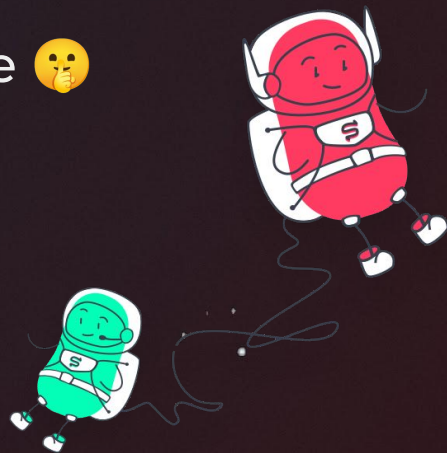




Building Infrastructure: **AWS CDK vs Terraform**

... and why CDK might be your choice 🤔

Nora Schöner | superluminar
AWS Summit Berlin 2023





COME COME,
LET'S TALK!



superluminar



What's in it for you...

- ✓ Know the **pain**
- ✓ Know the **treatment**
- ✓ Getting decision **guidance**

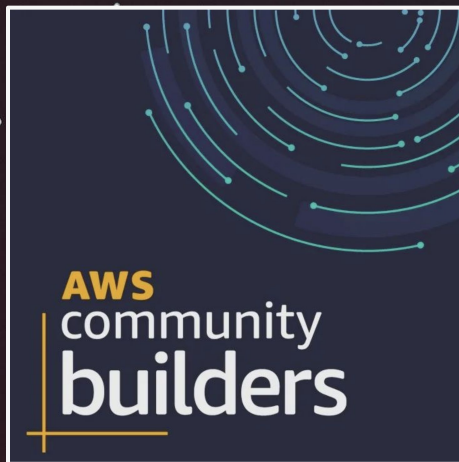
...and what not

✗ Terraform bashing

✗ Live Coding

✗ Terraform or CDK 101

Hi, I'm Nora 🖐️



初めまして、よろしくお願ひします。



@wolkencode

Infrastructure as
Code is cool!
But, which tool
should we use?

Er...
yeah, let
me check

Terraform ?

AWS CDK ?



photo by CodeCamp:N GmbH

Common Pain Points





Conditions

```
resource "aws_acm_certificate" "cert" {  
  count = (  
    (local.is_customer_1 && local.is_prod) ||  
    (local.is_customer_2 && local.is_dev) ?  
    0 : 1  
  )  
  ...  
}
```



Conditions

```
aws_acm_certificate.cert[0].status
```



Conditions

```
if (
  (props.customerId == '1' && props.stage == 'prod') ||
  (props.customerId == '2' && props.stage == 'dev')
){
  cert = Certificate.fromCertificateArn(...);
} else {
  cert = new Certificate(...);
}
```

Lambda Deployment





Lambda Deployment

```
data "archive_file" "lambda" {  
  type = "zip"  
  source_file = "..."  
  output_path = "..."  
}
```



```
resource "aws_lambda_function" "lambda" {  
  filename = data.archive_file.lambda.output_path  
  ...  
}
```




Lambda Deployment

```
resource "null_resource" "lambda_bundle" {  
  provisioner "local-exec" {  
    command = "/bin/bash bundle_lambda.sh"  
  }  
}
```



Lambda Deployment

```
const func = new NodejsFunction(this, 'func', {  
  ...  
  bundling: {  
    nodeModules: ['ms-teams-webhook'],  
  },  
});
```



Policy Handling

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "",
      "Effect": "Allow",
      "Action": [
        "dynamodb:*"
      ],
      "Resource": "*",
    }
  ]
}
```



Policy Handling

