



Avoiding AWS CDK? I was too, but now I'm in love!



Aaron Brighton

Helping product teams build **efficient**, **secure**,
and **scalable** modern applications in the cloud.

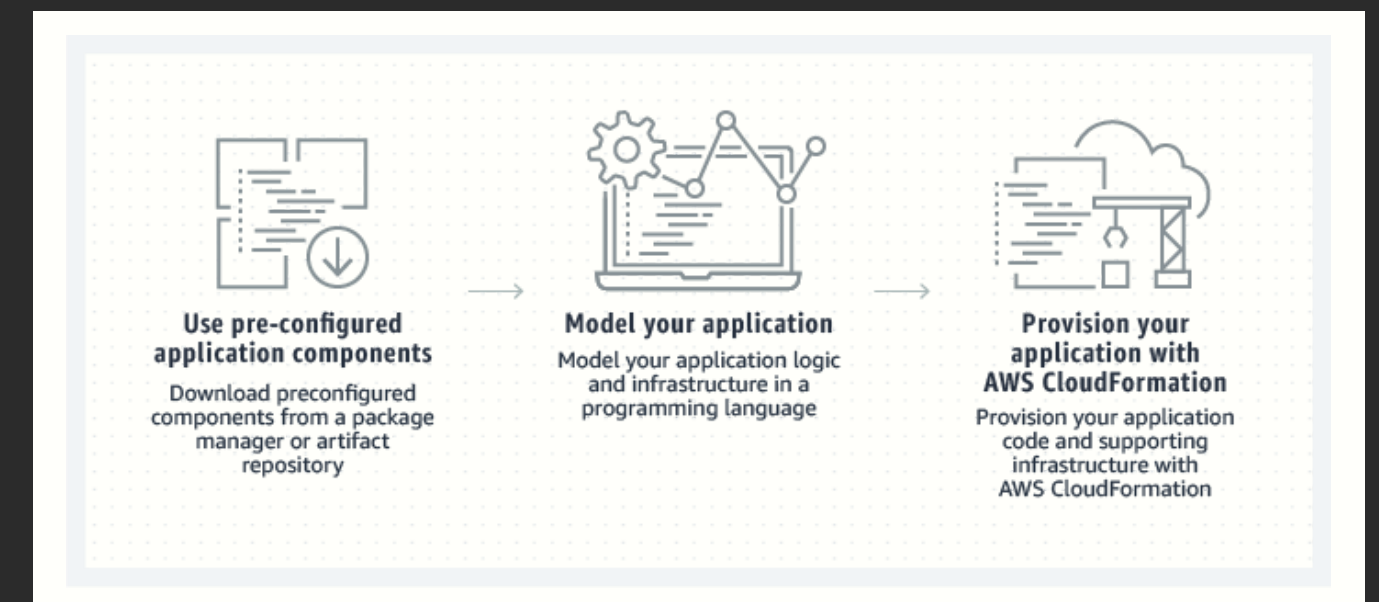


What is the AWS Cloud Development Kit?



“ The AWS Cloud Development Kit (AWS CDK) is an open source software development framework to define your cloud application resources using familiar programming languages. ”

