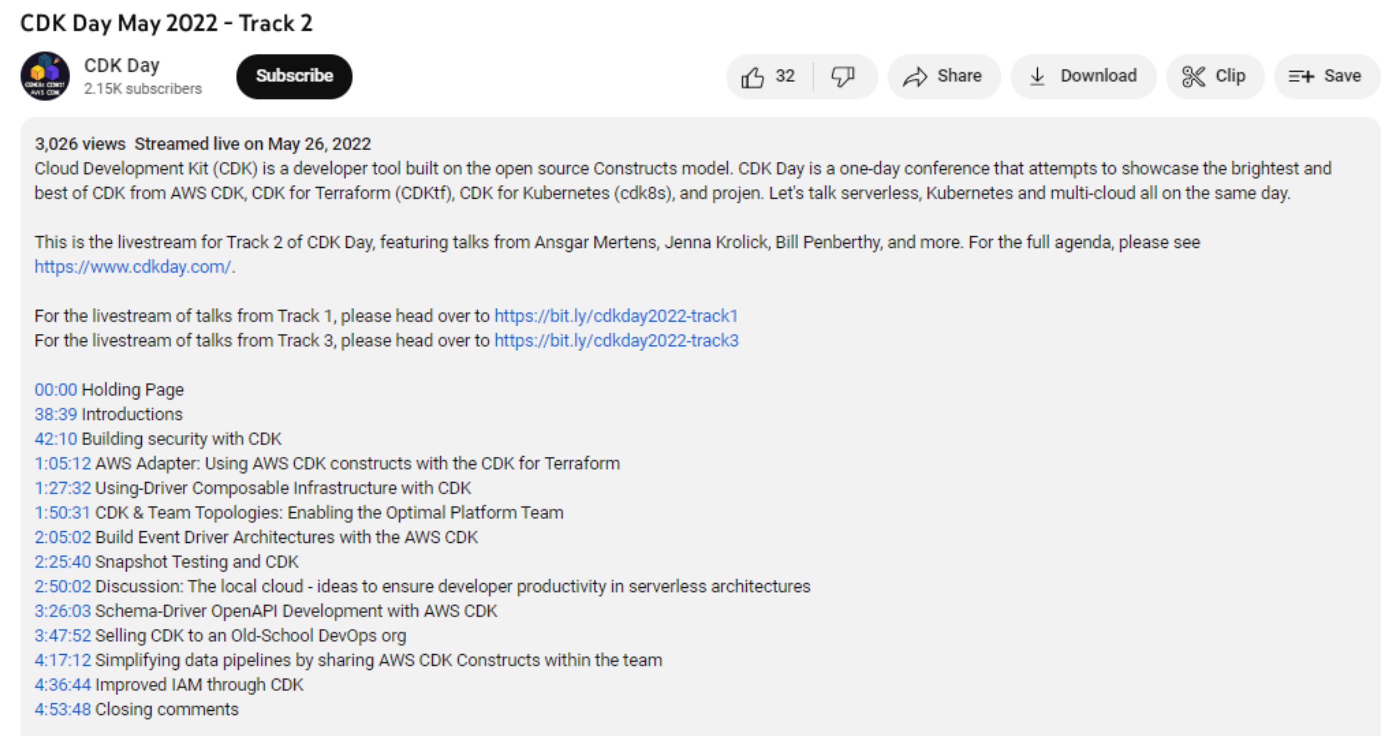


CDK Day May 2022 - Track 2

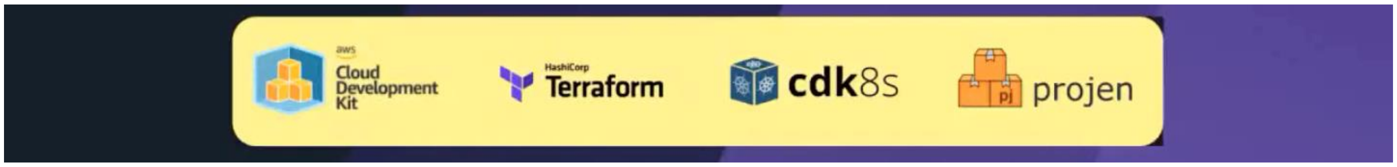
CDK Day
2.15K subscribers
Subscribe
32 Share Download Clip Save

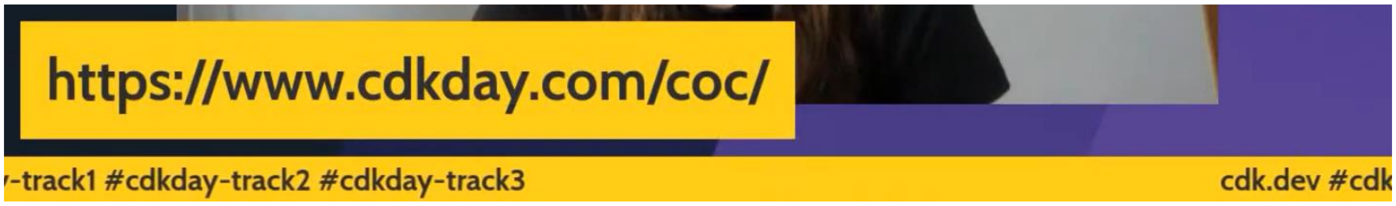
3,026 views Streamed live on May 26, 2022
Cloud Development Kit (CDK) is a developer tool built on the open source Constructs model. CDK Day is a one-day conference that attempts to showcase the brightest and best of CDK from AWS CDK, CDK for Terraform (CDKtf), CDK for Kubernetes (cdk8s), and projen. Let's talk serverless, Kubernetes and multi-cloud all on the same day.

This is the livestream for Track 2 of CDK Day, featuring talks from Ansgar Mertens, Jenna Krolick, Bill Penberthy, and more. For the full agenda, please see https://www.cdkday.com/.

For the livestream of talks from Track 1, please head over to https://bit.ly/cdkday2022-track1
For the livestream of talks from Track 3, please head over to https://bit.ly/cdkday2022-track3

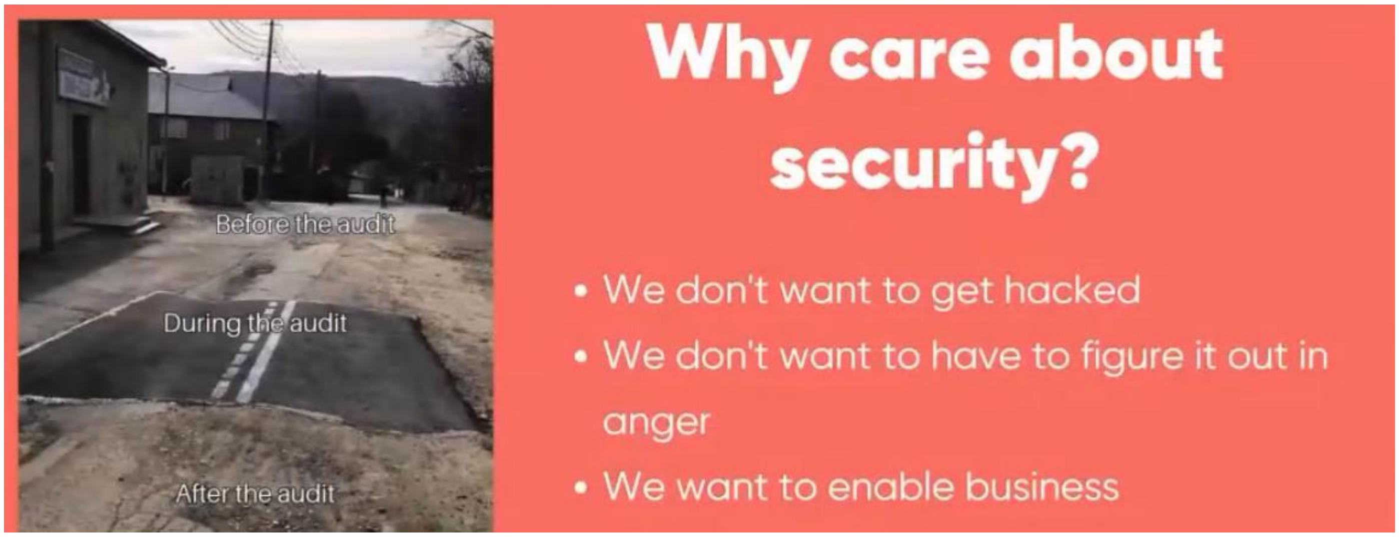
00:00 Holding Page
38:39 Introductions
42:10 Building security with CDK
1:05:12 AWS Adapter: Using AWS CDK constructs with the CDK for Terraform
1:27:32 Using-Driver Composable Infrastructure with CDK
1:50:31 CDK & Team Topologies: Enabling the Optimal Platform Team
2:05:02 Build Event Driver Architectures with the AWS CDK
2:25:40 Snapshot Testing and CDK
2:50:02 Discussion: The local cloud - ideas to ensure developer productivity in serverless architectures
3:26:03 Schema-Driver OpenAPI Development with AWS CDK
3:47:52 Selling CDK to an Old-School DevOps org
4:17:12 Simplifying data pipelines by sharing AWS CDK Constructs within the team
4:36:44 Improved IAM through CDK
4:53:48 Closing comments




