

## User-driven Composable Infrastructure with CDK, Step Functions and CodeBuild - Luc van Donkersgoed



AWS User Groups | The Netherlands  
441 subscribers

Subscribe



7



Share



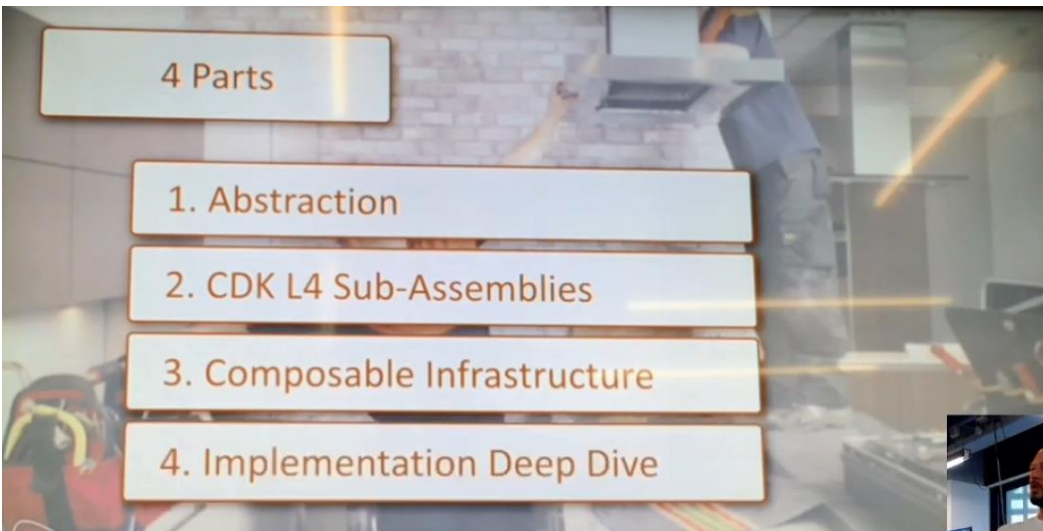
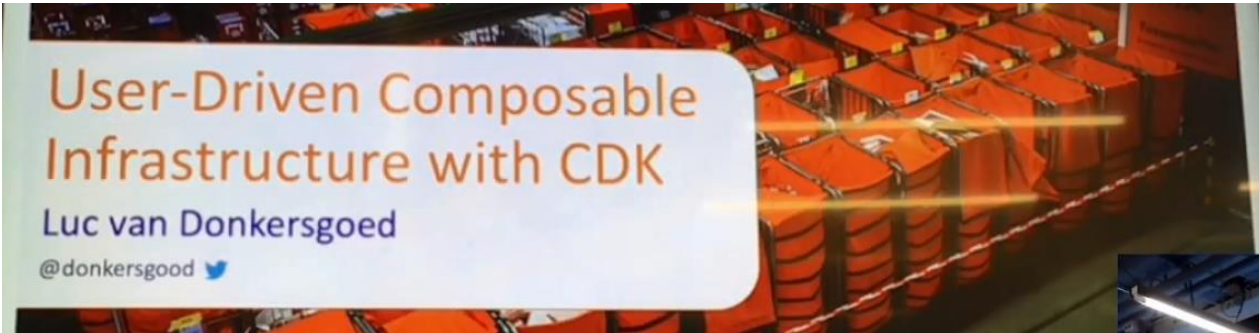
Download

333 views Jun 20, 2022

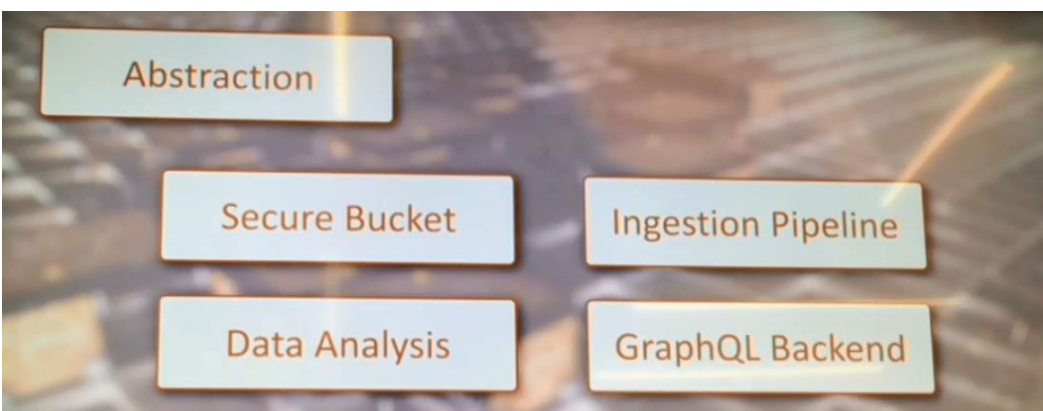
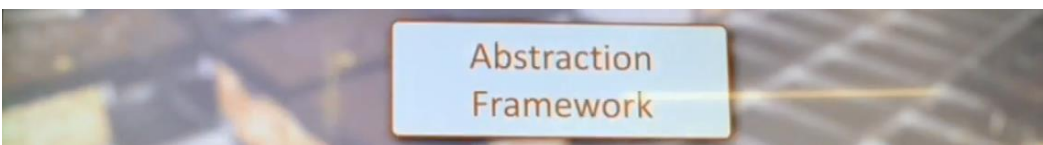
AWS Community Day Amsterdam 2022

