superwerker

Deep Dive into the open-source superwerker AWS Quick Start

@ AWS Community Day DACH 2021-10-21

Soenke Ruempler - co-founder superluminar

The AWS Cloud promises faster time to market

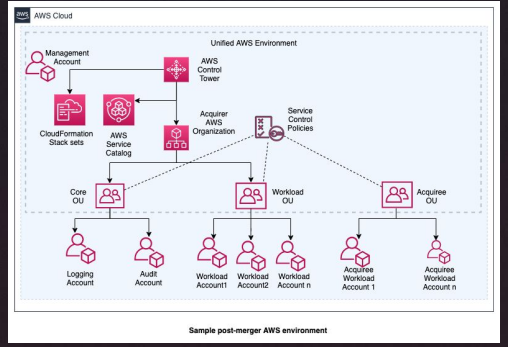
1. Idea

2. Create AWS Account

3. Start building!

But in reality ...

AWS console can lead to cognitive overload with over 200 services



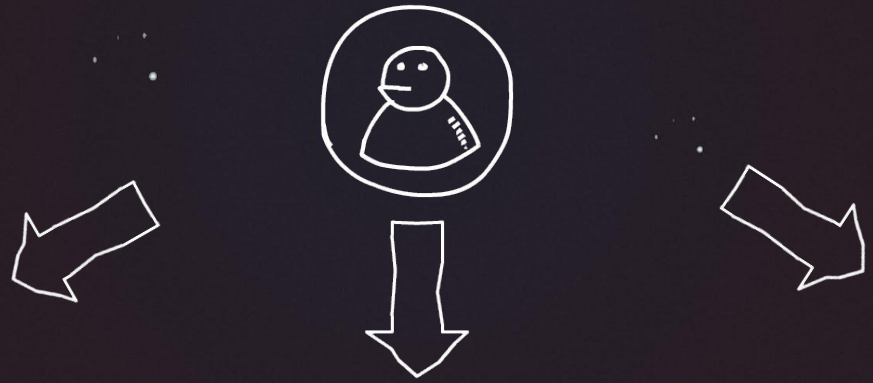
Multi-account setup recommended by AWS

Steep learning curve for AWS beginners

Information overload **without** prescriptive guidance:

white papers, documentation, Well-Architected framework

Building an AWS foundation



Pay consultants

OR

Custom-build in-house

OR

Start building right ahead

Challenges

Expensive

“Time to AWS” slowed down

Best practices not known

Insecure: Easy to overlook basic security services

THE SUPERWERKER AWS QUICK START AUTOMATES THE SETUP OF AN AWS CLOUD FOUNDATION WITH PRESCRIPTIVE BEST PRACTICES.
IT ENABLES AWS CUSTOMERS TO FOCUS ON THEIR CORE BUSINESS BY SAVING SETUP AND MAINTENANCE TIME AND MONEY.

superwerker features

superwerker enables the following AWS services and features in a **fully automated way**:

Control Tower
Base for a future-proof multi-account setup

Security Hub
org-wide ensure established security standards

GuardDuty
org-wide automatic detection of possible threats and breaches

AWS Backup
Org-wide automated backups

Budget Setup
Budget alarm

Preventive Guardrails
Protect the infrastructure

Living documentation
Guidance, next steps, standard operating procedures

SSM OpsCenter/Items
For notifications / incident response handling

Secure AWS Account Mailboxes
Dedicated secure mail domain for AWS accounts

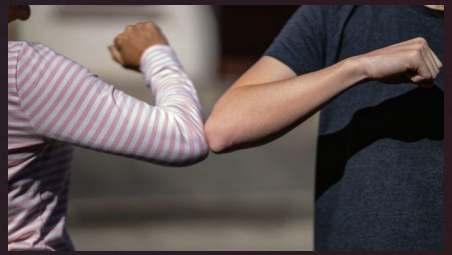
superwerker benefits

Off-the-shelf AWS experience (fully automated setup):

Secure AWS environment in one hour instead of weeks

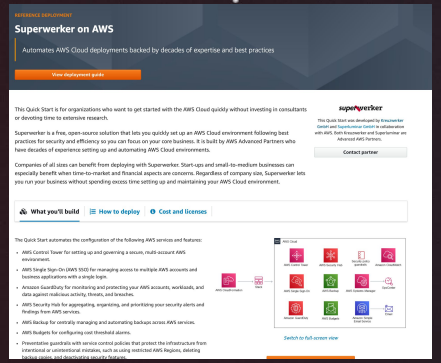
Runs in customer's AWS Account, not yet another SaaS tool