<https://aws.amazon.com/cdk/>

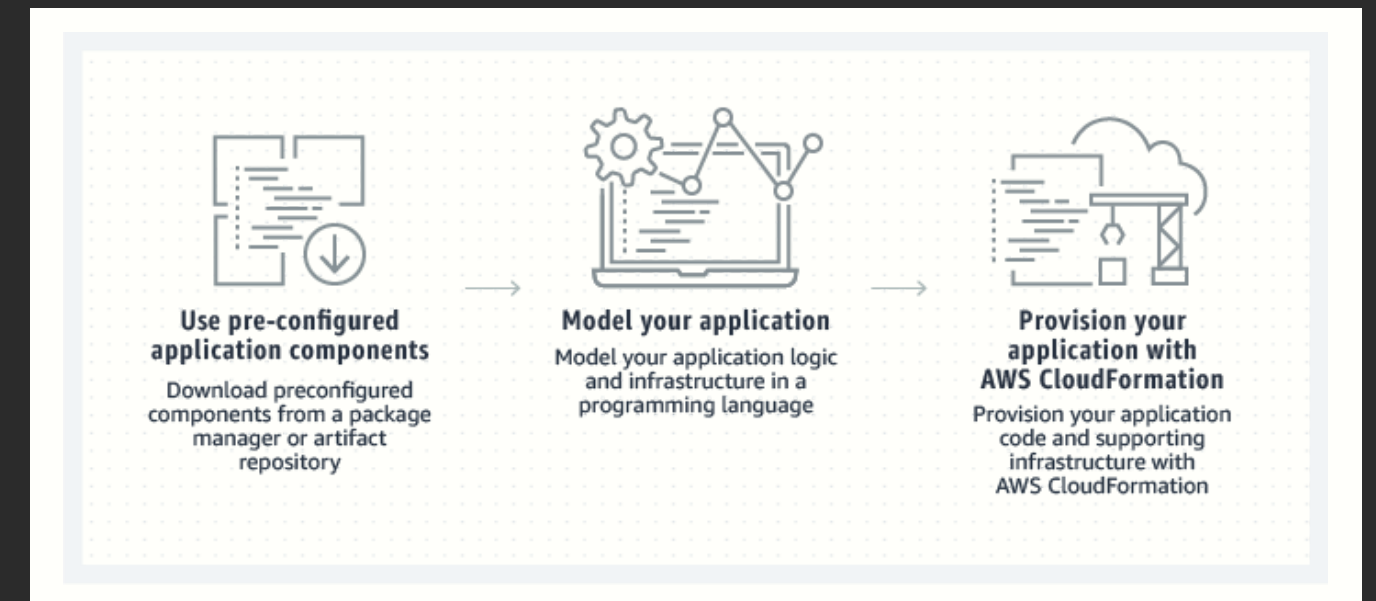


What is the AWS Cloud Development Kit?



“ The AWS Cloud Development Kit (AWS CDK) is an open source software development framework to define your cloud application resources using familiar programming languages. ”