```
resource "aws_lambda_function" "lambda" { ... }

resource "aws_iam_role" "role" { ... }

resource "aws_iam_policy" "policy" { ... }

resource "aws_iam_role_policy_attachment" "attach" {
  role = aws_iam_role.role.name
  policy_arn = aws_iam_policy.policy.arn
}
```



Policy Handling

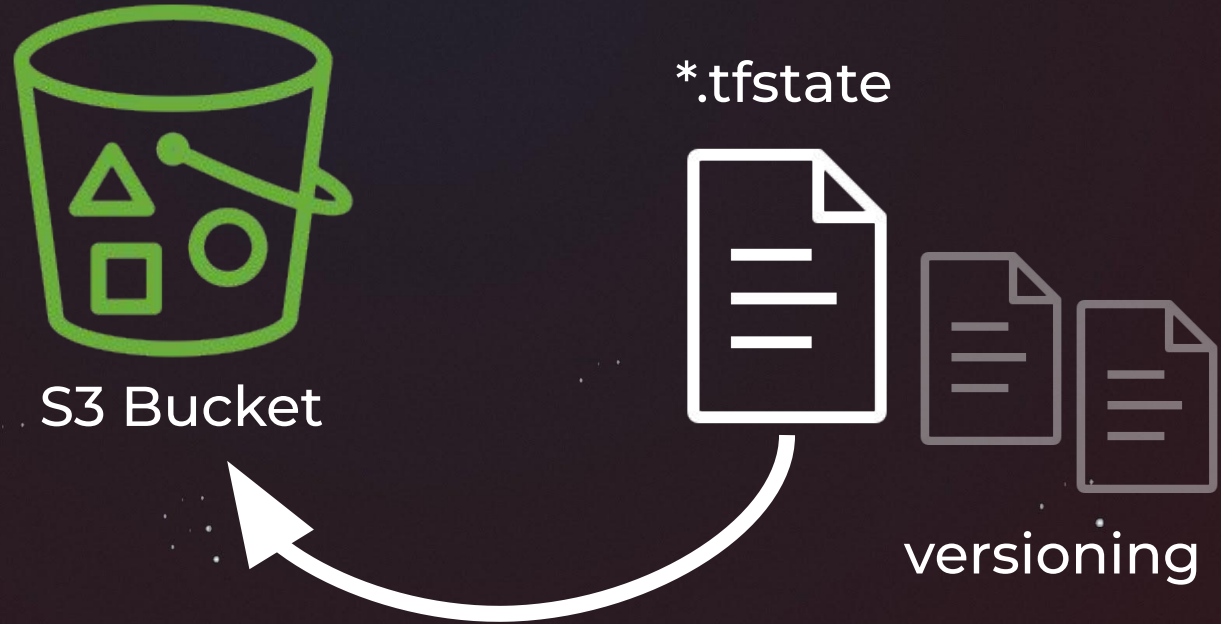
```
// table
const table = new dynamodb.Table(...);

// function
const function = new lambda.Function(...);

// adding permissions
table.grantReadWriteData(function);
```

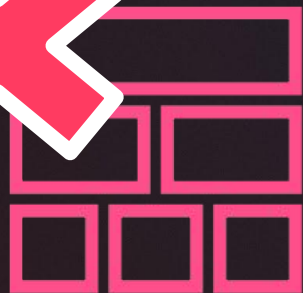


State & Rollbacks

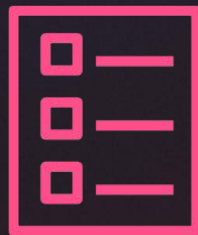




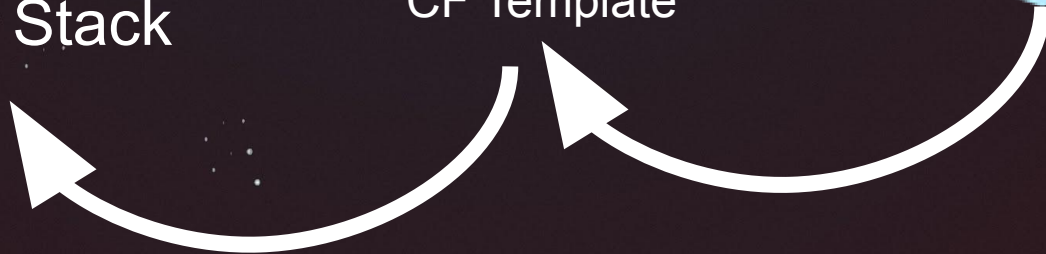
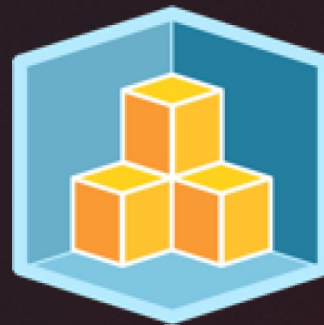
State & Rollbacks



CF Stack



CF Template





Testing

```
const template = Template.fromStack(myStack);

template.hasResourceProperties(
  "AWS::IAM::Role",
  Match.objectEquals({
    AssumeRolePolicyDocument: { ... },
  }),
);
```

Just bashing Terraform?



No programming skills needed

Easy Cross-Region Deployment

And now what?



JAKE-CLARK.TUMBLR

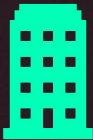
Decisions depending on...



the **team staffing**

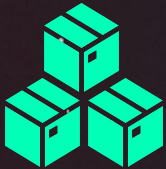


the **preferences** of the developers



the **compliance rules** of your company

Decisions depending on...



the **availability of the services** to be provisioned



if workflows are easily **automatable**



the quality of the **documentation**

The conclusion



Let's **connect**



Senior Cloud
Consultant
@superluminar



@wolkencode



Nora Schöner

<https://wolkencode.de>