User-driven Composable Infrastructure with CDK, Step Functions and CodeBuild

Luc van Donkersgoed



The **cdk synth** command will generate the CFN code for you



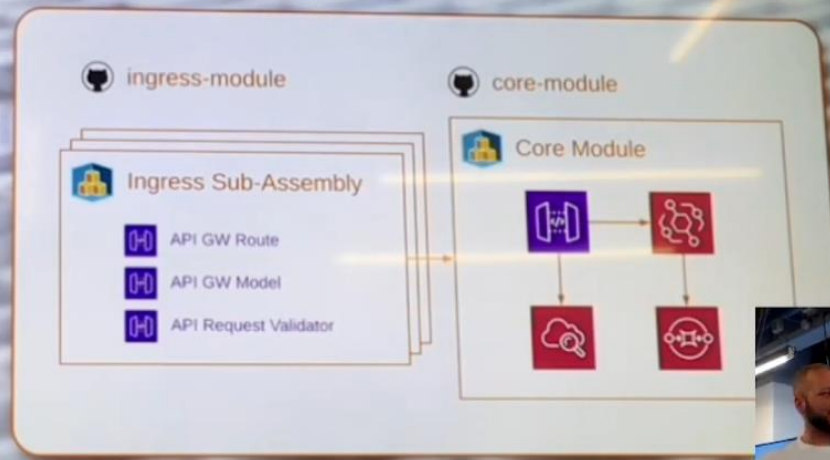




## CDK L4 Sub-Assemblies

The L4 Sub-Assembly is a new pattern and a new deployment method

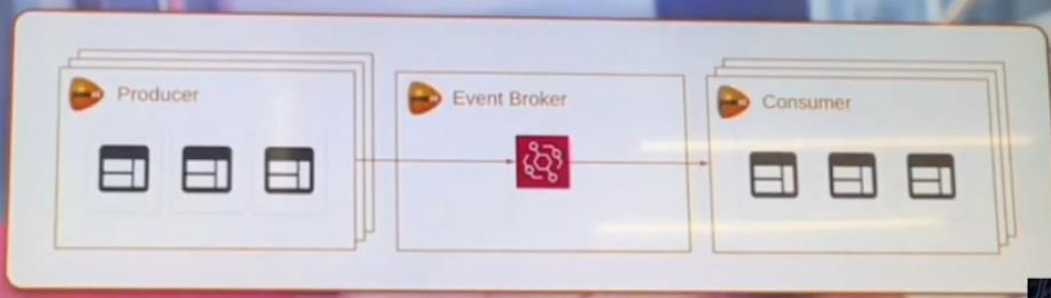
## L4 Sub-Assemblies



This is a standalone CDK project that can and will be deployed as a standalone stack, it does not deliver full functionality by itself but instead will integrate with another core module and use shared resources in that core module to provide the full functionality. The core module is its own CDK project with its CFN stack.

Why?