Free and open-source official AWS Quick Start



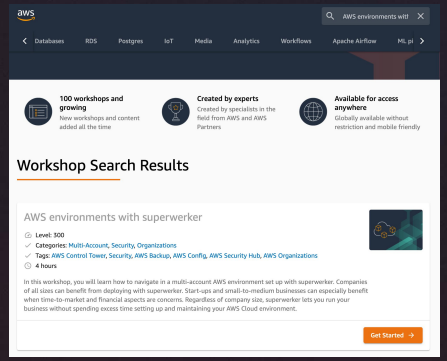
Bundled and codified experience of two AWS Advanced Partners

How to deploy superwerker



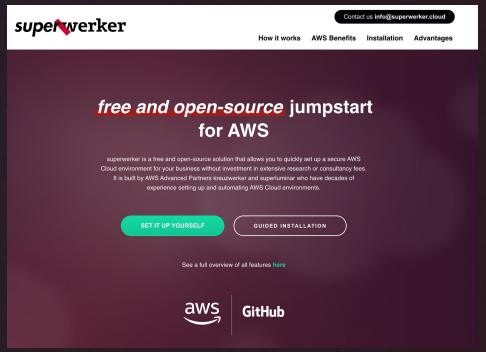
AWS Quick Start Deployment Guide

aws.amazon.com/quickstart/architecture/superwerker/



Workshop + Labs

superwerker.awsworkshop.io



Guided Installation

https://superwerker.cloud/#contact

@s0enke

superwerker

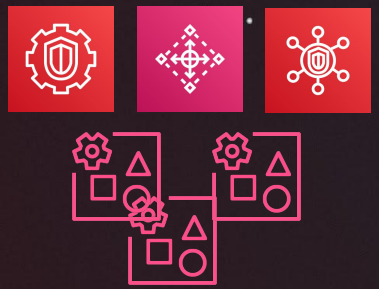
How to get help

- Guided installation / Commercial support offered by superluminar and kreuzwerker
- File issue on GitHub: github.com/superwerker/superwerker
- Architecture Decision Records: https://github.com/superwerker/superwerker/tree/main/docs/adrs
- Google Group: groups.google.com/forum/#!forum/superwerker/join
- Slack: og-aws #superwerker (Invite: http://slackhatesthe.cloud/)