<https://aws.amazon.com/cdk/>

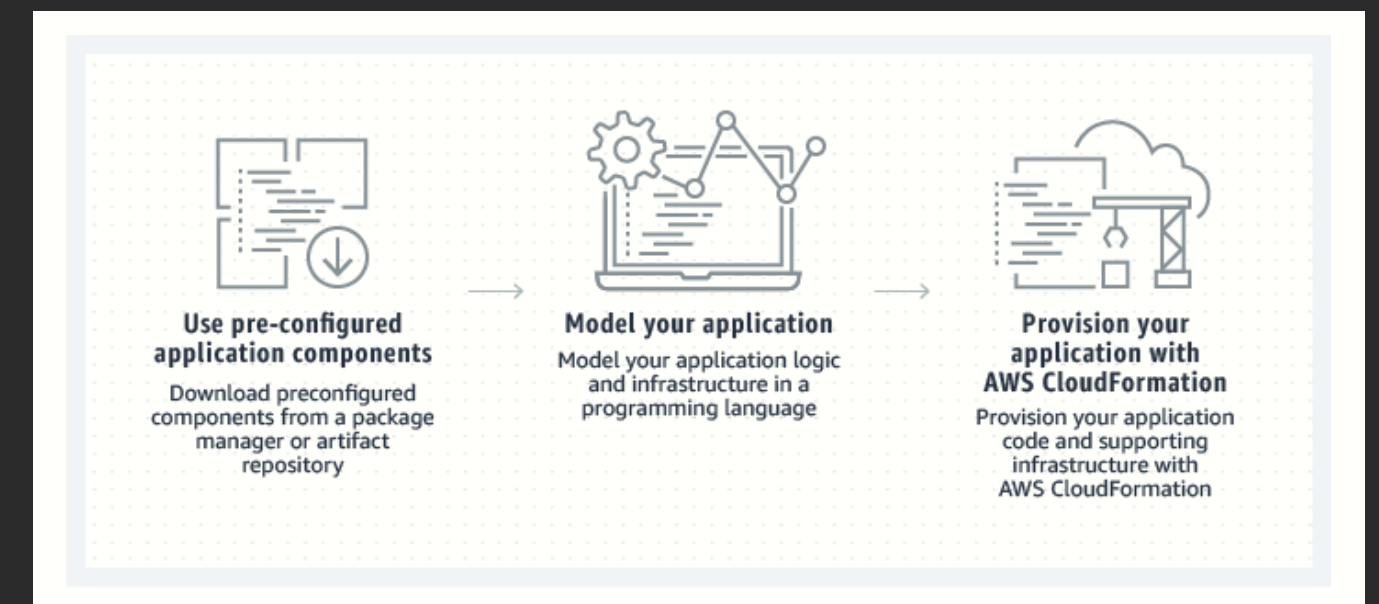


What is the AWS Cloud Development Kit?



“ The AWS Cloud Development Kit (AWS CDK) is an open source software development framework to define your cloud application resources using familiar programming languages. ”

