order2desk example Part 2

Dr Peadar Grant

February 23, 2024

1 Scenario

A tech company allows its in office employees to order snacks and beverages to their working desk:

- Workers are assigned a Mac, Windows or Linux computer.
- They can order food / drinks by means of a Python script `order2desk.py` that presents a menu on the command-line and accepts their selection(s).
- The script automatically discovers their username and hostname.
- Once prepared, their catering partner then brings the order to their desk.

In the kitchen:

- A PC has a Python script that presents each order in sequence on a screen.

1.1 Improvements

The system has been working well, but some improvements are needed:

1. **Dynamically loaded menu** replacing the hardcoded beverage options.
2. **Lambda function** needs to be setup in **CloudFormation**.
3. **Resources** should be looked up in CloudFormation and not hardcoded.
4. **Permissions** need to be scoped to the actual actions on the actual resources.

2 System architecture

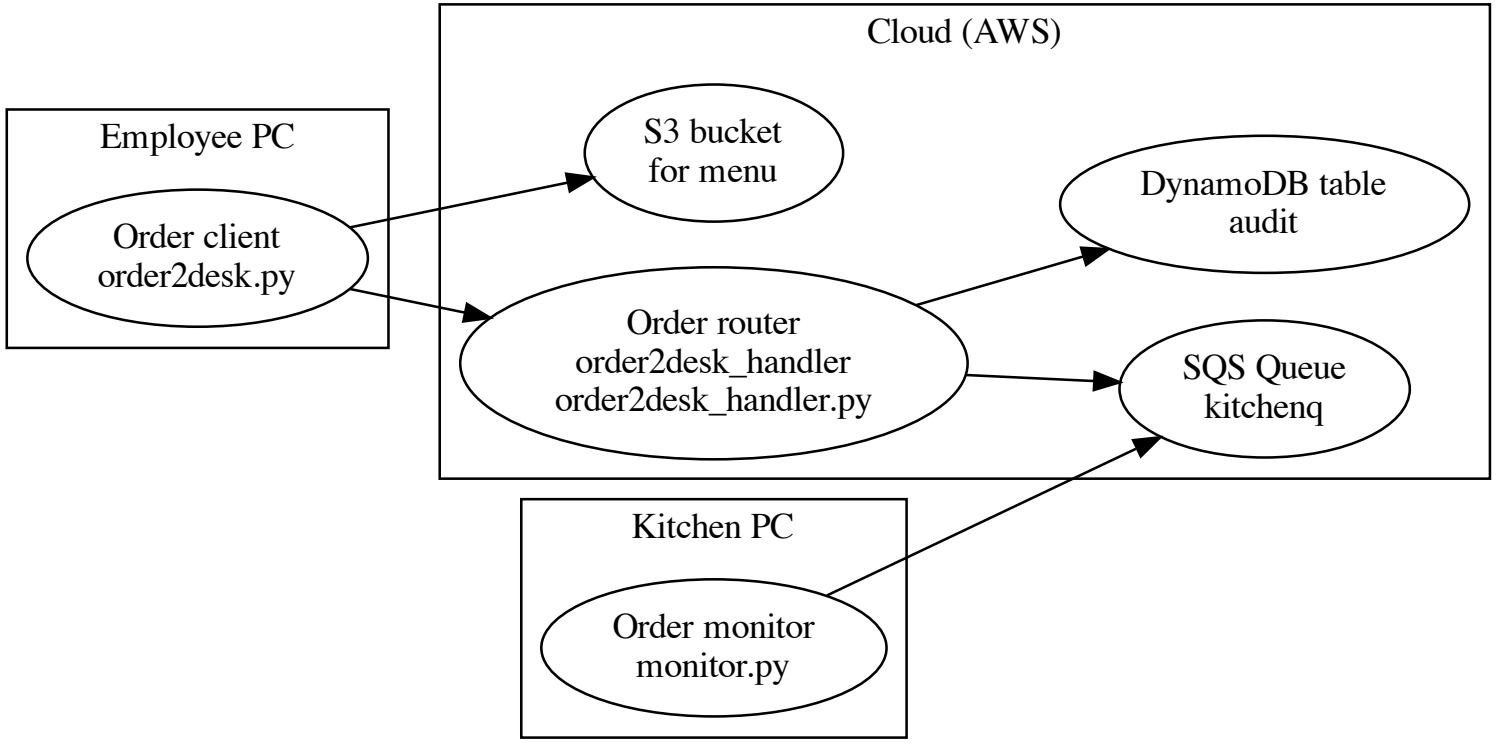

Figure 1: order2desk system

2.1 Dynamic lookup

We want all components to look up resources from the CloudFormation template:

- Outputs in CloudFormation template pointing to resources
- Dynamic lookup using a `cf_output` function in Python in `common.py`

2.2 Dynamic menu

- Menu is defined in a JSON file `menu.json`.
- Loaded by `order2desk.py` from S3 bucket.
- S3 bucket looked up from CloudFormation.

2.3 Permissions

What actions do the following components need to perform on what resources?

1. Order script `order2desk.py`
2. Order handler function
3. Monitor script `monitor.py`