Enough marketing

Deep Dive

superwerker design guidelines



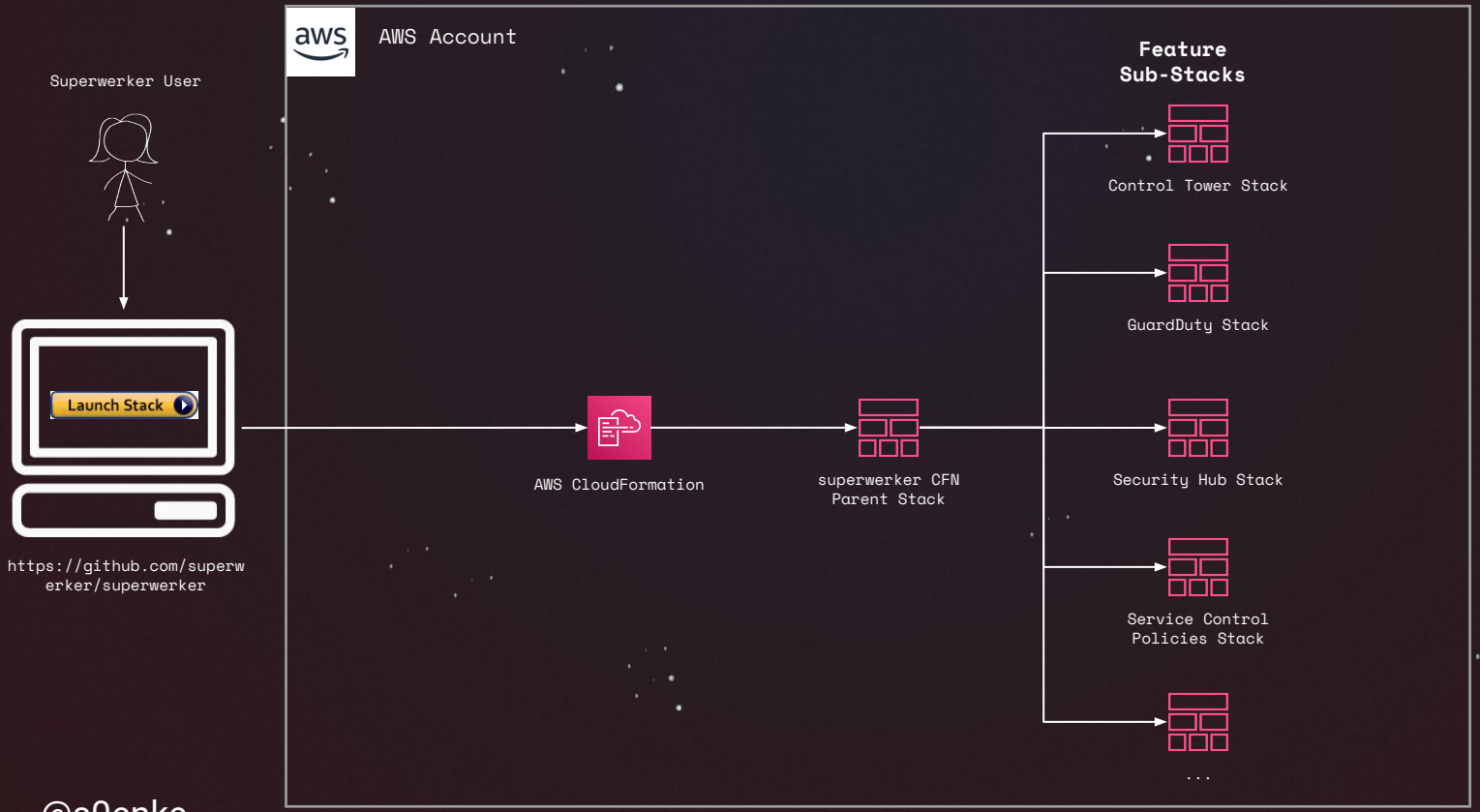
End-to-end tests with real AWS accounts and resources

Documented with Architecture Decision Records (ADR)

