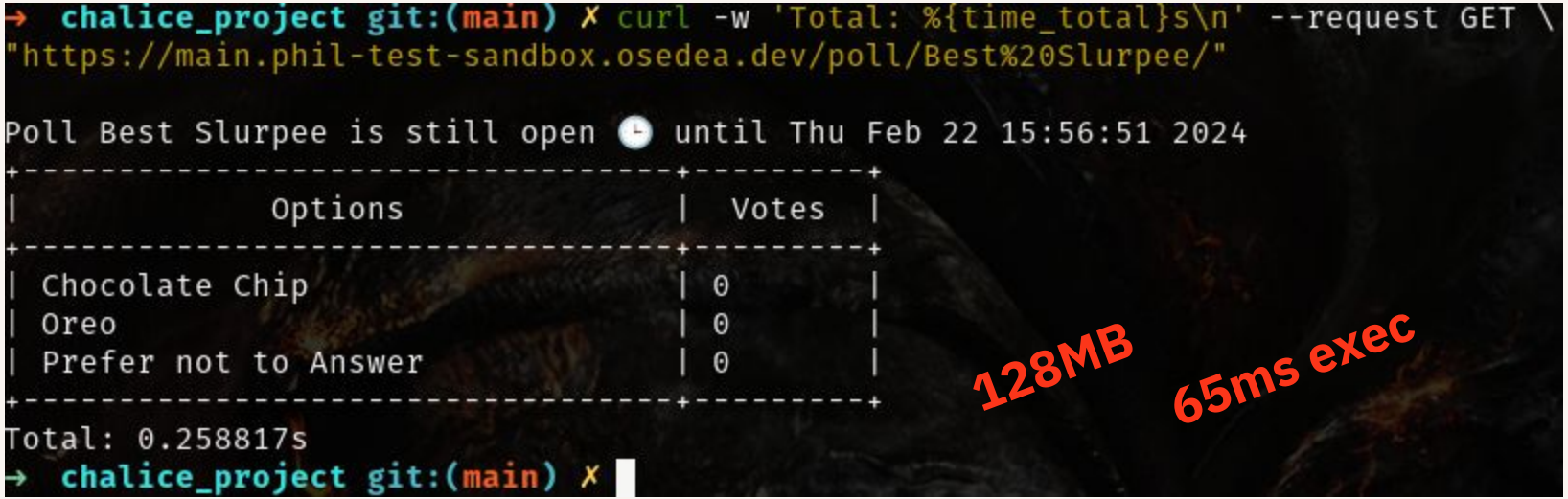

→ chalice_project git:(main) git checkout -b "awesome"
Switched to a new branch 'awesome'
→ chalice_project git:(awesome) time ./deploy.sh
/home/phil/Projects/adaptive-serverless/.venv/lib/python3.11/site-packages/_distutils_hack/__init__.py:26: User
warnings.warn("Setuptools is replacing distutils.")
Creating shared layer deployment package.
Reusing existing shared layer deployment package.
Creating app deployment package.
Creating lambda layer: chalice-project-awesome-managed-layer
Creating IAM role: chalice-project-awesome-api_handler
Creating lambda function: chalice-project-awesome
Creating Rest API
Creating custom domain name: awesome.phil-test-sandbox.osedea.dev
Creating api mapping: /
Resources deployed:
- Lambda Layer ARN: arn:aws:lambda:ca-central-1:795273512252:layer:chalice-project-awesome-managed-layer:1
- Lambda ARN: arn:aws:lambda:ca-central-1:795273512252:function:chalice-project-awesome
- Rest API URL: https://0d1jbkksrf.execute-api.ca-central-1.amazonaws.com/awesome/
- Custom domain name:
HostedZoneId: Z19DQILCV0OWEC
AliasDomainName: d-zikiyh0cla.execute-api.ca-central-1.amazonaws.com
{
"ChangeInfo": {
"Id": "/change/C055535430OCNKENSQDI",
"Status": "PENDING",
"SubmittedAt": "2024-02-21T22:31:50.034000+00:00"
}
}
./deploy.sh 3.12s user 2.81s system 13% cpu 44.725 total
→ chalice_project git:(awesome) ✗