Why

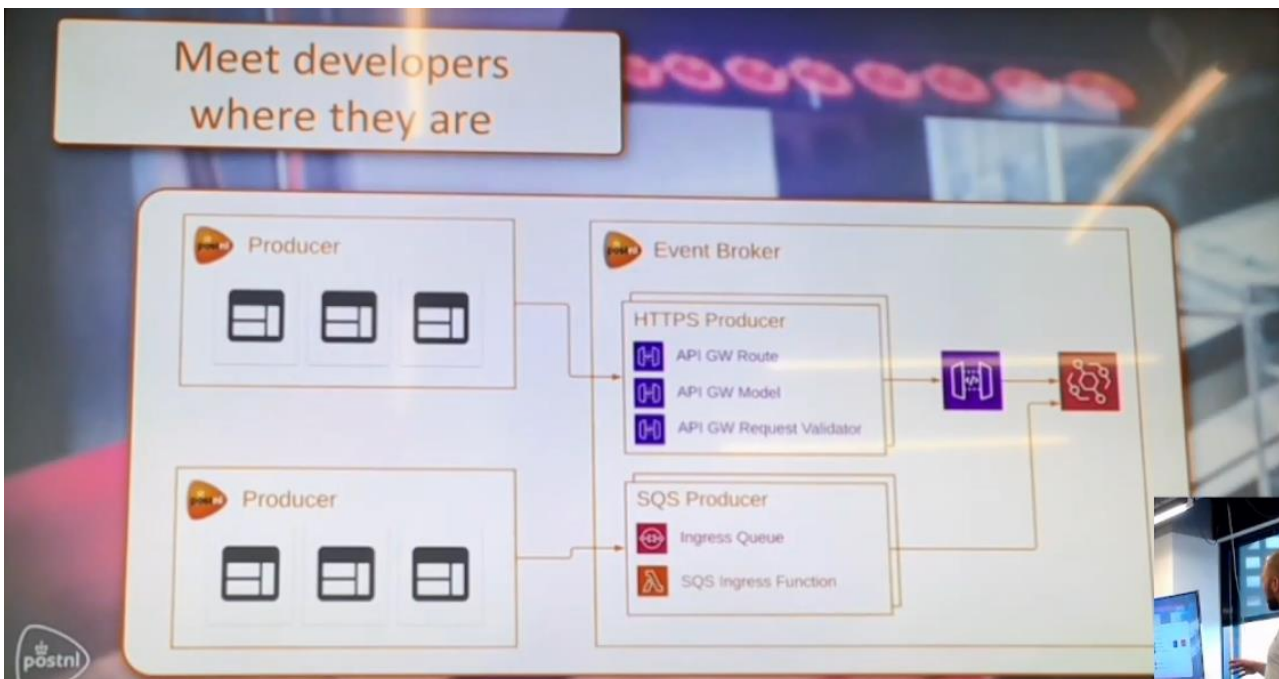
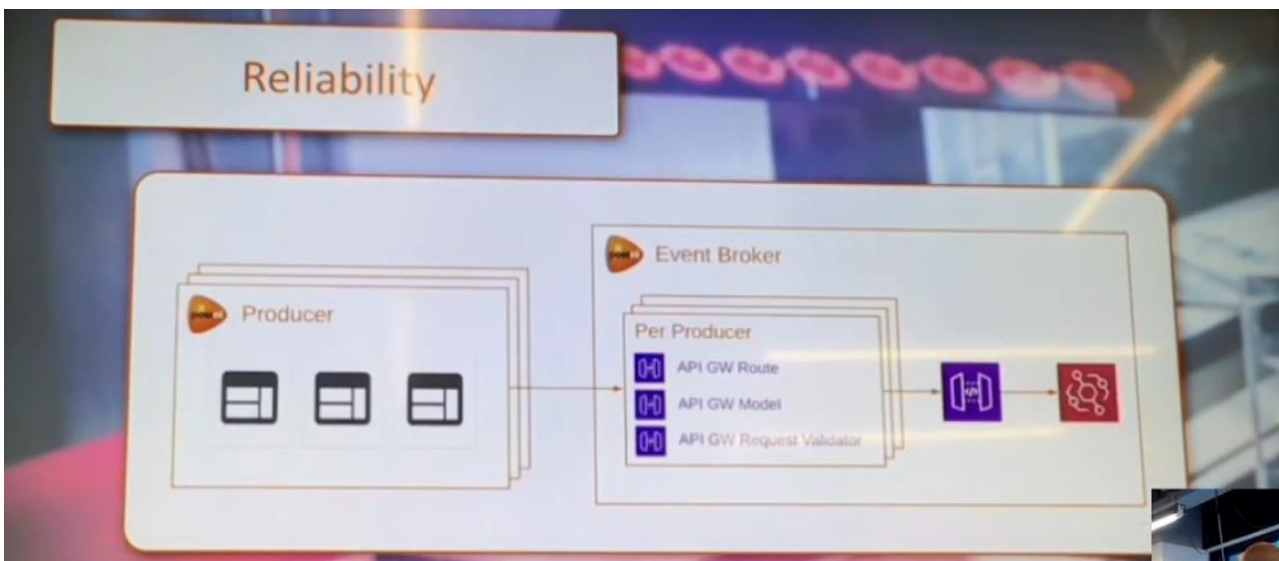


My team is responsible for building the central event broker based on AWS Event-Bridge

Innovate  
Faster



We put an API Gateway in front of Event-Bridge to do things like schema validation and events payload





## L4 Sub-Assemblies

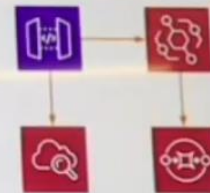
ingress-module

Ingress Sub-Assembly

- API GW Route
- API GW Model
- API Request Validator

core-module

Core Module



## Real-Life Example

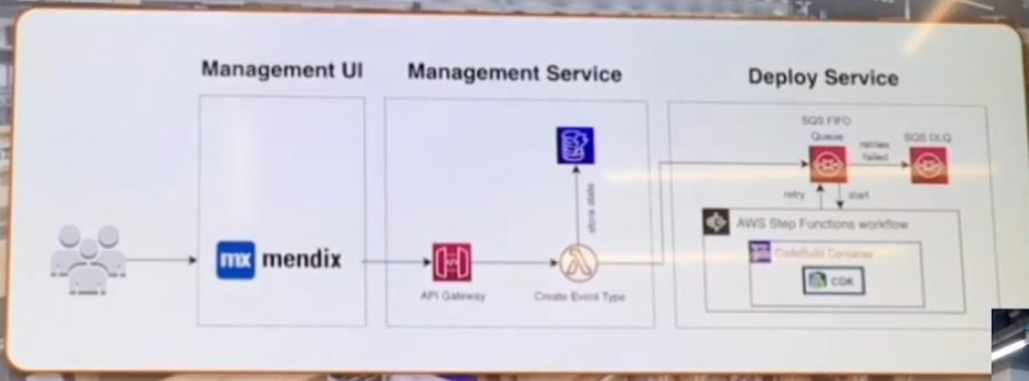
Stack name	Status
ProducerCeplaRetaillocationsV1	CREATE_COMPLETE
ProducerOompdDdaExecutioneventsV2	CREATE_COMPLETE
ProducerFsnExecutioneventV1	CREATE_COMPLETE
ProducerOompdDdaExecutioneventsV1	CREATE_COMPLETE
ProducerOompdDdaExecutioneventsDdV1	CREATE_COMPLETE
ProducerOompdprltemrouteupdateV1	CREATE_COMPLETE
ProducerOompdProtemrouteupdatesV1	CREATE_COMPLETE