https://www.cdkday.com/coc/
-track1 #cdkday-track2 #cdkday-track3
cdk.dev #cdk

Building securely with CDK

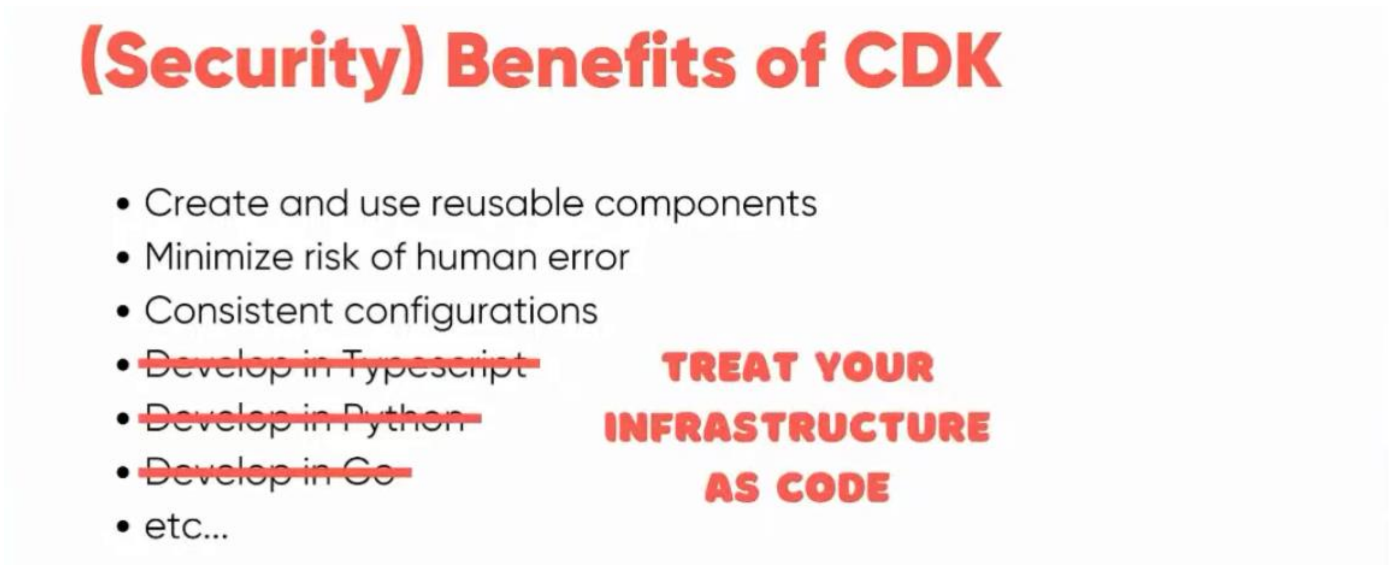
- Security consultant
- Head of product at cysense
- Honorary DevOps engineer



Why care about security?

- We don't want to get hacked
- We don't want to have to figure it out in anger
- We want to enable business

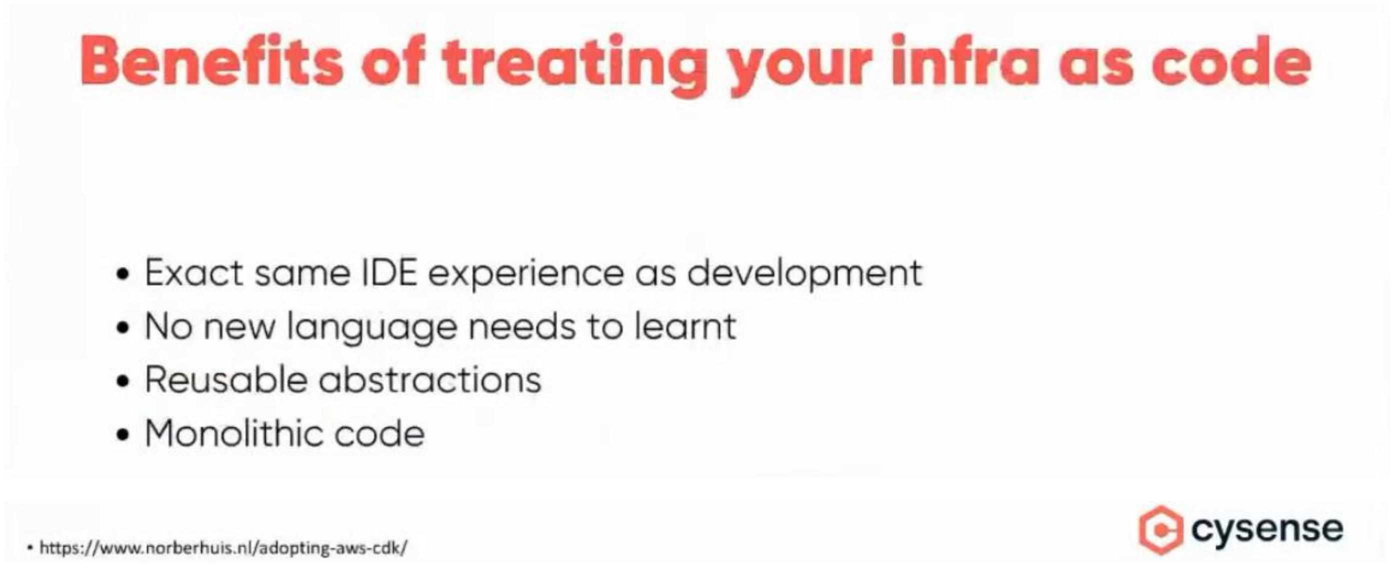
We should have security and controls in place before the hacking or breach actually happens.



(Security) Benefits of CDK

- Create and use reusable components
- Minimize risk of human error
- Consistent configurations
- ~~Develop in Typescript~~
- ~~Develop in Python~~
- ~~Develop in Go~~
- etc...

TREAT YOUR INFRASTRUCTURE AS CODE

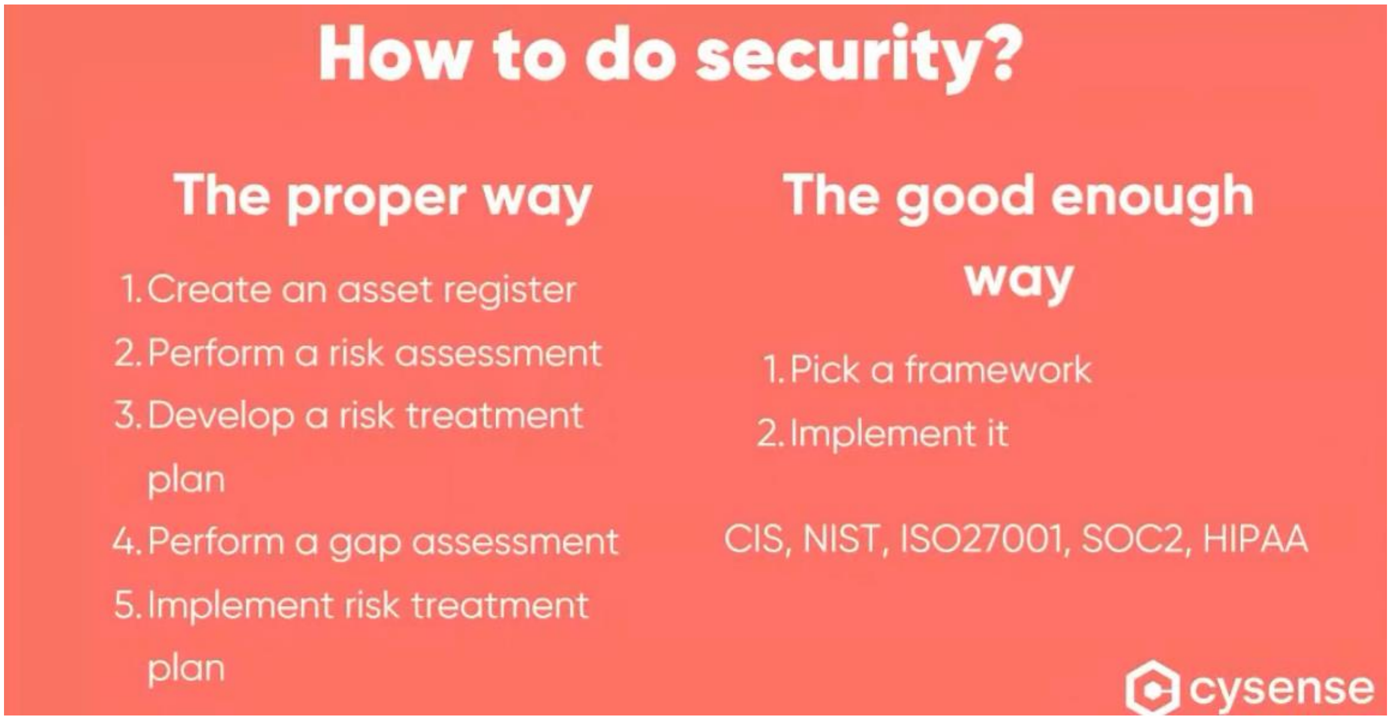


Benefits of treating your infra as code

- Exact same IDE experience as development
- No new language needs to learnt
- Reusable abstractions
- Monolithic code

• https://www.norberhuis.nl/adopting-aws-cdk/

cysense



How to do security?