Low total cost of ownership: only serverless AWS services are used

Forward compatibility and adoption

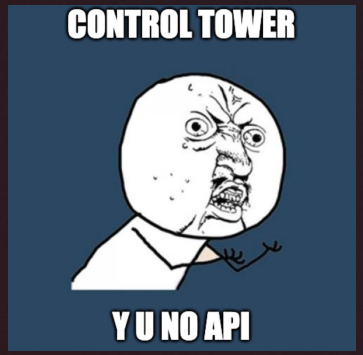
superwerker installation process



superwerker

Automating the Non-automatable

Control Tower



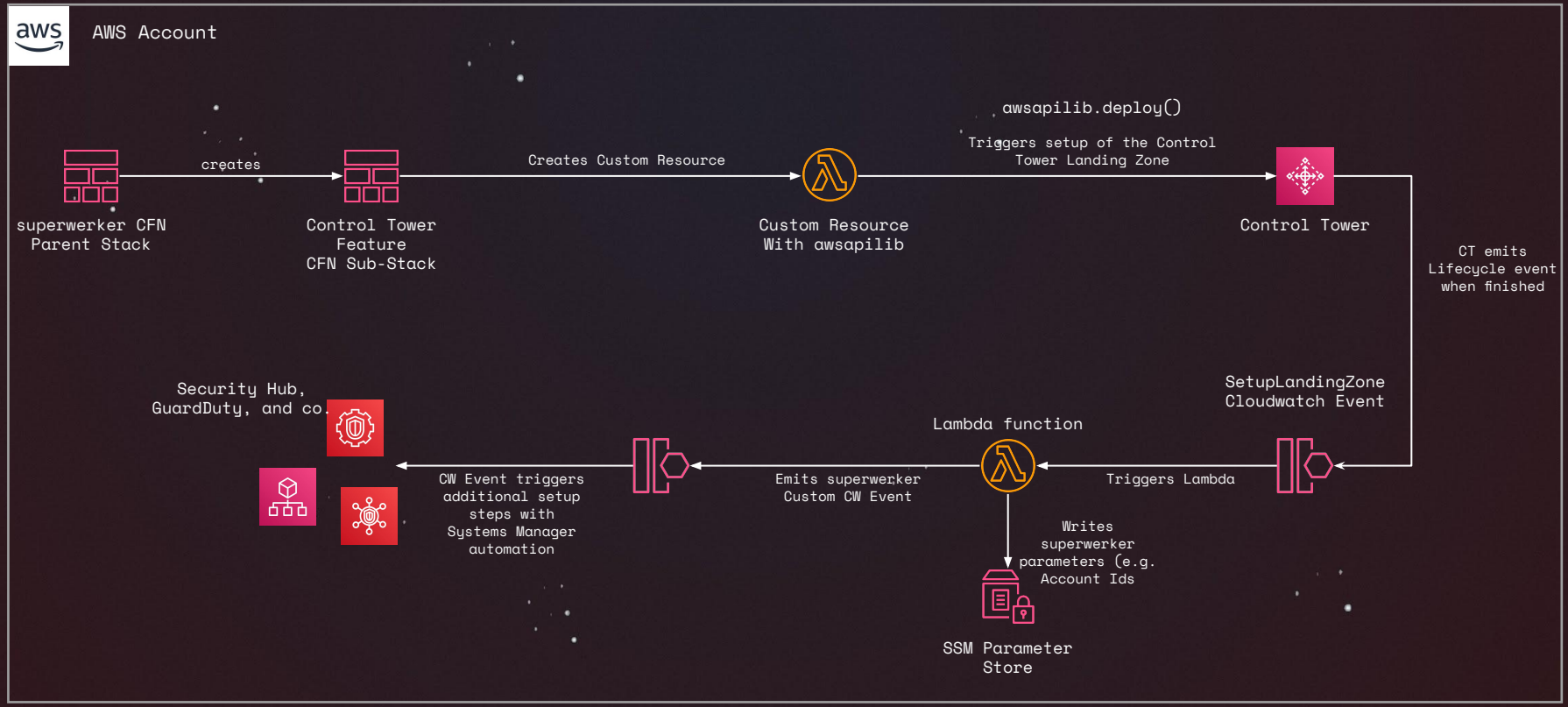
Solution 1: Automate with Cloudwatch Synthetics / Puppeteer

Solution 2: Use awsapilib

https://github.com/schubergphilis/awsapilib

```
tower.deploy(logging_account_email, security_account_email)
```

superwerker Control Tower setup



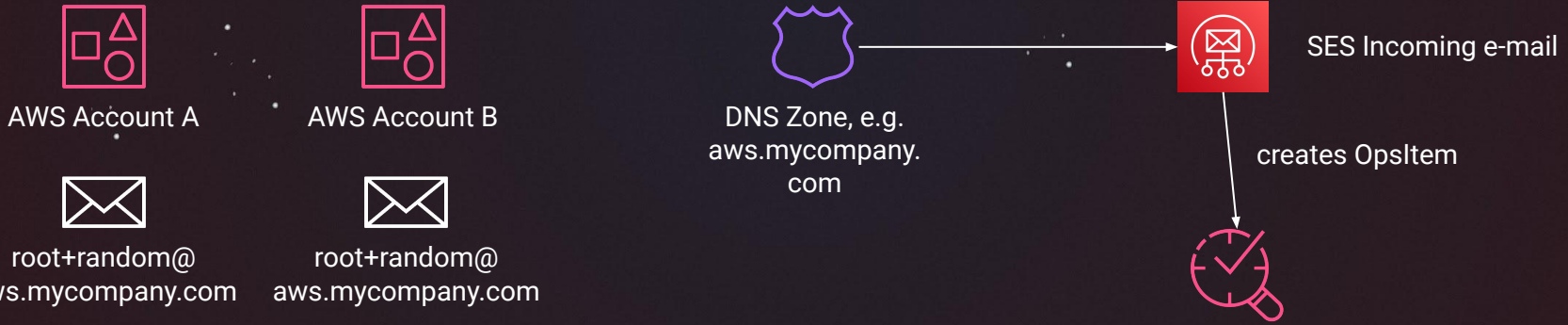
From Serverless to Functionless: SSM Automation ❤️