## Real-Life Example

ProducerEdeSanitytestawsV1  
2022-05-20 15:45:29 UTC+0200  
CREATE\_COMPLETE

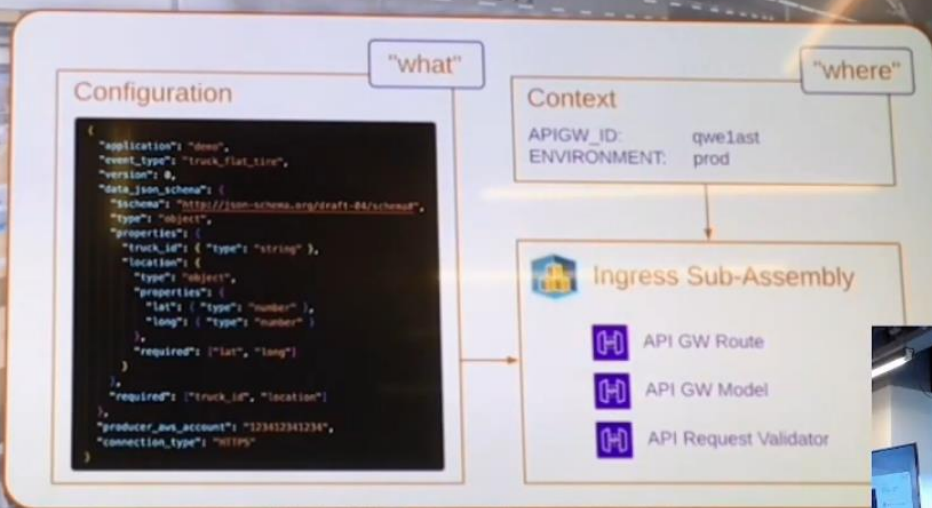
Physical ID	Type	Status
90147340-d287-11ec-988b-021b261d06b7	AWS::CDK::Metadata	CREATE_COMPLETE
ProduProdu6IBSUvh2nhK	AWS::ApiGateway::Model	CREATE_COMPLETE
w4e4xi	AWS::ApiGateway::Resource	CREATE_COMPLETE
Produ-Produ-1OTOWAT1SKIPM	AWS::ApiGateway::Method	CREATE_COMPLETE
pmwynk	AWS::ApiGateway::RequestValidator	CREATE_COMPLETE

## Implementation Deep Dive



Above diagram shows the deployment process of an L4 Sub-Assembly

## Implementation Deep Dive





## Configuration

```
{
  "application": "demo",
  "event_type": "truck_flat_tire",
  "version": 0,
  "data_json_schema": {
    "$schema": "http://json-schema.org/draft-04/schema#",
    "type": "object",
    "properties": {
      "truck_id": { "type": "string" },
      "location": {
        "type": "object",
        "properties": {
          "lat": { "type": "number" },
          "long": { "type": "number" }
        },
        "required": ["lat", "long"]
      }
    },
    "required": ["truck_id", "location"]
  },
  "producer_aws_account": "123412341234",
  "connection_type": "HTTPS"
}
```

## Context

APIGW\_ID: qwe1ast  
ENVIRONMENT: prod

## The CDK!

└─ .pylintrc

└─ app.py

( ) cdk.json

└─ input\_models

└─ \_\_pycache\_\_

└─ \_\_init\_\_.py

└─ models.py

```
from input_models.models import ProducerModel

if __name__ == "__main__":
    deploy_config = ProducerModel.model_from_file()
    build_app(deploy_config).synth()

@staticmethod
def model_from_dict(producer_deploy_config: dict) -> ProducerModel:
    """Instantiate a ProducerModel from an input dict.
    producer_connection_type = producer_deploy_config.get("connection_type")

    if producer_connection_type == "HTTPS":
        return HttpsProducerModel(producer_deploy_config)
    if producer_connection_type == "SQS":
        return SqsProducerModel(producer_deploy_config)

    raise RuntimeError(f"Unknown connection type: {producer_connection_type}")

@classmethod
def model_from_file(cls) -> ProducerModel:
    """Instantiate a ProducerModel from an input file."""
    with open(
        BaseSystemContext().path_to_producer_config_file, encoding="utf-8"
    ) as deploy_request_handle:
        msg = json.load(deploy_request_handle)
        return cls.model_from_dict(msg)
```



## Store Context Locally

└─ .pylintrc  
└─ app.py  
└─ cdk.json

└─ cdk.json  
└─ context.py  
└─ deploy.py

```
def build_app(producer_deploy_config: ProducerModel) -> App:
    """Build the CDK app. Wrapped in a function for testability."""
    local_app = App()
    context = BaseSystemContext()

def __init__(self) -> None:
    """Construct a new Context class."""
    self._stack_prefix = os.environ.get("STACK_PREFIX", "").lower()
    self._environment = self._osgetenv("ENVIRONMENT")
    self._deploy_role_arn = self._osgetenv("DEPLOY_ROLE_ARN", False)
    self._execution_role_arn = self._osgetenv("EXECUTION_ROLE_ARN", False)
    self._path_to_producer_config_file = "config/deploy.json"
```