The proper way

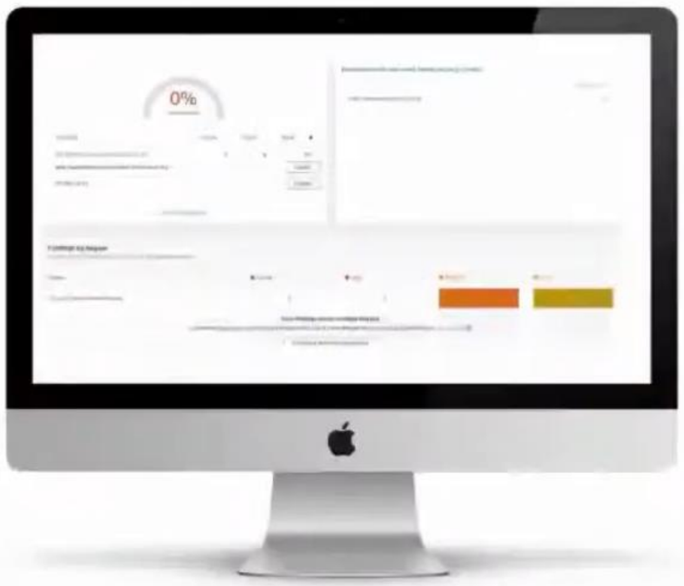
1. Create an asset register
2. Perform a risk assessment
3. Develop a risk treatment plan
4. Perform a gap assessment
5. Implement risk treatment plan

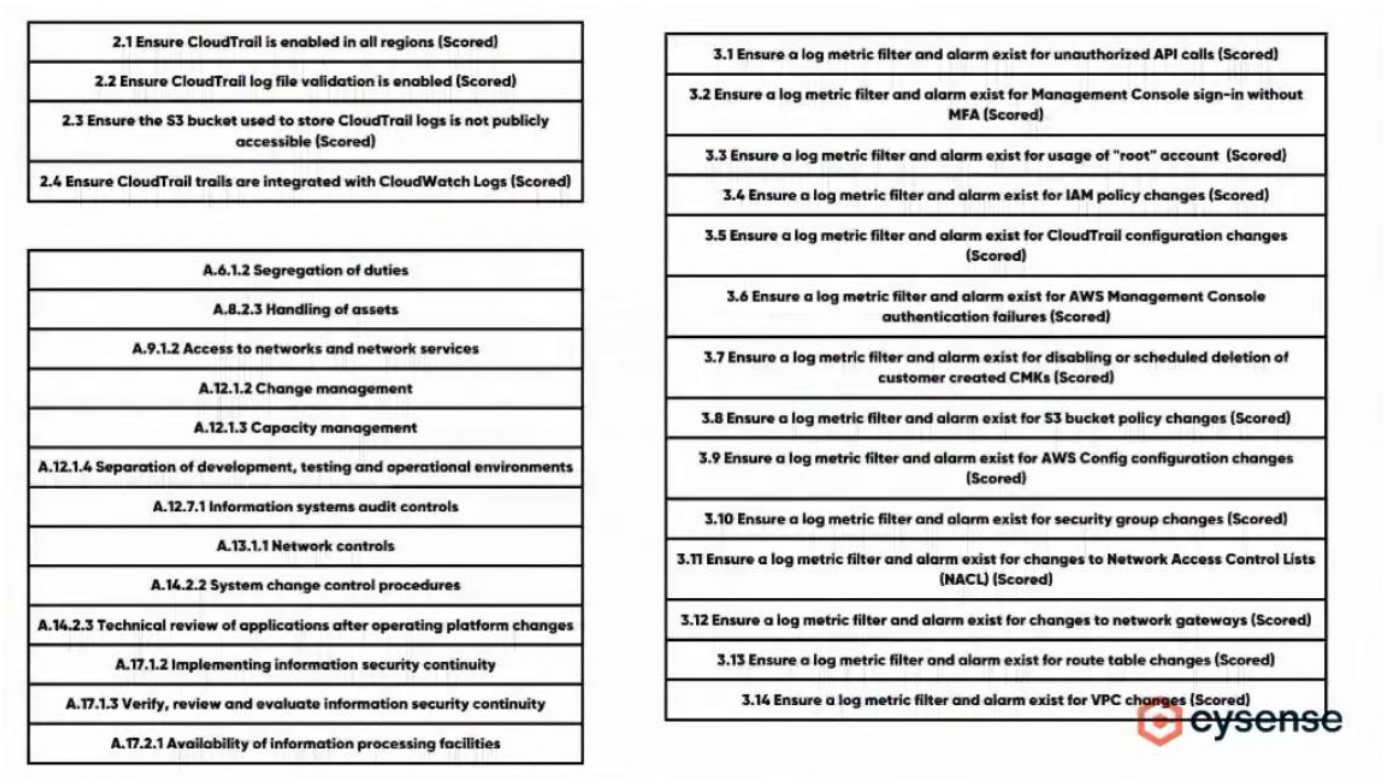
The good enough way

1. Pick a framework
2. Implement it

CIS, NIST, ISO27001, SOC2, HIPAA

cysense



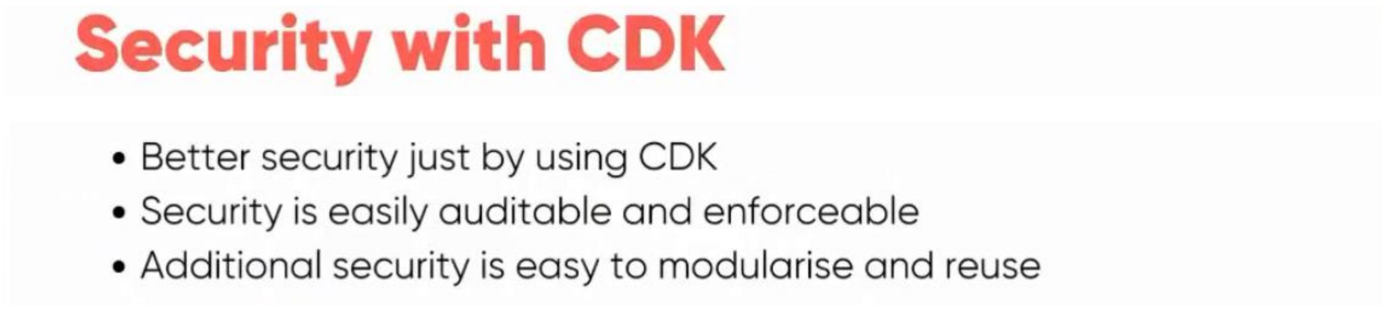


2.1 Ensure CloudTrail is enabled in all regions (Scored)
2.2 Ensure CloudTrail log file validation is enabled (Scored)
2.3 Ensure the S3 bucket used to store CloudTrail logs is not publicly accessible (Scored)
2.4 Ensure CloudTrail trails are integrated with CloudWatch Logs (Scored)

A.6.1.2 Segregation of duties
A.8.2.3 Handling of assets
A.9.1.2 Access to networks and network services
A.12.1.2 Change management
A.12.1.3 Capacity management
A.12.1.4 Separation of development, testing and operational environments
A.12.7.1 Information systems audit controls
A.13.1.1 Network controls
A.14.2.2 System change control procedures
A.14.2.3 Technical review of applications after operating platform changes
A.17.1.2 Implementing information security continuity
A.17.1.3 Verify, review and evaluate information security continuity
A.17.2.1 Availability of information processing facilities