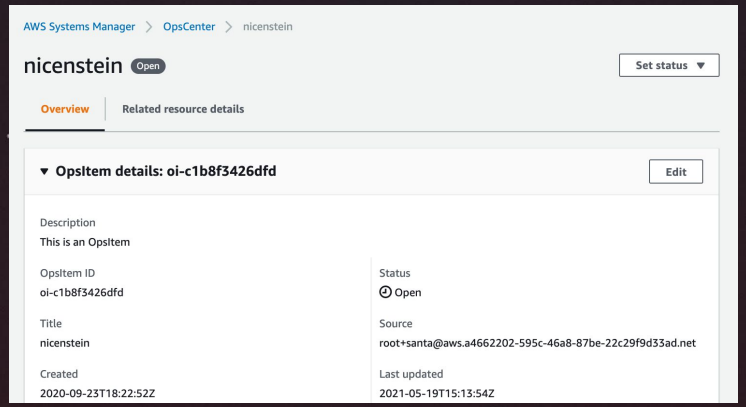
superwerker utilizes System's Manager Automation:

- Simple YAML language, no Lambdas needed
- Call arbitrary AWS APIs
- Multi-OU / Multi-account / Multi-region executions built-in
- Python runtime available for more complex tasks

```yaml
EnableGuardDutyOrganizations:
 Type: AWS::SSM::Document
 Properties:
   DocumentType: Automation
   Content:
     schemaVersion: '0.3'
     assumeRole: !GetAtt EnableGuardDutyOrganizationsRole.Arn
     parameters:
       AuditAccountId:
         type: String
         default: '{{ssm:/superwerker/account_id_audit}}'
       LogArchiveAccountId:
         type: String
         default: '{{ssm:/superwerker/account_id_logarchive}}'
     mainSteps:
       - name: CheckIfOrganizationAdminAccountIsAlReadyRegistered
         action: aws:executeAwsApi
         inputs:
           Service: guardduty
           Api: ListOrganizationAdminAccounts
         outputs:
           - Name: AdminAccountId
             Selector: $.AdminAccounts[0].AdminAccountId
         nextStep: EnableOrganizationAdminAccountChoice
```

Secure AWS root user email inboxes

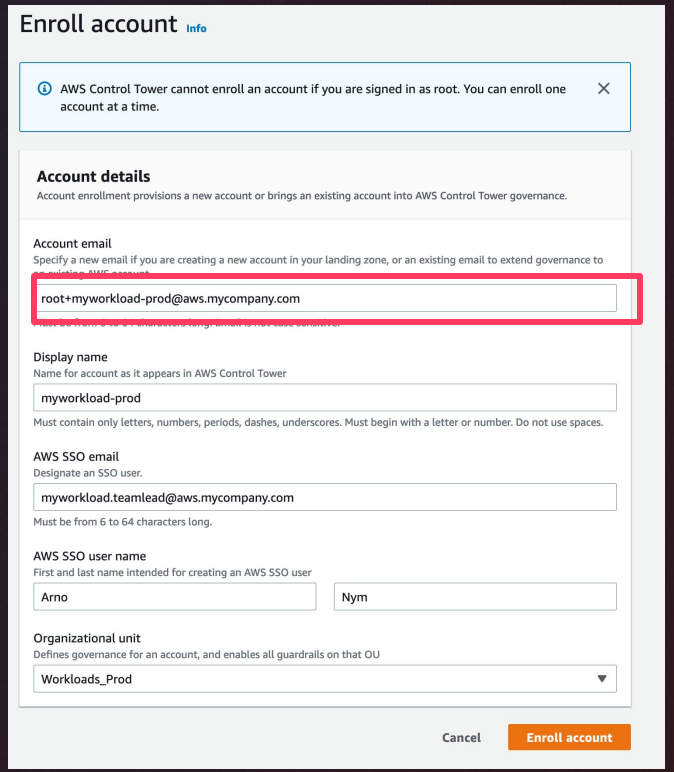




Creating new (workload) AWS Accounts

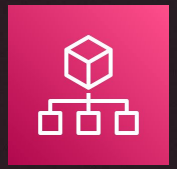
Control Tower Account Factory creates new AWS accounts