## Create the Stack

└─ .pylintrc  
└─ app.py  
└─ cdk.json

```
def build_app(producer_deploy_config: ProducerModel) -> App:
    """Build the CDK app. Wrapped in a function for testability."""
    local_app = App()
    context = BaseSystemContext()

    EbeProducerComponentsCdkStack(
        scope=local_app,
        construct_id="EbeProducerComponentsCdkStack",
        producer_deploy_config=producer_deploy_config,
        stack_name=(
            pascalcase(context.stack_prefix)
            + pascalcase(f"Producer_{producer_deploy_config.unique_resource_id}")
        ),
        synthesizer=DefaultStackSynthesizer(
            deploy_role_arn=context.deploy_role_arn,
            file_asset_publishing_role_arn=context.deploy_role_arn,
            image_asset_publishing_role_arn=context.deploy_role_arn,
            lookup_role_arn=context.deploy_role_arn,
            cloud_formation_execution_role=context.execution_role_arn,
        ),
    )
```

## Create the Stack

└─ .pylintrc  
└─ app.py  
└─ cdk.json

Stack name	Status
ProducerCapitalAllocationV1	CREATE_COMPLETE
ProducerCompdbExecutionEventsV1	CREATE_COMPLETE
ProducerFeedExecutionEventsV1	CREATE_COMPLETE
ProducerCompdbExecutionEventsV1	CREATE_COMPLETE
ProducerCompdbExecutionEventsV1	CREATE_COMPLETE
ProducerCompdbExecutionEventsV1	CREATE_COMPLETE
ProducerCompdbExecutionEventsV1	CREATE_COMPLETE

```
def build_app(producer_deploy_config: ProducerModel) -> App:
    """Build the CDK app. Wrapped in a function for testability."""
    local_app = App()
    context = BaseSystemContext()

    EbeProducerComponentsCdkStack(
        scope=local_app,
        construct_id="EbeProducerComponentsCdkStack",
        producer_deploy_config=producer_deploy_config,
        stack_name=(
            pascalcase(context.stack_prefix)
            + pascalcase(f"Producer_{producer_deploy_config.unique_resource_id}")
        ),
        synthesizer=DefaultStackSynthesizer(
            deploy_role_arn=context.deploy_role_arn,
            file_asset_publishing_role_arn=context.deploy_role_arn,
            image_asset_publishing_role_arn=context.deploy_role_arn,
            lookup_role_arn=context.deploy_role_arn,
            cloud_formation_execution_role=context.execution_role_arn,
        ),
    )
```



## Select the L3 Pattern

> constructs

❖ \_\_init\_\_.py

❖ producer\_stack.py

```
class EbeProducerComponentsCdkStack(Stack):
    """The EbeProducerComponentsCdkStack deploys the L3 pattern for the selected ingress type."""

    def __init__(self,
                 scope: Construct,
                 construct_id: str,
                 producer_deploy_config: ProducerModel,
                 **kwargs):
        """Construct a new producer sub-assembly stack."""
        super().__init__(scope, construct_id, **kwargs)

        construct_mapping = {
            "HTTPS": HttpsProducerConstruct,
            "SQS": SqsProducerConstruct,
        }

        # Fetch the right construct based on the ingestion type in the config
        construct_class = construct_mapping[producer_deploy_config.connection_type]

        # Initialize the construct
        construct_class(
            scope=self,
            construct_id="ProducerConstruct",
            producer_deploy_config=producer_deploy_config,
        )
```

## The HTTPS L3 Pattern

❖ ebe\_alarm.py

❖ https\_producer.py

❖ postnl\_lambda\_base.py

```
class HttpsProducerConstruct(Construct):
    """The EBE Backend Producer HTTPS Components L3 Pattern."""

    def __init__(self,
                 scope: Construct,
                 construct_id: str,
                 producer_deploy_config: HttpsProducerModel,
                 **kwargs):
        """Construct a new EBE Backend Producer HTTPS sub-assembly."""
        super().__init__(scope, construct_id, **kwargs)
        system_context = HttpsSystemContext()

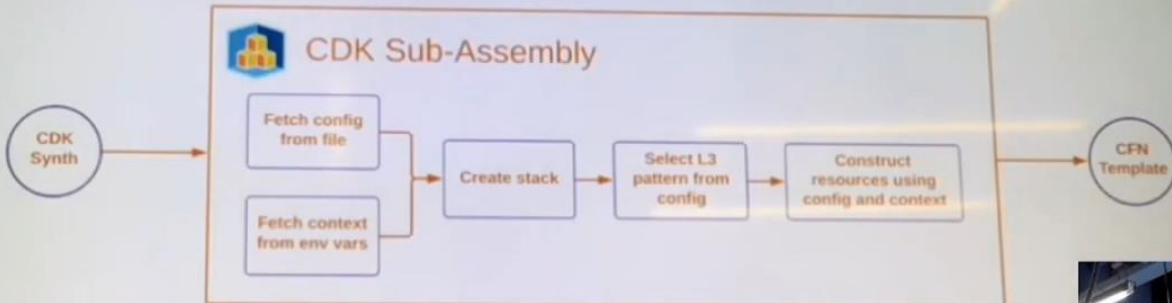
        rest_api = apigateway.RestApi.from_rest_api_attributes(
            scope=self,
            id="RestApi",
            rest_api_id=system_context.rest_api_id,
            root_resource_id=system_context.root_resource_id,
        )
```

```
request_model = apigateway.Model(
    scope=self,
    id="Model",
    rest_api=rest_api,
    content_type="application/json",
    schema=apigateway.JsonSchema(),
)

# Need to set model directly to prevent issues with schema conversion
cfn_model: apigateway.CfnModel = request_model.node.default_child
cfn_model.schema = producer_deploy_config.data_json_schema

request_validator = apigateway.RequestValidator(
    scope=self,
    id="Validator",
    rest_api=rest_api,
    validate_request_body=True,
    validate_request_parameters=False,
)
```