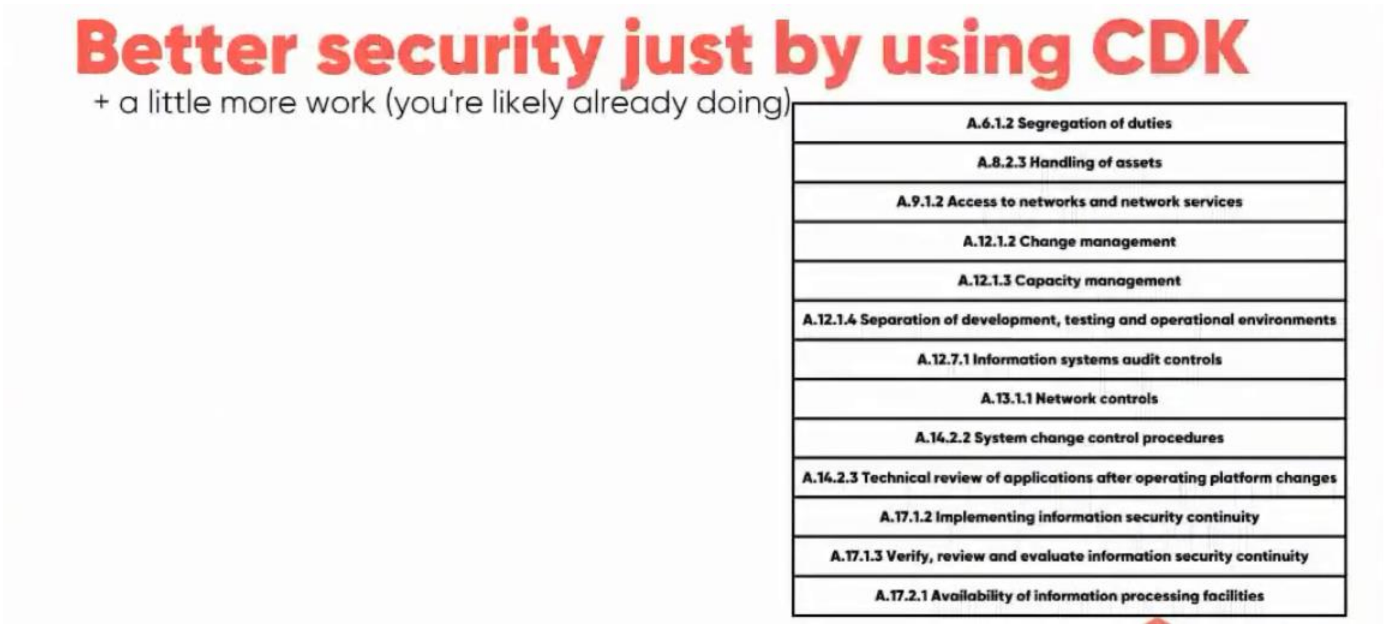
3.1 Ensure a log metric filter and alarm exist for unauthorized API calls (Scored)
3.2 Ensure a log metric filter and alarm exist for Management Console sign-in without MFA (Scored)
3.3 Ensure a log metric filter and alarm exist for usage of "root" account (Scored)
3.4 Ensure a log metric filter and alarm exist for IAM policy changes (Scored)
3.5 Ensure a log metric filter and alarm exist for CloudTrail configuration changes (Scored)
3.6 Ensure a log metric filter and alarm exist for AWS Management Console authentication failures (Scored)
3.7 Ensure a log metric filter and alarm exist for disabling or scheduled deletion of customer created CMKs (Scored)
3.8 Ensure a log metric filter and alarm exist for S3 bucket policy changes (Scored)
3.9 Ensure a log metric filter and alarm exist for AWS Config configuration changes (Scored)
3.10 Ensure a log metric filter and alarm exist for security group changes (Scored)
3.11 Ensure a log metric filter and alarm exist for changes to Network Access Control Lists (NACL) (Scored)
3.12 Ensure a log metric filter and alarm exist for changes to network gateways (Scored)
3.13 Ensure a log metric filter and alarm exist for route table changes (Scored)
3.14 Ensure a log metric filter and alarm exist for VPC changes (Scored)

cysense



Security with CDK

- Better security just by using CDK
- Security is easily auditable and enforceable
- Additional security is easy to modularise and reuse

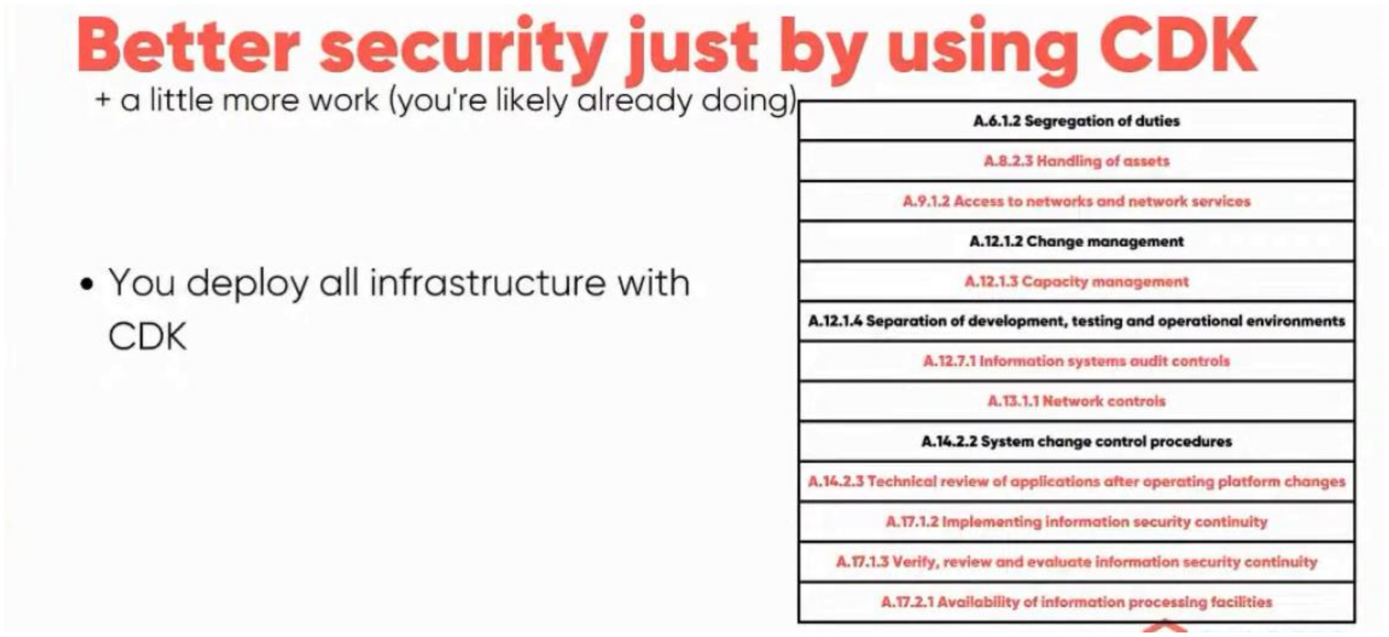


Better security just by using CDK

+ a little more work (you're likely already doing)

A.6.1.2 Segregation of duties
A.8.2.3 Handling of assets
A.9.1.2 Access to networks and network services
A.12.1.2 Change management
A.12.1.3 Capacity management
A.12.1.4 Separation of development, testing and operational environments
A.12.7.1 Information systems audit controls
A.13.1.1 Network controls
A.14.2.2 System change control procedures
A.14.2.3 Technical review of applications after operating platform changes
A.17.1.2 Implementing information security continuity
A.17.1.3 Verify, review and evaluate information security continuity
A.17.2.1 Availability of information processing facilities

These are security controls snippets from the ISO-27001 standard that we need to comply with