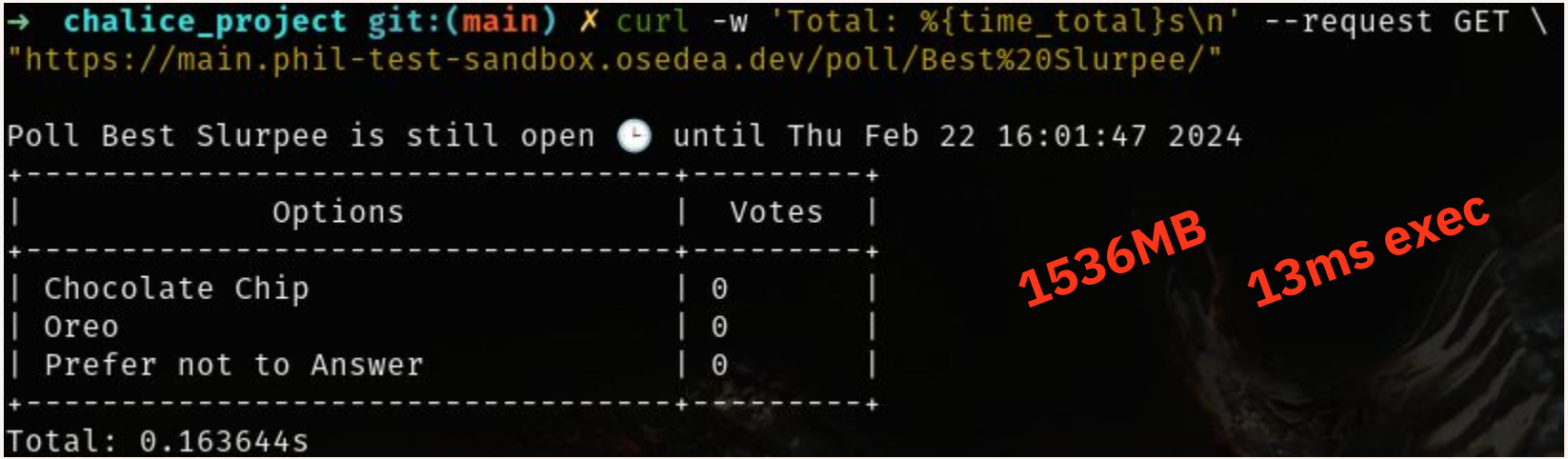

➜ chalice_project git:(awesome) ✗ curl -w 'Total: %{time_total}s\n' \
--header "Content-Type: application/json" \
--request POST \
--data '{"name": "Best Cookie", "options": ["Chocolate Chip", "Oreo", "Prefer not to Answer"]}' \
https://awesome.phil-test-sandbox.osedea.dev/poll
Poll Best Cookie is still open 🕒 until Wed Feb 21 23:03:21 2024
Options	Votes
Chocolate Chip	0
Oreo	0
Prefer not to Answer	0
Total: 23.675975s

Cool 👍

- ~1m30s for a fresh environment
- 200-350ms (warm) calls
- Total Price of ~4USD/Request




→ chalice_project git:(main) ✗ curl -w 'Total: %{time_total}s\n' --request GET \
"https://main.phil-test-sandbox.osedea.dev/poll/Best%20Slurpee/"
Poll Best Slurpee is still open 🕒 until Thu Feb 22 15:56:51 2024
Options	Votes
Chocolate Chip	0
Oreo	0
Prefer not to Answer	0
Total: 0.258817s
→ chalice_project git:(main) ✗
128MB
65ms exec


→ chalice_project git:(main) ✗ curl -w 'Total: %{time_total}s\n' --request GET \
"https://main.phil-test-sandbox.osedea.dev/poll/Best%20Slurpee/"
Poll Best Slurpee is still open 🕒 until Thu Feb 22 16:01:47 2024
Options	Votes
Chocolate Chip	0
Oreo	0
Prefer not to Answer	0
Total: 0.163644s
1536MB
13ms exec

Takeaways

- Chalice is simple *enough*
- Python Lambdas are fast *enough*
- Unoptimized config is cheap *enough*
- Setup is quick *enough*
- Mono-lambda is pretty sweet

🔧 Some Further Upgrades

- IP Whitelisting
- Delete Old Environments
- Setup on CI/CD Pipeline
- Write Tests (duh)
- Authentication
- Health Check (5m, keep warm)
- CloudFront CDN
  - Lambda@Edge

👓 Useful Links

- Lambda Getting Started
  - https://docs.aws.amazon.com/lambda/latest/dg/getting-started.html
- DynamoDB Getting Started
  - https://docs.aws.amazon.com/amazondynamodb/latest/developerguide/GettingStartedDynamoDB.html
- Chalice Repo
  - https://github.com/aws/chalice
- Chalice Workshop
  - https://chalice-workshop.readthedocs.io/
- Alternative: AWS SAM (Serverless Application Model)
  - https://aws.amazon.com/serverless/sam/

Thank You! Questions?

👨‍💻 Philippe Trépanier
linkedin.com/in/philippe-trepanier/
github.com/philtrep/

/ OSEDEA – **Projects** – **Careers**
linkedin.com/company/osedea/
osedea.com

Now Hiring!