## CDK Sub-Assembly Recap



## Local Deployment

```
(.venv) ➤ ebe-producer-components-cdk git:(feature/cdk_cleanup) # cdk deploy --all --require-approval never --profile PostNL-ENG-Dev
> Synthesis time: 9.29s

EbeProducerComponentsCdkStack (LucProducerDemoTruckFlatTireV0): deploying...
[0%] start: Publishing 69ddcd2a4a486fbc17af0f469a8c7af181248d5392a44ebd67a7684884c83c6:current_account-current_region
[100%] success: Published 69ddcd2a4a486fbc17af0f469a8c7af181248d5392a44ebd67a7684884c83c6:current_account-current_region
LucProducerDemoTruckFlatTireV0: creating CloudFormation changeset...
[.....] (1/6)

2:35:48 PM | CREATE_IN_PROGRESS | AWS::CloudFormation::Stack | LucProducerDemoTruckFlatTireV0
2:35:52 PM | CREATE_IN_PROGRESS | AWS::ApiGateway::Resource | EbeProducerCompos...truck_flat_tire_v0
2:35:54 PM | CREATE_IN_PROGRESS | AWS::ApiGateway::RequestValidator | EbeProducerCompos...onstruct/Validator
2:35:54 PM | CREATE_IN_PROGRESS | AWS::ApiGateway::Model | EbeProducerCompos...cerConstruct/Model

✔ EbeProducerComponentsCdkStack (LucProducerDemoTruckFlatTireV0)

> Deployment time: 23.45s

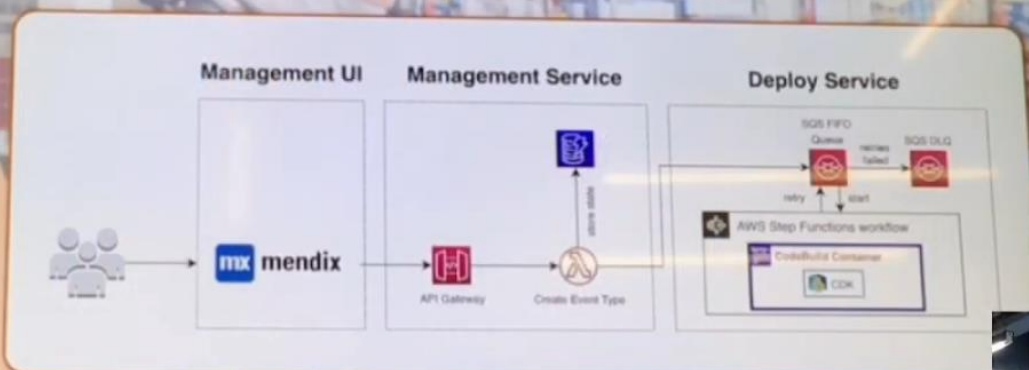
Stack ARN:
arn:aws:cloudformation:eu-west-1:927525665196:stack/LucProducerDemoTruckFlatTireV0/5c8b47d8-d839-11ec-8fe9-82c1dc863d97

> Total time: 32.74s

(.venv) ➤ ebe-producer-components-cdk git:(feature/cdk_cleanup) #
```

## User-Driven Deployments

### CodeBuild





# User-Driven Deployments

```
"version": "0.2",
"phases": {
  "install": {
    "commands": [
      "npm install -g aws-cdk@2",
    ],
  },
  "pre_build": {
    "commands": [
      "echo 'Fetching producer sub-assembly artifact'",
      "aws s3 cp s3://$ARTIFACT_BUCKET/producer_components/$PRODUCER_COMPONENTS_VERSION.zip .",
      "unzip -o $PRODUCER_COMPONENTS_VERSION.zip",
      "echo 'Fetching producer configuration'",
      "aws s3 cp s3://$ARTIFACT_BUCKET/producer_config_payloads/$STATE_MACHINE_NAME/producer_config/deploy.json",
      "echo 'Installing requirements'",
      "pip install -r requirements.txt",
    ],
  },
  "build": {
    "commands": [
      "cdk deploy --require-approval never",
    ],
  },
}
```

# User-Driven Deployments

```
"version": "0.2",
"phases": {
  "install": {
    "commands": [
      "npm install -g aws-cdk@2",
    ],
  },
  "pre_build": {
    "commands": [
      "echo 'Fetching producer sub-assembly artifact'",
      "aws s3 cp s3://$ARTIFACT_BUCKET/producer_components/$PRODUCER_COMPONENTS_VERSION.zip .",
      "unzip -o $PRODUCER_COMPONENTS_VERSION.zip",
      "echo 'Fetching producer configuration'",
      "aws s3 cp s3://$ARTIFACT_BUCKET/producer_config_payloads/$STATE_MACHINE_NAME/producer_config/deploy.json",
      "echo 'Installing requirements'",
      "pip install -r requirements.txt",
    ],
  },
  "build": {
    "commands": [
      "cdk deploy --require-approval never",
    ],
  },
}
```