Account root user email format: root+<suffix>@ aws.mycompany.com



Testing a Landing Zone

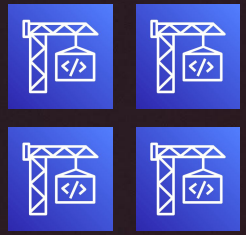
Problems with re-used AWS management accounts for testing



Cleaning up an entire AWS Organization is nearly impossible

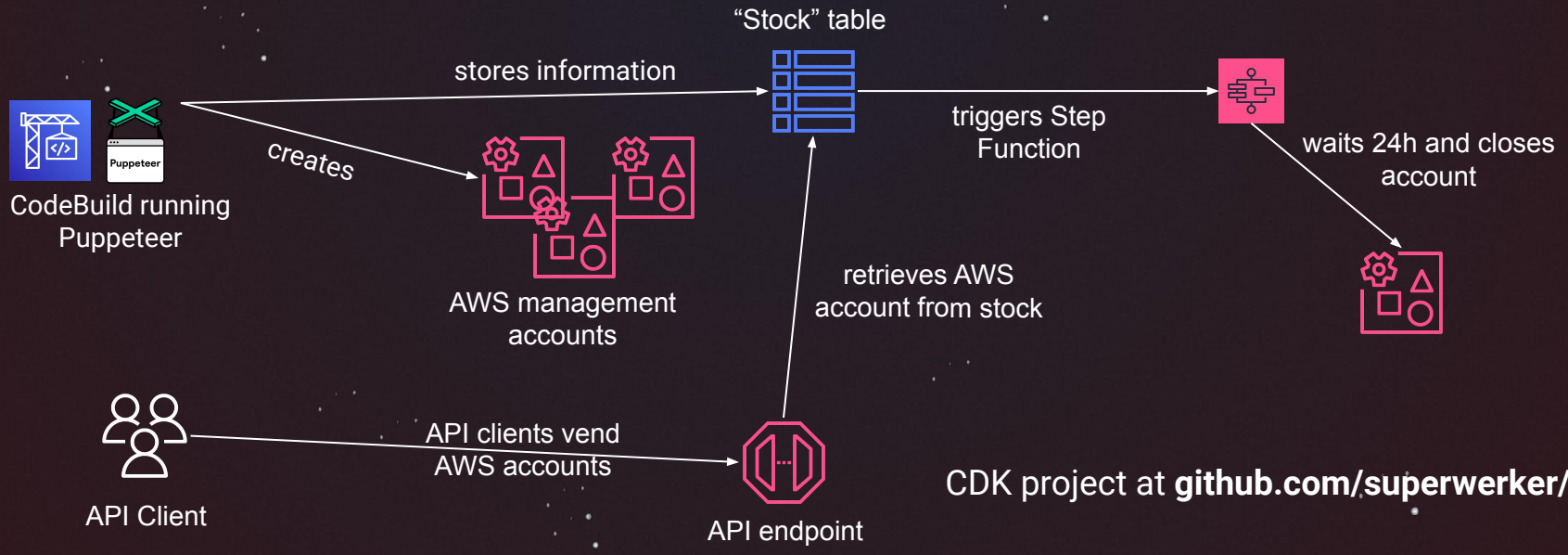
Reusing hides issues happening only in greenfield