<https://aws.amazon.com/cdk/>



Comparing CloudFormation to CDK

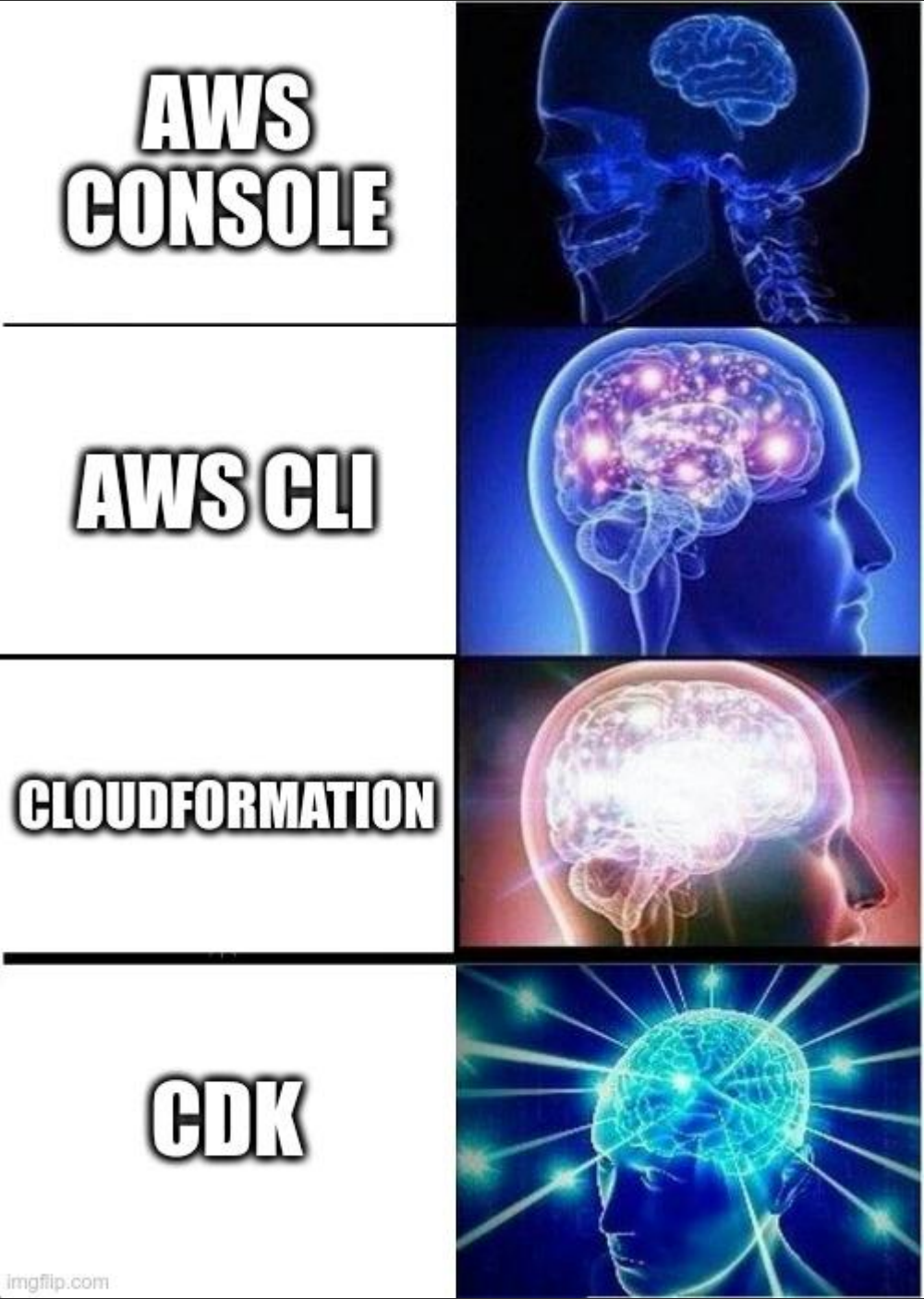


```
CloudFrontDistribution:
  Type: AWS::CloudFront::Distribution
  Properties:
    DistributionConfig:
      Enabled: true
      Aliases: !If [ CustomApiDomainSpecified, [ !Ref CustomApiDomain, !Sub 'www.${CustomApiDomain}' ], !Ref AWS::NoValue ]
      HttpVersion: http2 # Maximum HTTP version that you want viewers to use
      DefaultCacheBehavior:
        TargetOriginId: !Sub 'S3-${S3Bucket}'
        ViewerProtocolPolicy: redirect-to-https
        CachePolicyId: 4135ea2d-6df8-44a3-9df3-4b5a84be39ad # Managed-CachingDisabled
    Origins:
      - DomainName: !GetAtt S3Bucket.RegionalDomainName
        Id: !Sub 'S3-${S3Bucket}'
        S3OriginConfig:
          OriginAccessIdentity: !Sub 'origin-access-identity/cloudfront/${CloudFrontOriginAccessIdentity}'
      - DomainName: !Sub '${ApiGatewayRestApi}.execute-api.${AWS::Region}.amazonaws.com'
        Id: !Sub 'API-${ApiGatewayRestApi}'
        CustomOriginConfig:
          OriginProtocolPolicy: https-only
    CacheBehaviors:
      - PathPattern: 'api/**'
        TargetOriginId: !Sub 'API-${ApiGatewayRestApi}'
        ViewerProtocolPolicy: https-only
        AllowedMethods: # Only have three options, GET+HEAD, GET+HEAD+OPTIONS, or ALL as seen below.
          - GET
          - HEAD
          - OPTIONS
          - PUT
          - PATCH
          - POST
          - DELETE
        CachePolicyId: 4135ea2d-6df8-44a3-9df3-4b5a84be39ad # Managed-CachingDisabled
    ViewerCertificate: # Specify the certificate generated in ACM only if a CustomApiDomain was specified
      !If
        - CustomApiDomainSpecified
        - AcmCertificateArn: !Ref ACMCertificate
          MinimumProtocolVersion: TLSv1.2_2019 # Required when selecting custom certificate
          SslSupportMethod: sni-only # Required when selecting custom certificate
        - CloudFrontDefaultCertificate: true

Route53BaseRecord:
  Type: AWS::Route53::RecordSet
  Condition: CustomApiDomainSpecified # Only create certificate if CustomApiDomain provided
  Properties:
    HostedZoneId: !Ref CustomApiDomainZoneId
    AliasTarget:
      DNSName: !GetAtt CloudFrontDistribution.DomainName
      HostedZoneId: Z2FDTDATAQYW2 # Specific ID used for specialized A-Record based alias pointers for CloudFront
    Name: !Ref CustomApiDomain
    Type: A

Route53WwwRecord:
  Type: AWS::Route53::RecordSet
```