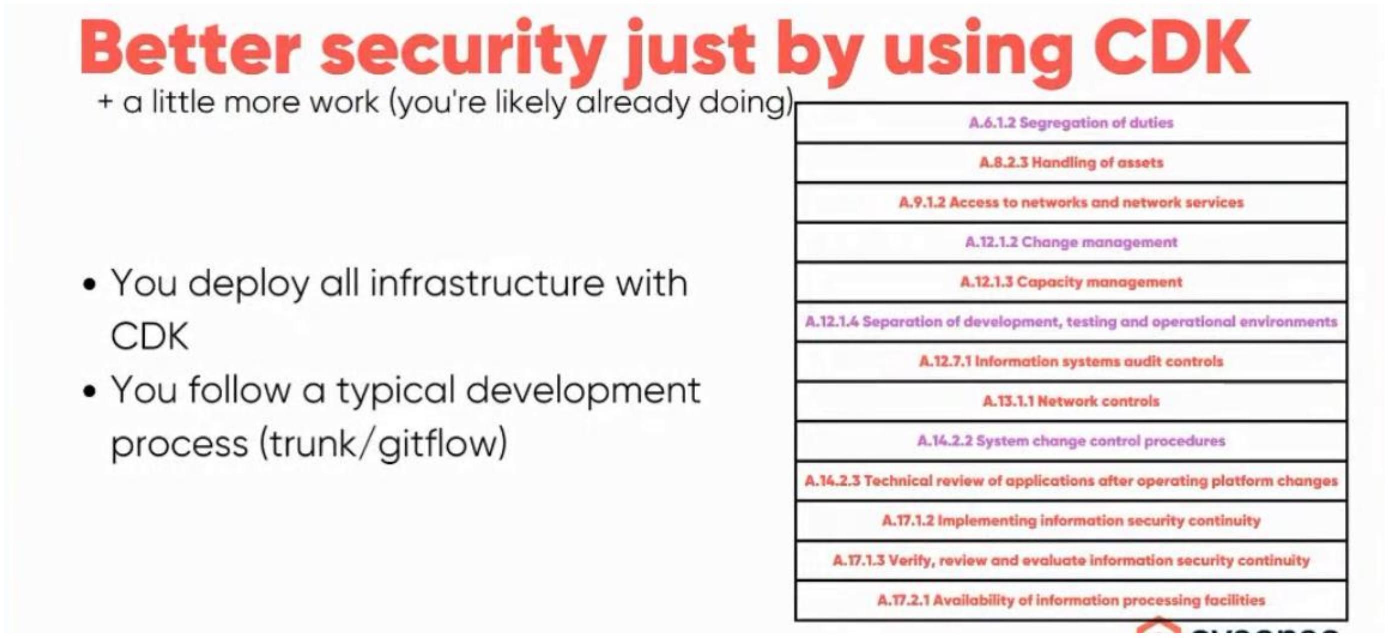
Better security just by using CDK

+ a little more work (you're likely already doing)

- You deploy all infrastructure with CDK

A.6.1.2 Segregation of duties
A.8.2.3 Handling of assets
A.9.1.2 Access to networks and network services
A.12.1.2 Change management
A.12.1.3 Capacity management
A.12.1.4 Separation of development, testing and operational environments
A.12.7.1 Information systems audit controls
A.13.1.1 Network controls
A.14.2.2 System change control procedures
A.14.2.3 Technical review of applications after operating platform changes
A.17.1.2 Implementing information security continuity
A.17.1.3 Verify, review and evaluate information security continuity
A.17.2.1 Availability of information processing facilities

If you are using the CDK, your assets have clear configuration that are logged and visible in your version control that allows you to make changes that then goes through a proper management process. CDK can be used to set up relationships between different policies, users, roles, and also to set up configurations between your networks. CDK can be used for capacity management. If you need more capacity, you can simply deploy more constructs or more resources, you can also point your CDK to another environment and deploy.



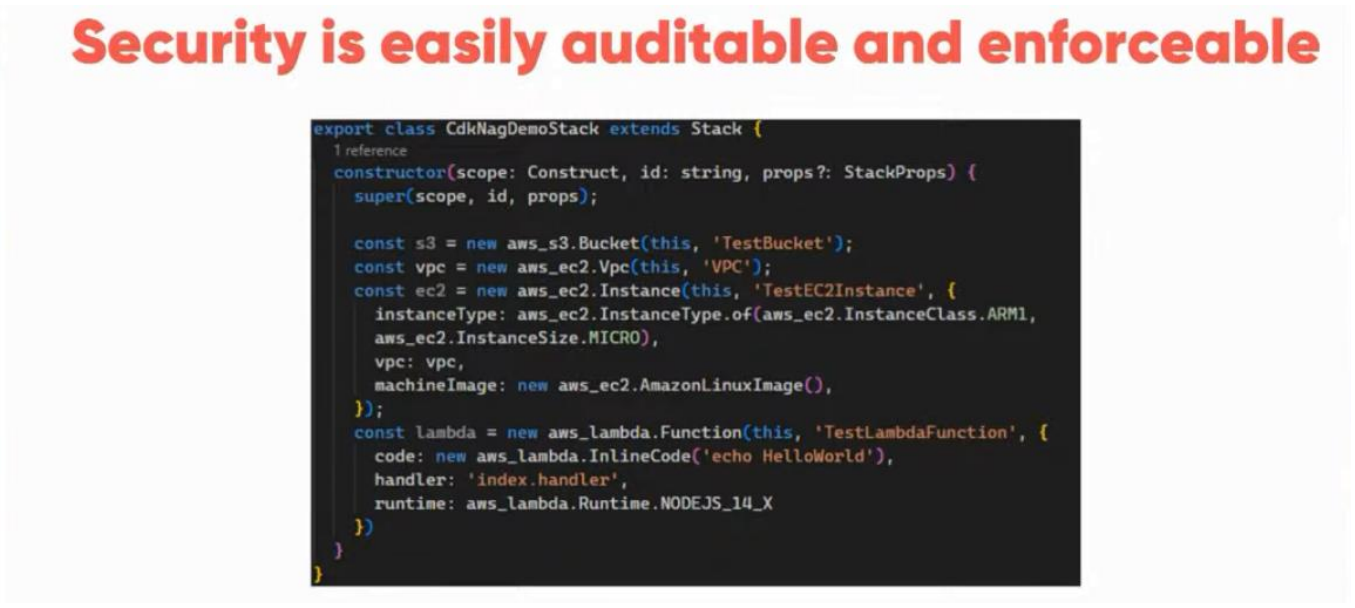
Better security just by using CDK

+ a little more work (you're likely already doing)

- You deploy all infrastructure with CDK
- You follow a typical development process (trunk/gitflow)

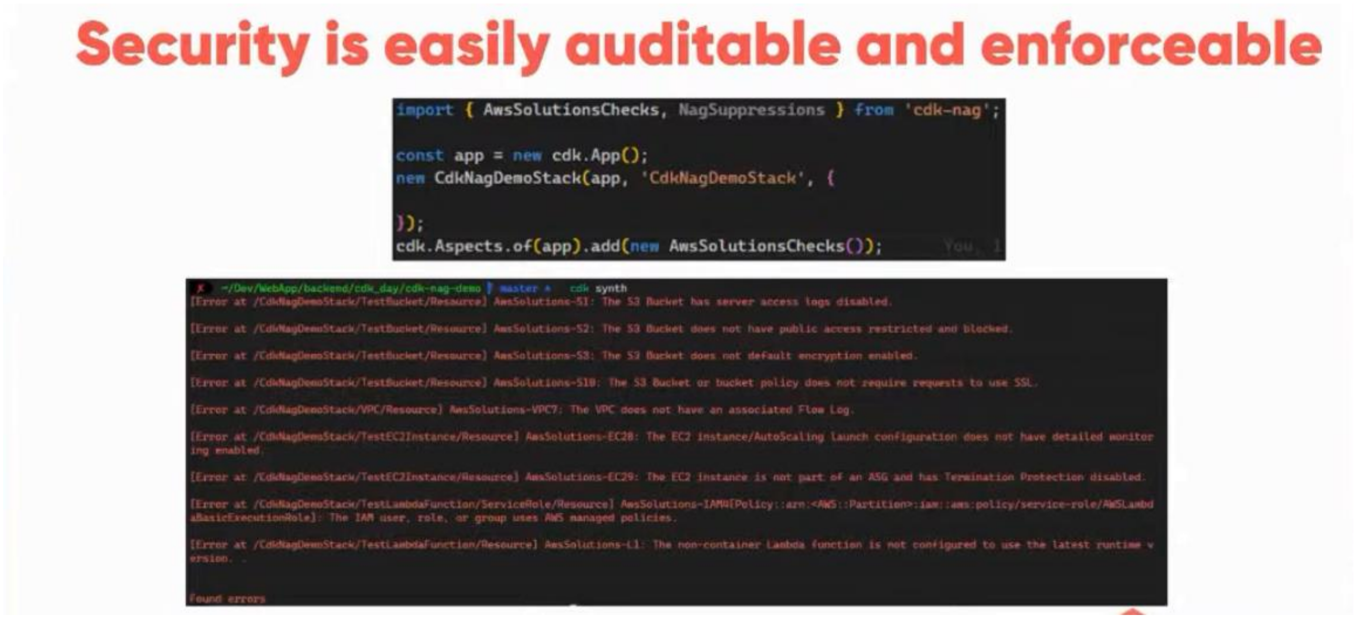
A.6.1.2 Segregation of duties
A.8.2.3 Handling of assets
A.9.1.2 Access to networks and network services
A.12.1.2 Change management
A.12.1.3 Capacity management
A.12.1.4 Separation of development, testing and operational environments
A.12.7.1 Information systems audit controls
A.13.1.1 Network controls
A.14.2.2 System change control procedures
A.14.2.3 Technical review of applications after operating platform changes
A.17.1.2 Implementing information security continuity
A.17.1.3 Verify, review and evaluate information security continuity
A.17.2.1 Availability of information processing facilities

Your CDK can follow your code development because it is also in GitHub repos, you go through the same test and audit tools for proper change management.