## User-Driven Deployments

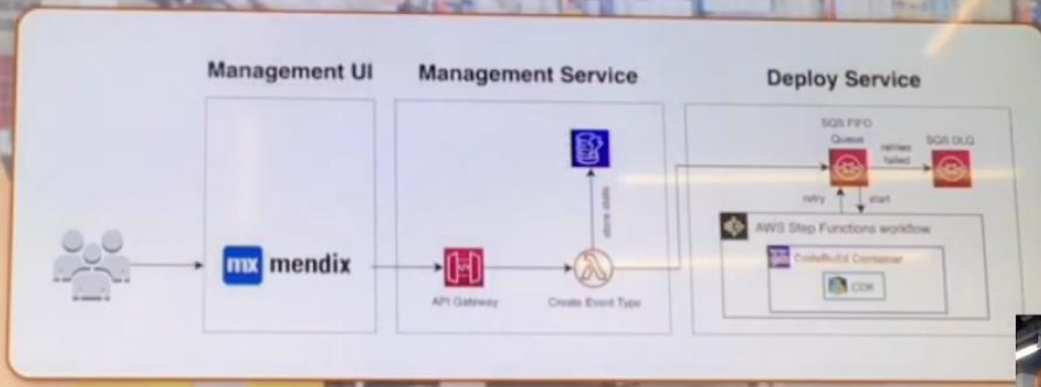
```
environment_variables=(  
    "PRODUCER_COMPONENTS_VERSION": codebuild.BuildEnvironmentVariable(value=config.producer_components_version),  
    "ARTIFACT_BUCKET": codebuild.BuildEnvironmentVariable(value=params.artifact_bucket.bucket_name),  
    "REST_API_ID": codebuild.BuildEnvironmentVariable(value=params.rest_api.rest_api_id),  
    "EVENT_BUS_ARN": codebuild.BuildEnvironmentVariable(value=params.event_bus_arn),  
    "APIGATEWAY_INGESTION_FUNCTION_ARN": codebuild.BuildEnvironmentVariable(value=params.apigateway_ingestion_function.function_arn),  
    "ROOT_RESOURCE_ID": codebuild.BuildEnvironmentVariable(value=params.rest_api.root_resource_id),  
    "STACK_PREFIX": codebuild.BuildEnvironmentVariable(value=config.stack_prefix or ""),  
    "DEPLOY_ROLE_ARN": codebuild.BuildEnvironmentVariable(value=deploy_role.role_arn),  
    "EXECUTION_ROLE_ARN": codebuild.BuildEnvironmentVariable(value=cfn_execution_role.role_arn),  
    "ENVIRONMENT": codebuild.BuildEnvironmentVariable(value=config.environment),  
)
```

## User-Driven Deployments

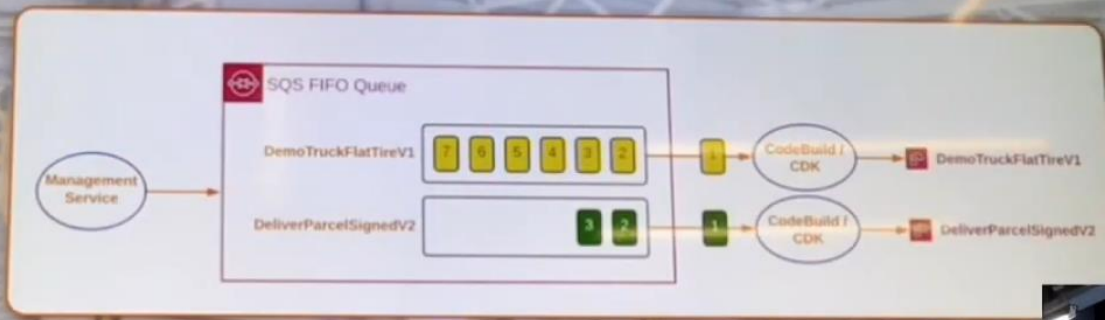
```
environment_variables=(  
    "PRODUCER_COMPONENTS_VERSION": codebuild.BuildEnvironmentVariable(value=config.producer_components_version),  
    "ARTIFACT_BUCKET": codebuild.BuildEnvironmentVariable(value=params.artifact_bucket.bucket_name),  
    "REST_API_ID": codebuild.BuildEnvironmentVariable(value=params.rest_api.rest_api_id),  
    "EVENT_BUS_ARN": codebuild.BuildEnvironmentVariable(value=params.event_bus_arn),  
    "APIGATEWAY_INGESTION_FUNCTION_ARN": codebuild.BuildEnvironmentVariable(value=params.apigateway_ingestion_function.function_arn),  
    "ROOT_RESOURCE_ID": codebuild.BuildEnvironmentVariable(value=params.rest_api.root_resource_id),  
    "STACK_PREFIX": codebuild.BuildEnvironmentVariable(value=config.stack_prefix or ""),  
    "DEPLOY_ROLE_ARN": codebuild.BuildEnvironmentVariable(value=deploy_role.role_arn),  
    "EXECUTION_ROLE_ARN": codebuild.BuildEnvironmentVariable(value=cfn_execution_role.role_arn),  
    "ENVIRONMENT": codebuild.BuildEnvironmentVariable(value=config.environment),  
)
```

[illegible][illegible]