Comparing CloudFormation to CDK



CloudFormation

```
CloudFrontDistribution:
  Type: AWS::CloudFront::Distribution
  Properties:
    DistributionConfig:
      Enabled: true
      Aliases: !If [ CustomApiDomainSpecified, [ !Ref CustomApiDomain, !Sub 'www.${CustomApiDomain}' ], !Ref AWS::NoValue ]
      HttpVersion: http2 # Maximum HTTP version that you want viewers to use
      DefaultCacheBehavior:
        TargetOriginId: !Sub 'S3-${S3Bucket}'
        ViewerProtocolPolicy: redirect-to-https
        CachePolicyId: 4135ea2d-6df8-44a3-9df3-4b5a84be39ad # Managed-CachingDisabled
    Origins:
      - DomainName: !GetAtt S3Bucket.RegionalDomainName
        Id: !Sub 'S3-${S3Bucket}'
        S3OriginConfig:
          OriginAccessIdentity: !Sub 'origin-access-identity/cloudfront/${CloudFrontOriginAccessIdentity}'
          DomainName: !Sub '${ApiGatewayRestApi}.execute-api.${AWS::Region}.amazonaws.com'
          Id: !Sub 'API-${ApiGatewayRestApi}'
          CustomOriginConfig:
            OriginProtocolPolicy: https-only
            Paths:
              - PathPattern: 'api/*'
                TargetOriginId: !Sub 'API-${ApiGatewayRestApi}'
```

Constructs

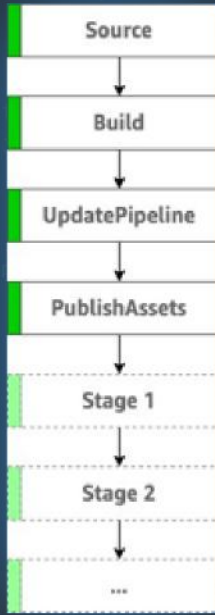
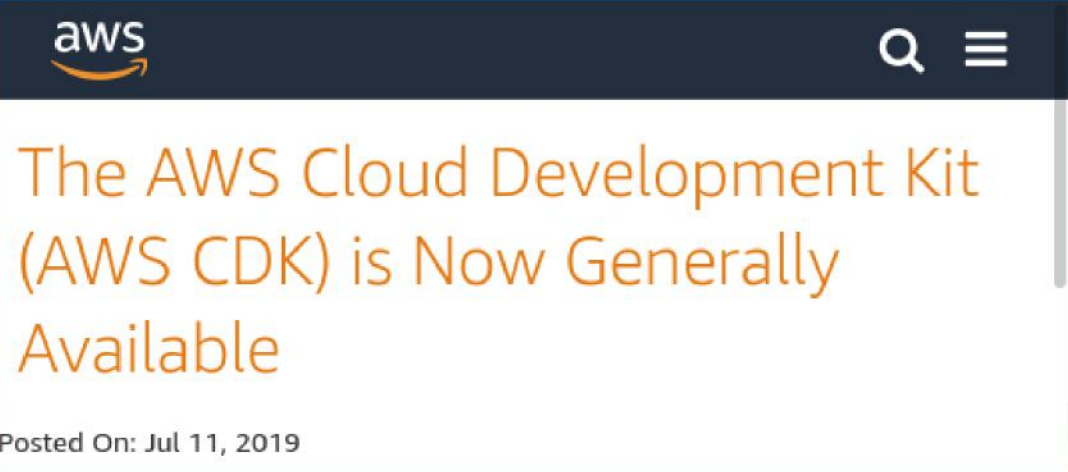
```
const cfToS3 = new CloudFrontToS3(this, 'test-cloudfront-s3', {
  insertHttpSecurityHeaders: false,
  cloudFrontDistributionProps: <cloudfront.DistributionProps>{
    defaultBehavior: {
      cachePolicy: cloudfront.CachePolicy.CACHING_DISABLED,
    },
  },
});

new s3deploy.BucketDeployment(this, 'DeployWebsite', {
  sources: [s3deploy.Source.asset('./website')],
  destinationBucket: cfToS3.s3Bucket,
});
```