Security is easily auditable and enforceable

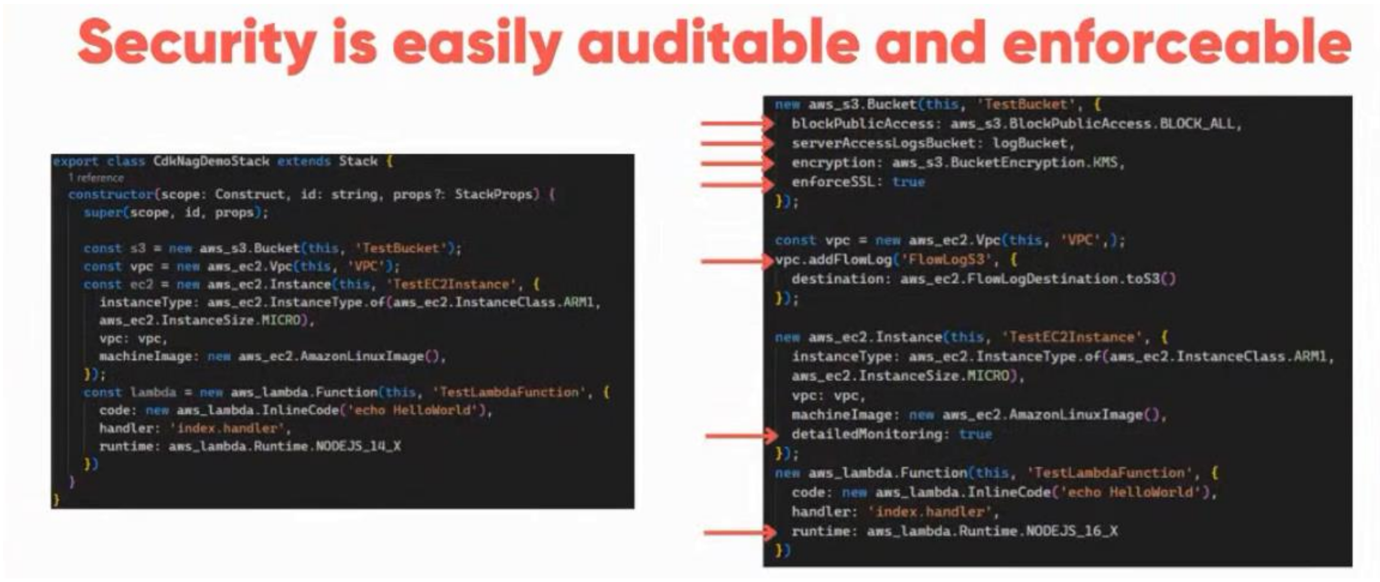
```
export class CdkNagDemoStack extends Stack {
  1 reference
  constructor(scope: Construct, id: string, props?: StackProps) {
    super(scope, id, props);

    const s3 = new aws_s3.Bucket(this, 'TestBucket');
    const vpc = new aws_ec2.Vpc(this, 'VPC');
    const ec2 = new aws_ec2.Instance(this, 'TestEC2Instance', {
      instanceType: aws_ec2.InstanceType.of(aws_ec2.InstanceClass.ARM1,
      aws_ec2.InstanceSize.MICRO),
      vpc: vpc,
      machineImage: new aws_ec2.AmazonLinuxImage(),
    });
    const lambda = new aws_lambda.Function(this, 'TestLambdaFunction', {
      code: new aws_lambda.InlineCode('echo HelloWorld'),
      handler: 'index.handler',
      runtime: aws_lambda.Runtime.NODEJS_14_X
    })
  }
}
```



Security is easily auditable and enforceable

```
import { AwsSolutionsChecks, NagSuppressions } from 'cdk-nag';

const app = new cdk.App();
new CdkNagDemoStack(app, 'CdkNagDemoStack', {

});
cdk.Aspects.of(app).add(new AwsSolutionsChecks());          You, 1
```

```
✗ ~/Dev/WebApp/backend/cdk_day/cdk-nag-demo  master ±  cdk synth
[Error at /CdkNagDemoStack/TestBucket/Resource] AwsSolutions-S1: The S3 Bucket has server access logs disabled.

[Error at /CdkNagDemoStack/TestBucket/Resource] AwsSolutions-S2: The S3 Bucket does not have public access restricted and blocked.

[Error at /CdkNagDemoStack/TestBucket/Resource] AwsSolutions-S3: The S3 Bucket does not default encryption enabled.

[Error at /CdkNagDemoStack/TestBucket/Resource] AwsSolutions-S10: The S3 Bucket or bucket policy does not require requests to use SSL.

[Error at /CdkNagDemoStack/VPC/Resource] AwsSolutions-VPC7: The VPC does not have an associated Flow Log.

[Error at /CdkNagDemoStack/TestEC2Instance/Resource] AwsSolutions-EC28: The EC2 instance/AutoScaling launch configuration does not have detailed monitor
ing enabled.

[Error at /CdkNagDemoStack/TestEC2Instance/Resource] AwsSolutions-EC29: The EC2 instance is not part of an ASG and has Termination Protection disabled.

[Error at /CdkNagDemoStack/TestLambdaFunction/ServiceRole/Resource] AwsSolutions-IAM4[Policy::arn:<AWS::Partition>:iam::aws:policy/service-role/AWSLambd
aBasicExecutionRole]: The IAM user, role, or group uses AWS managed policies.

[Error at /CdkNagDemoStack/TestLambdaFunction/Resource] AwsSolutions-L1: The non-container Lambda function is not configured to use the latest runtime v
ersion. .


Found errors
```