Weird errors like “GuardDuty’s GuardDuty” block the pipeline



Batch/matrix tests hardly possible

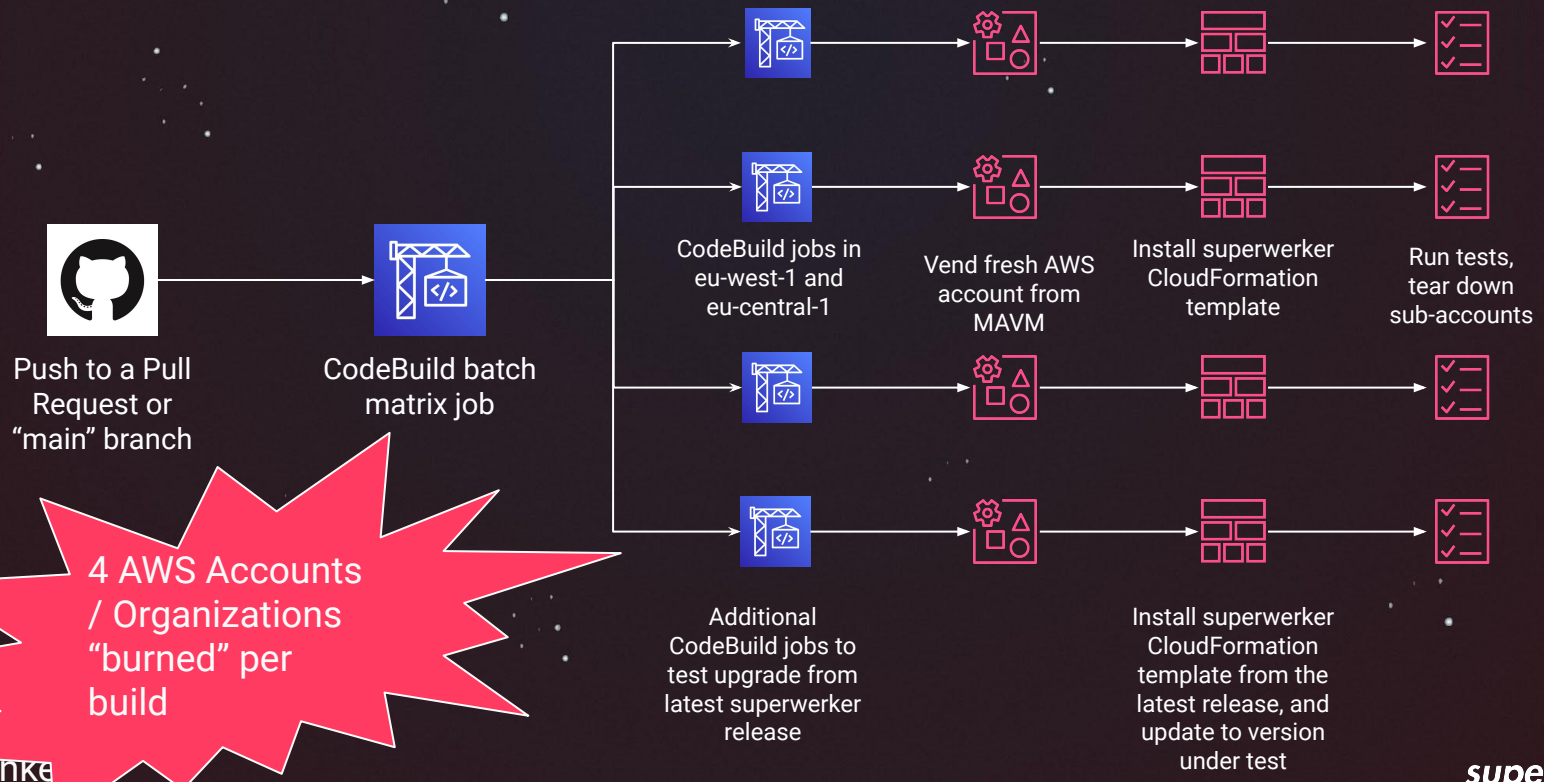
Solution: Disposable AWS Management accounts with the Management Account Vending Machine (MAVM)



CDK project at **github.com/superwerker/mavm**

```
$ curl https://xxxxxxxxx.execute-api.eu-west-1.amazonaws.com/vend
{"account_id":"123456789012","cross_account_role":"arn:aws:iam::123456789012:role/CrossAccountRole"}
```

Our current superwerker testing pipeline (powered by MAVM)



superwerker

MAVM - “Fun” with Amazon Services

- Telephone authorization done with Amazon Connect inbound feature
- Audio Captcha solving with Amazon Transcribe
- Other captchas solved with 2Captcha
- Credit Card 3D-Secure solved with SMS + Inbound SMS Gateway (sms77.io)
  - Tried Amazon Pinpoint, but this blocks messages containing TANs
- Each test runs costs money, about 1 EUR / run

Standing on the shoulders of giants: Special thanks goes to Ian McKay and his https://github.com/iann0036/aws-account-controller

About

superwerker is a joint-venture of **AWS Advanced Consulting Partners** kreuzwerker and superluminar.

Soenke Ruempler

Co-founder superluminar

soenke.ruempler@superluminar.io

twitter.com/s0enke
github.com/s0enke

Interested in working on superwerker? Apply at jobs@superluminar.io

Thanks!

Time for Q&A!