sam-beta-cdk

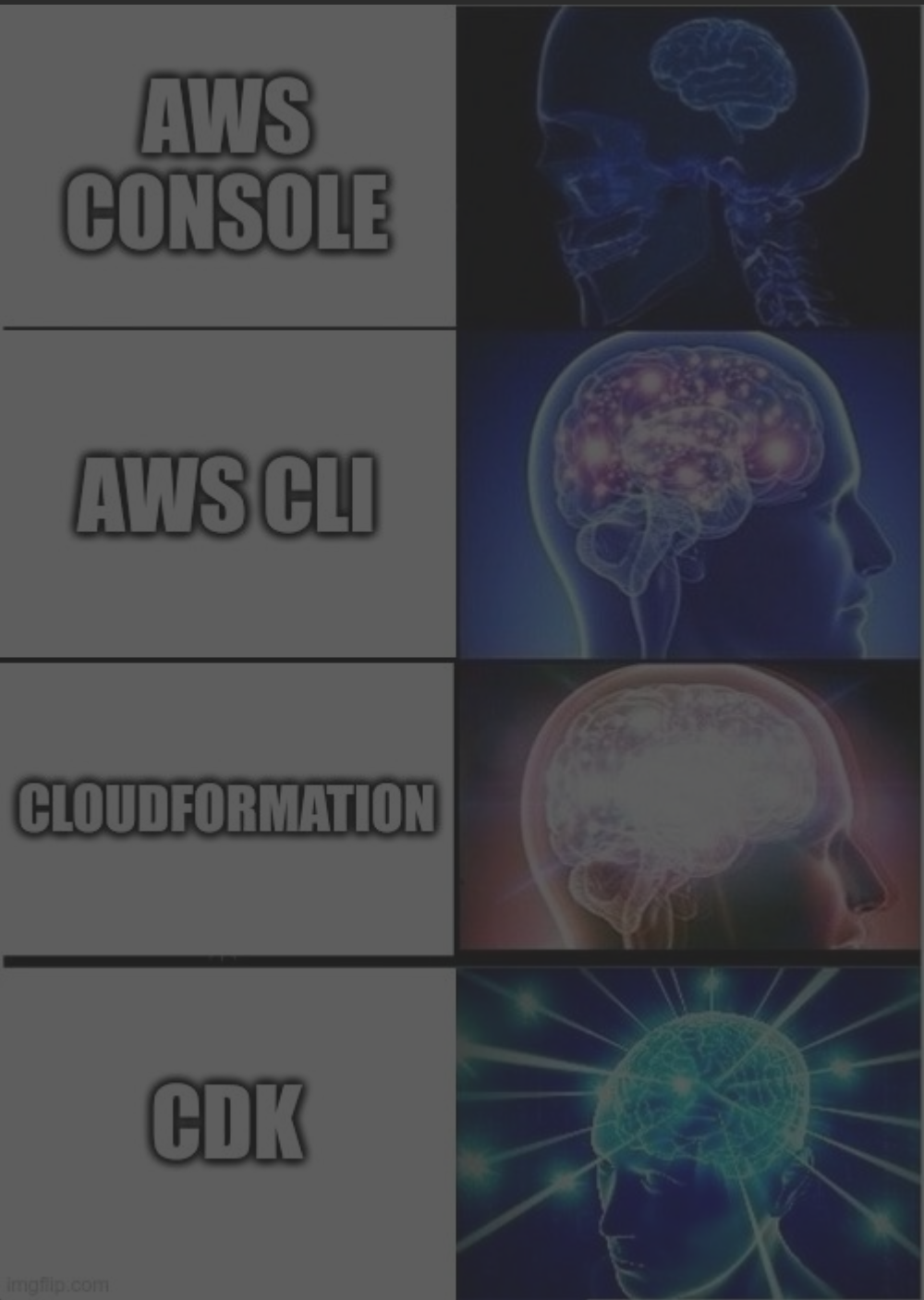


CDK



cdk-pipelines

Comparing CloudFormation to CDK



CloudFormation

```
CloudFrontDistribution:
  Type: AWS::CloudFront::Distribution
  Properties:
    DistributionConfig:
      Enabled: true
      Aliases: !If [ CustomApiDomainSpecified, [ !Ref CustomApiDomain, !Sub 'www.${CustomApiDomain}' ], !Ref AWS::NoValue ]
      HttpVersion: http2 # Maximum HTTP version that you want viewers to use
      DefaultCacheBehavior:
        TargetOriginId: !Sub 'S3-${S3Bucket}'
        ViewerProtocolPolicy: redirect-to-https
        CachePolicyId: 4135ea2d-6df8-44a3-9df3-4b5a84be39ad # Managed-CachingDisabled
    Origins:
      - DomainName: !GetAtt S3Bucket.RegionalDomainName
        Id: !Sub 'S3-${S3Bucket}'
        S3OriginConfig:
          OriginAccessIdentity: !Sub 'origin-access-identity/cloudfront/${CloudFrontOriginAccessIdentity}'
      - DomainName: !Sub '${ApiGatewayRestApi}.execute-api.${AWS::Region}.amazonaws.com'
        Id: !Sub 'API-${ApiGatewayRestApi}'
        CustomOriginConfig:
          OriginProtocolPolicy: https-only
      - PathPattern: 'api/*'
        TargetOriginId: !Sub 'API-${ApiGatewayRestApi}'
```