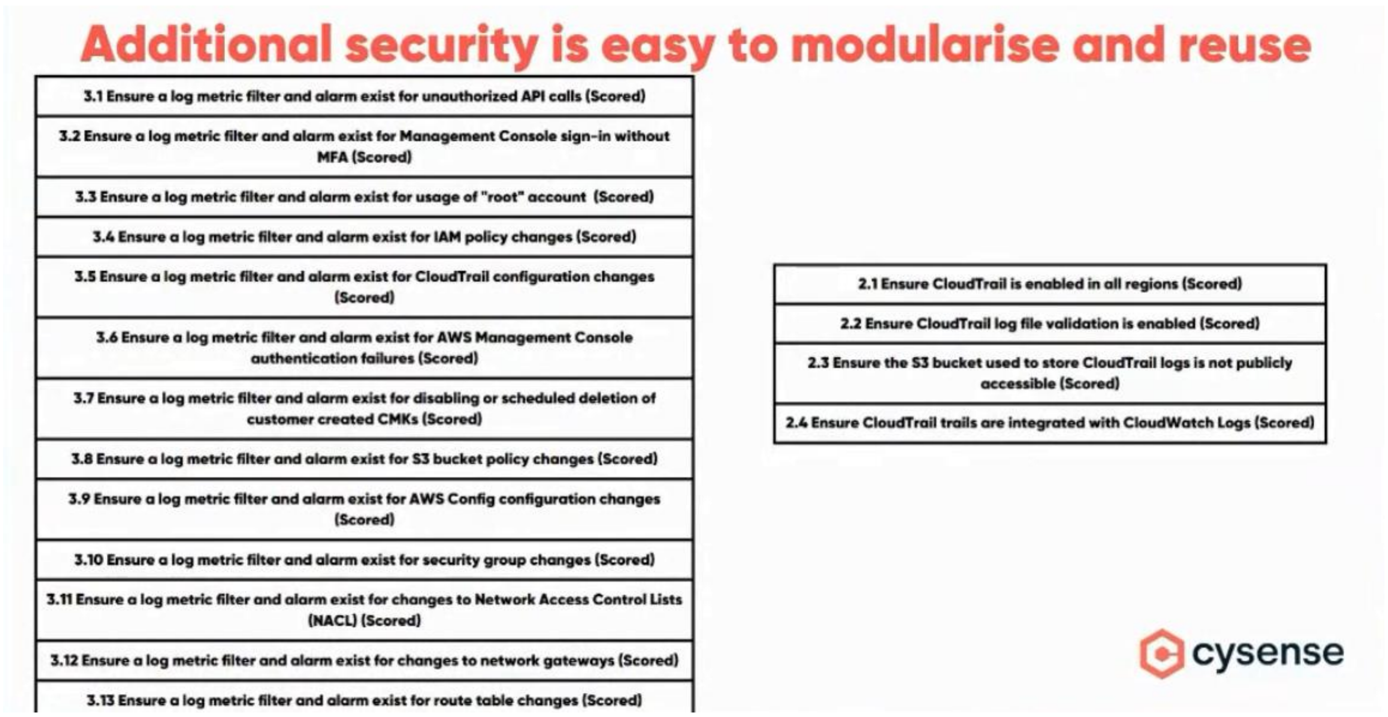


Security is easily auditable and enforceable

```
export class CdkNagDemoStack extends Stack {
  1 reference
  constructor(scope: Construct, id: string, props?: StackProps) {
    super(scope, id, props);

    const s3 = new aws_s3.Bucket(this, 'TestBucket');
    const vpc = new aws_ec2.Vpc(this, 'VPC');
    const ec2 = new aws_ec2.Instance(this, 'TestEC2Instance', {
      instanceType: aws_ec2.InstanceType.of(aws_ec2.InstanceClass.ARM1,
      aws_ec2.InstanceSize.MICRO),
      vpc: vpc,
      machineImage: new aws_ec2.AmazonLinuxImage(),
    });
    const lambda = new aws_lambda.Function(this, 'TestLambdaFunction', {
      code: new aws_lambda.InlineCode('echo HelloWorld'),
      handler: 'index.handler',
      runtime: aws_lambda.Runtime.NODEJS_14_X
    })
  }
}
```

```
new aws_s3.Bucket(this, 'TestBucket', {
  blockPublicAccess: aws_s3.BlockPublicAccess.BLOCK_ALL,
  serverAccessLogsBucket: logBucket,
  encryption: aws_s3.BucketEncryption.KMS,
  enforceSSL: true
});

const vpc = new aws_ec2.Vpc(this, 'VPC',);
vpc.addFlowLog('FlowLogS3', {
  destination: aws_ec2.FlowLogDestination.toS3()
});

new aws_ec2.Instance(this, 'TestEC2Instance', {
  instanceType: aws_ec2.InstanceType.of(aws_ec2.InstanceClass.ARM1,
  aws_ec2.InstanceSize.MICRO),
  vpc: vpc,
  machineImage: new aws_ec2.AmazonLinuxImage(),
  detailedMonitoring: true
});
new aws_lambda.Function(this, 'TestLambdaFunction', {
  code: new aws_lambda.InlineCode('echo HelloWorld'),
  handler: 'index.handler',
  runtime: aws_lambda.Runtime.NODEJS_16_X
})
```

With a little extra effort, we can implement additional security for our code highlighted by our audit tool as above.



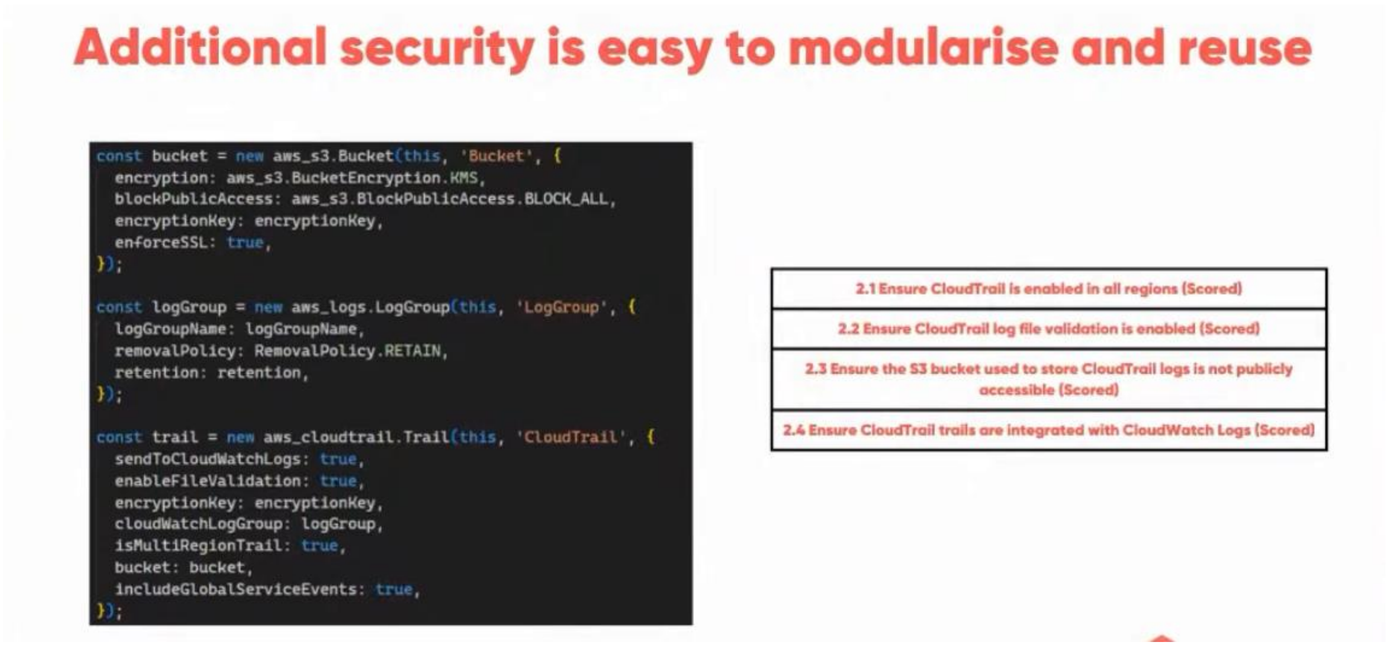
Additional security is easy to modularise and reuse

3.1 Ensure a log metric filter and alarm exist for unauthorized API calls (Scored)
3.2 Ensure a log metric filter and alarm exist for Management Console sign-in without MFA (Scored)
3.3 Ensure a log metric filter and alarm exist for usage of "root" account (Scored)
3.4 Ensure a log metric filter and alarm exist for IAM policy changes (Scored)
3.5 Ensure a log metric filter and alarm exist for CloudTrail configuration changes (Scored)
3.6 Ensure a log metric filter and alarm exist for AWS Management Console authentication failures (Scored)
3.7 Ensure a log metric filter and alarm exist for disabling or scheduled deletion of customer created CMKs (Scored)
3.8 Ensure a log metric filter and alarm exist for S3 bucket policy changes (Scored)
3.9 Ensure a log metric filter and alarm exist for AWS Config configuration changes (Scored)
3.10 Ensure a log metric filter and alarm exist for security group changes (Scored)
3.11 Ensure a log metric filter and alarm exist for changes to Network Access Control Lists (NACL) (Scored)
3.12 Ensure a log metric filter and alarm exist for changes to network gateways (Scored)
3.13 Ensure a log metric filter and alarm exist for route table changes (Scored)

2.1 Ensure CloudTrail is enabled in all regions (Scored)
2.2 Ensure CloudTrail log file validation is enabled (Scored)
2.3 Ensure the S3 bucket used to store CloudTrail logs is not publicly accessible (Scored)
2.4 Ensure CloudTrail trails are integrated with CloudWatch Logs (Scored)

cysense

In 12 hours, I managed to build a CDK Construct that implemented all the above so that my team can reuse it going forward.

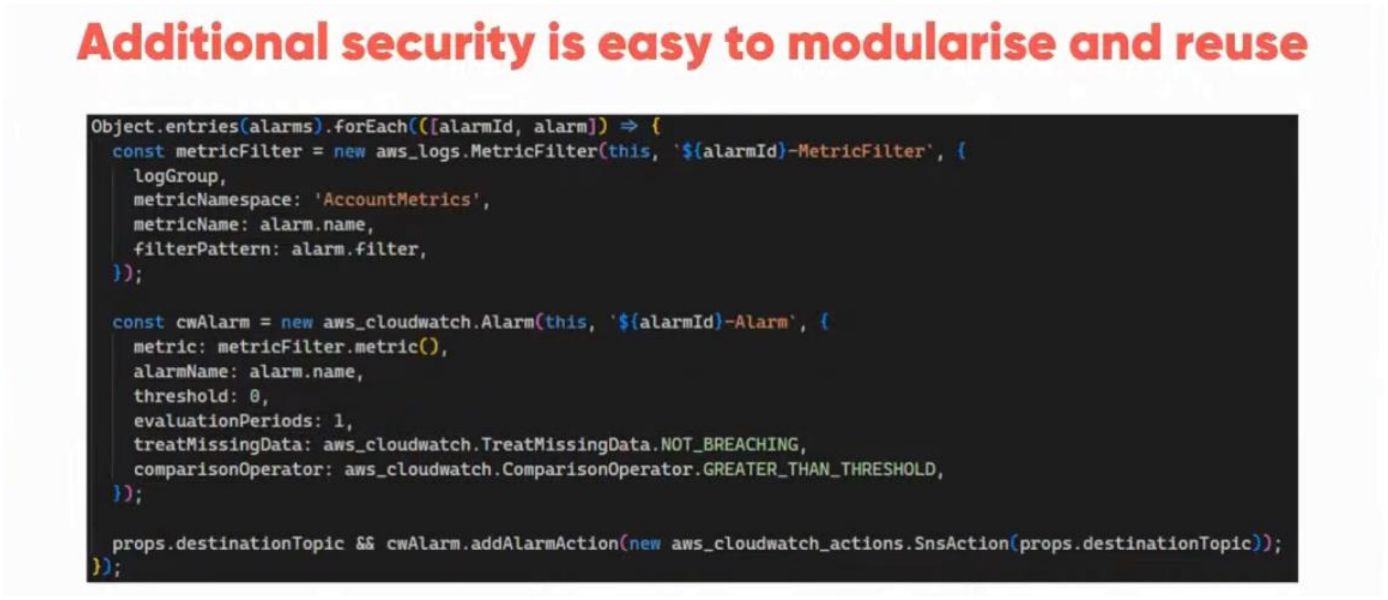


Additional security is easy to modularise and reuse

```
const bucket = new aws_s3.Bucket(this, 'Bucket', {
  encryption: aws_s3.BucketEncryption.KMS,
  blockPublicAccess: aws_s3.BlockPublicAccess.BLOCK_ALL,
  encryptionKey: encryptionKey,
  enforceSSL: true,
});

const logGroup = new aws_logs.LogGroup(this, 'LogGroup', {
  logGroupName: logGroupName,
  removalPolicy: RemovalPolicy.RETAIN,
  retention: retention,
});

const trail = new aws_cloudtrail.Trail(this, 'CloudTrail', {
  sendToCloudWatchLogs: true,
  enableFileValidation: true,
  encryptionKey: encryptionKey,
  cloudWatchLogGroup: logGroup,
  isMultiRegionTrail: true,
  bucket: bucket,
  includeGlobalServiceEvents: true,
});
```