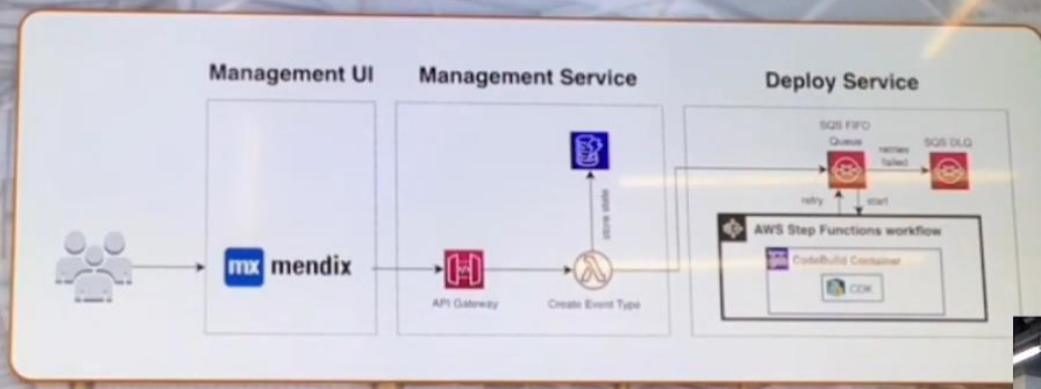
## Trigger CodeBuild



## SQS FIFO Queues

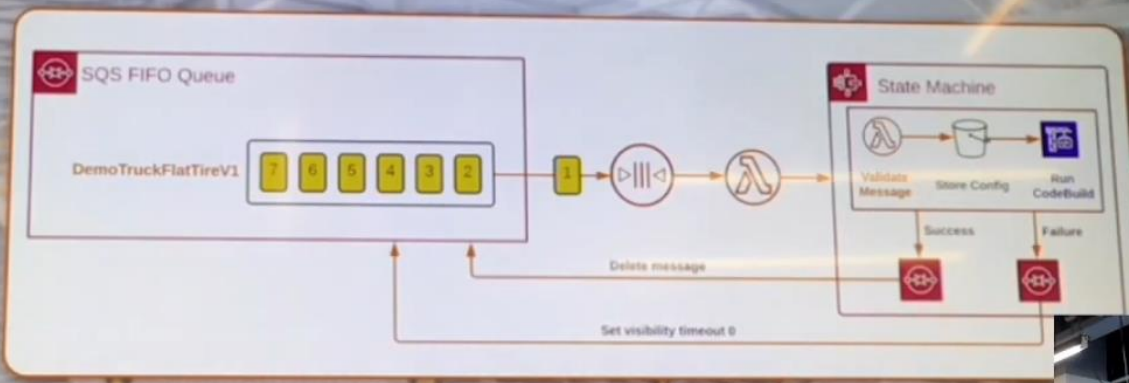


## SQS FIFO Queues

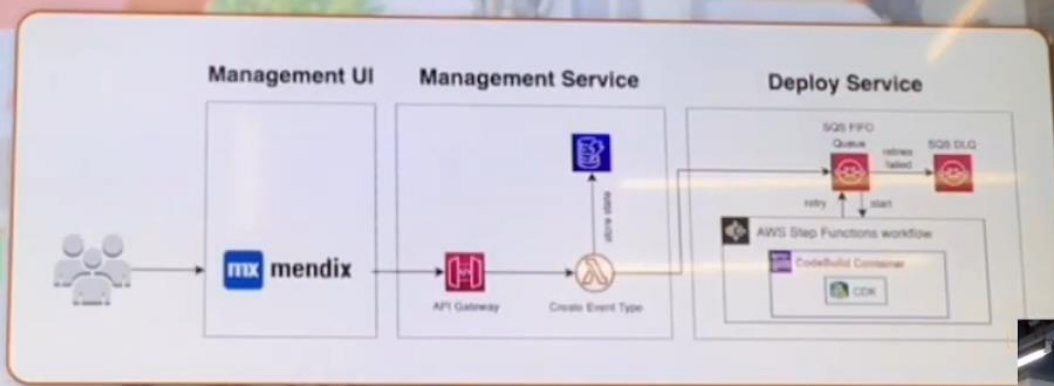




## SQS FIFO Queues



## Wrap-up



## Wrap-up

Stack name	Status
ProducerCepilalTailorationsV1	CREATE_COMPLETE
ProducerOompdddaExecutioneventsV2	CREATE_COMPLETE
ProducerFsnExecutioneventV1	CREATE_COMPLETE
ProducerOompdddaExecutioneventsV1	CREATE_COMPLETE
ProducerOompdddaExecutioneventsddv1	CREATE_COMPLETE
ProducerOompddprtemrouteupdateV1	CREATE_COMPLETE
ProducerOompddPrtemrouteupdatesV1	CREATE_COMPLETE

Physical ID	Type	Status
90147340-e287-11e1-988b-0216261806b7	AWS::CDK::Metadata	CREATE_COMPLETE
ProdutProdu6l8t5UvH2nK	AWS::ApiGateway::Model	CREATE_COMPLETE
w4e4cl	AWS::ApiGateway::Resource	CREATE_COMPLETE
Proda-Prada-10TOWAT15XCPH	AWS::ApiGateway::Method	CREATE_COMPLETE
prwyrnk	AWS::ApiGateway::RequestValidator	CREATE_COMPLETE