Constructs

```
const cfToS3 = new CloudFrontToS3(this, 'test-cloudfront-s3', {
  insertHttpSecurityHeaders: false,
  cloudFrontDistributionProps: <cloudfront.DistributionProps>{
    defaultBehavior: {
      cachePolicy: cloudfront.CachePolicy.CACHING_DISABLED,
    },
  },
});

new s3deploy.BucketDeployment(this, 'DeployWebsite', {
  sources: [s3deploy.Source.asset('./website')],
  destinationBucket: cfToS3.s3Bucket,
});
```



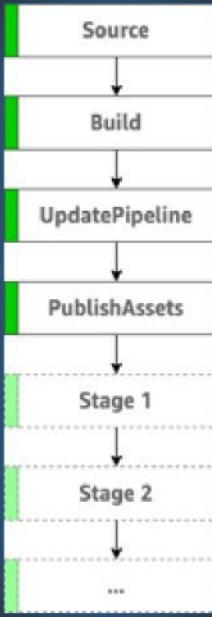
CDK

sam-beta-cdk



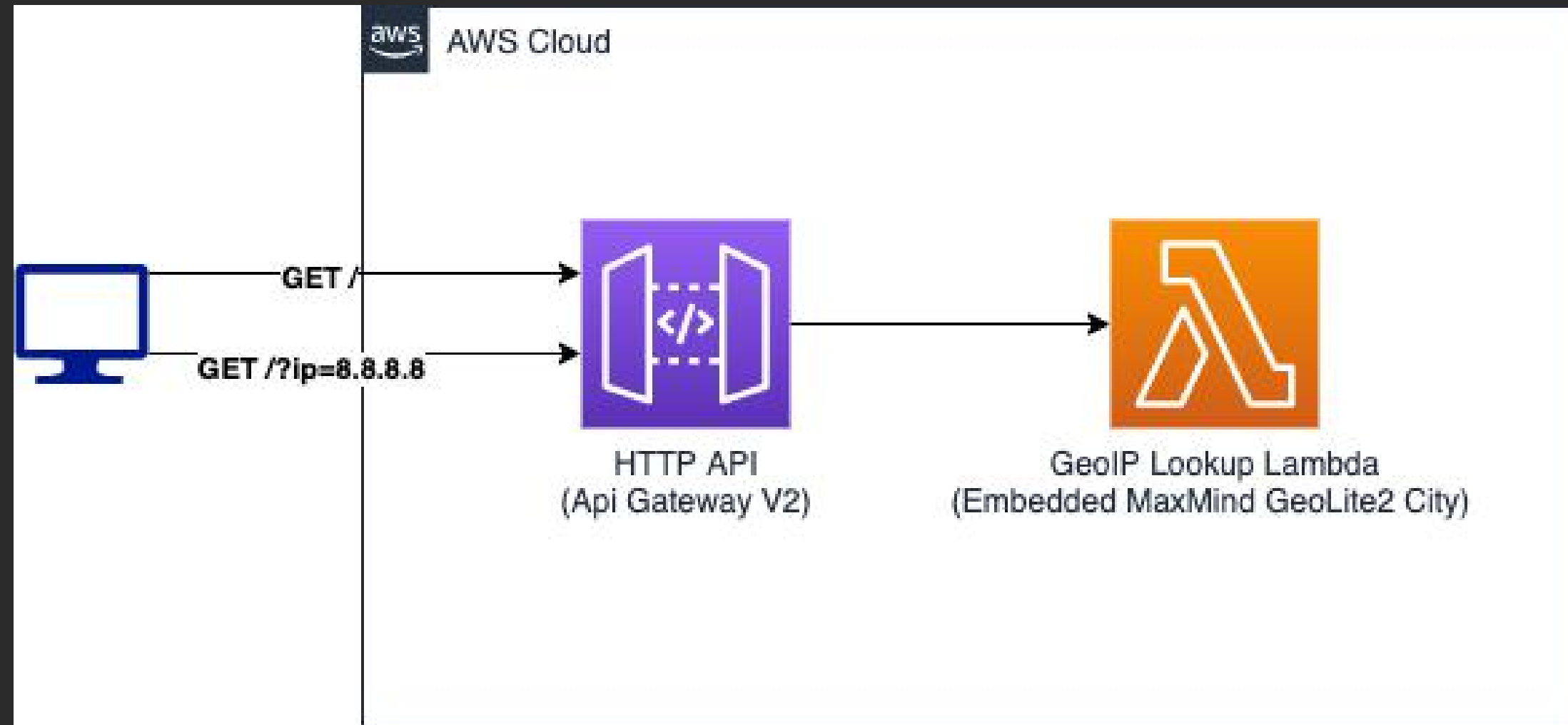
The AWS Cloud Development Kit (AWS CDK) is Now Generally Available