2.1 Ensure CloudTrail is enabled in all regions (Scored)
2.2 Ensure CloudTrail log file validation is enabled (Scored)
2.3 Ensure the S3 bucket used to store CloudTrail logs is not publicly accessible (Scored)
2.4 Ensure CloudTrail trails are integrated with CloudWatch Logs (Scored)

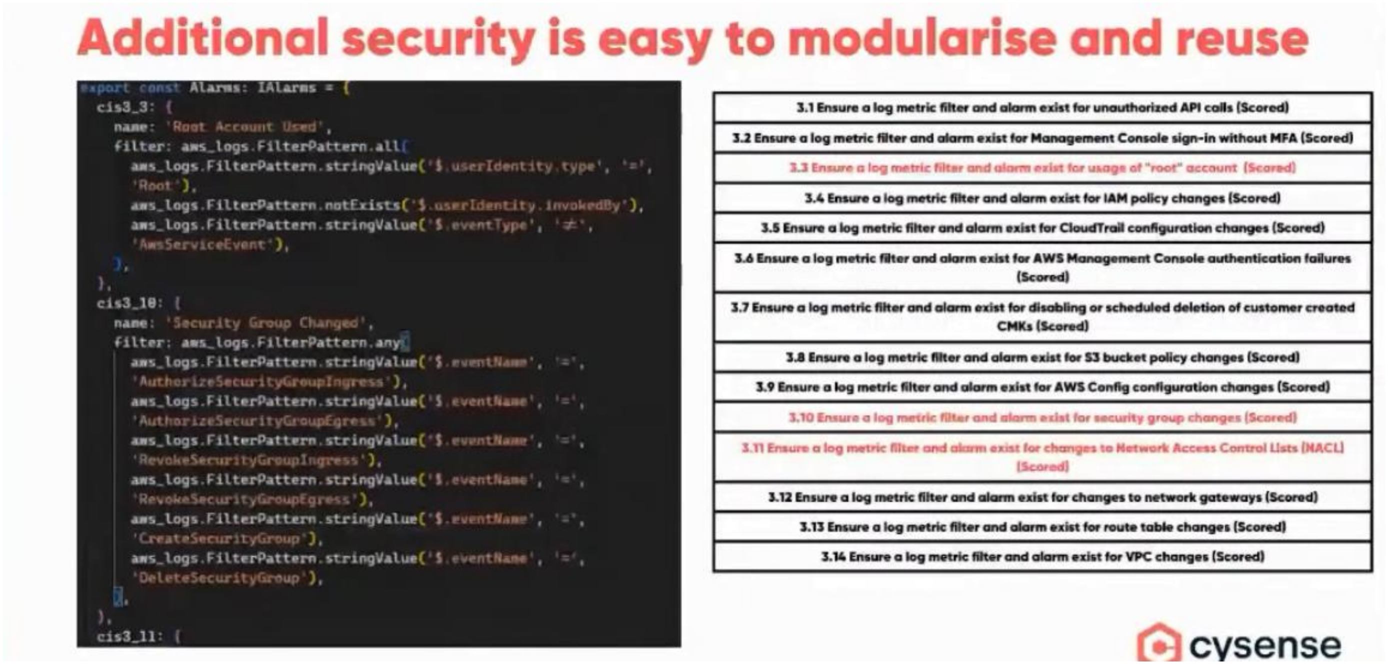
I created a CloudTrail with the needed configuration settings like file validation, encryption, cloudwatch logs being enabled and integrated with cloudTrail, set up a log group and made it multiregional. Then I sent those CloudTrail to a LogGroup and then sent it to an S3 Bucket.



Additional security is easy to modularise and reuse

```
Object.entries(alarms).forEach(([alarmId, alarm]) => {
  const metricFilter = new aws_logs.MetricFilter(this, `${alarmId}-MetricFilter`, {
    logGroup,
    metricNamespace: 'AccountMetrics',
    metricName: alarm.name,
    filterPattern: alarm.filter,
  });

  const cwAlarm = new aws_cloudwatch.Alarm(this, `${alarmId}-Alarm`, {
    metric: metricFilter.metric(),
    alarmName: alarm.name,
    threshold: 0,
    evaluationPeriods: 1,
    treatMissingData: aws_cloudwatch.TreatMissingData.NOT_BREACHING,
    comparisonOperator: aws_cloudwatch.ComparisonOperator.GREATER_THAN_THRESHOLD,
  });

  props.destinationTopic && cwAlarm.addAlarmAction(new aws_cloudwatch_actions.SnsAction(props.destinationTopic));
});
```

Next, we need to create Alarms. I created an Alarms object, we then iterate through the Alarms object and created a metricFilter for each alarm, name and forwarded each alarm to a destination like an email or a SNS topic.



Additional security is easy to modularise and reuse

```
export const Alarms: IAlarms = {
  cis3_3: {
    name: 'Root Account Used',
    filter: aws_logs.FilterPattern.all(
      aws_logs.FilterPattern.stringValue('$.userIdentity.type', '=',
      'Root'),
      aws_logs.FilterPattern.notExists('$.userIdentity.invokedBy'),
      aws_logs.FilterPattern.stringValue('$.eventType', '≠',
      'AwsServiceEvent'),
    ),
  },
  cis3_10: {
    name: 'Security Group Changed',
    filter: aws_logs.FilterPattern.any(
      aws_logs.FilterPattern.stringValue('$.eventName', '=',
      'AuthorizeSecurityGroupIngress'),
      aws_logs.FilterPattern.stringValue('$.eventName', '=',
      'AuthorizeSecurityGroupEgress'),
      aws_logs.FilterPattern.stringValue('$.eventName', '=',
      'RevokeSecurityGroupIngress'),
      aws_logs.FilterPattern.stringValue('$.eventName', '=',
      'RevokeSecurityGroupEgress'),
      aws_logs.FilterPattern.stringValue('$.eventName', '=',
      'CreateSecurityGroup'),
      aws_logs.FilterPattern.stringValue('$.eventName', '=',
      'DeleteSecurityGroup'),
    ),
  },
  cis3_11: {
```

3.1 Ensure a log metric filter and alarm exist for unauthorized API calls (Scored)
3.2 Ensure a log metric filter and alarm exist for Management Console sign-in without MFA (Scored)
3.3 Ensure a log metric filter and alarm exist for usage of "root" account (Scored)
3.4 Ensure a log metric filter and alarm exist for IAM policy changes (Scored)
3.5 Ensure a log metric filter and alarm exist for CloudTrail configuration changes (Scored)
3.6 Ensure a log metric filter and alarm exist for AWS Management Console authentication failures (Scored)
3.7 Ensure a log metric filter and alarm exist for disabling or scheduled deletion of customer created CMKs (Scored)
3.8 Ensure a log metric filter and alarm exist for S3 bucket policy changes (Scored)
3.9 Ensure a log metric filter and alarm exist for AWS Config configuration changes (Scored)
3.10 Ensure a log metric filter and alarm exist for security group changes (Scored)
3.11 Ensure a log metric filter and alarm exist for changes to Network Access Control Lists (NACL) (Scored)
3.12 Ensure a log metric filter and alarm exist for changes to network gateways (Scored)
3.13 Ensure a log metric filter and alarm exist for route table changes (Scored)
3.14 Ensure a log metric filter and alarm exist for VPC changes (Scored)

cysense

Above is a snippet of what the alarm looks like. We pass the alarms to our Construct that chunks through to check if any of the alarm conditions is fired to act upon.

Additional security is easy to modularise and reuse

~100 Security related modules

CIS AWS Foundations Benchmark v1.2.0

Overview

Security score

100%

0% failed

	Failed	Unknown	No data	Passed	Disabled
43	0	0	0	43	0

All enabled

Thank you

Chad (cysense)

Chad Richts