Posted On: Jul 11, 2019



cdk-pipelines

"quick and dirty serverless geoip api"



+ cdk-pipelines
resources

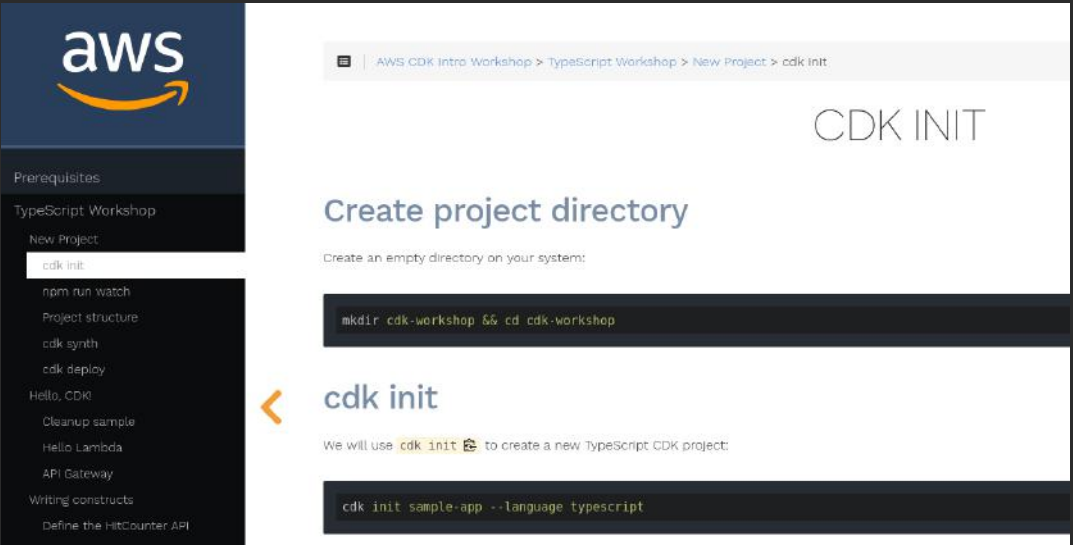
Additional Resources + Q&A

projen

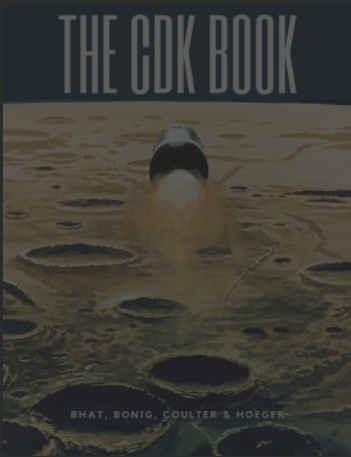



<https://github.com/projen/projen>

cdkworkshop.com



Much anticipated...



 @mattbonig, @hoegertn,
@SathyaBhat, @nideveloper



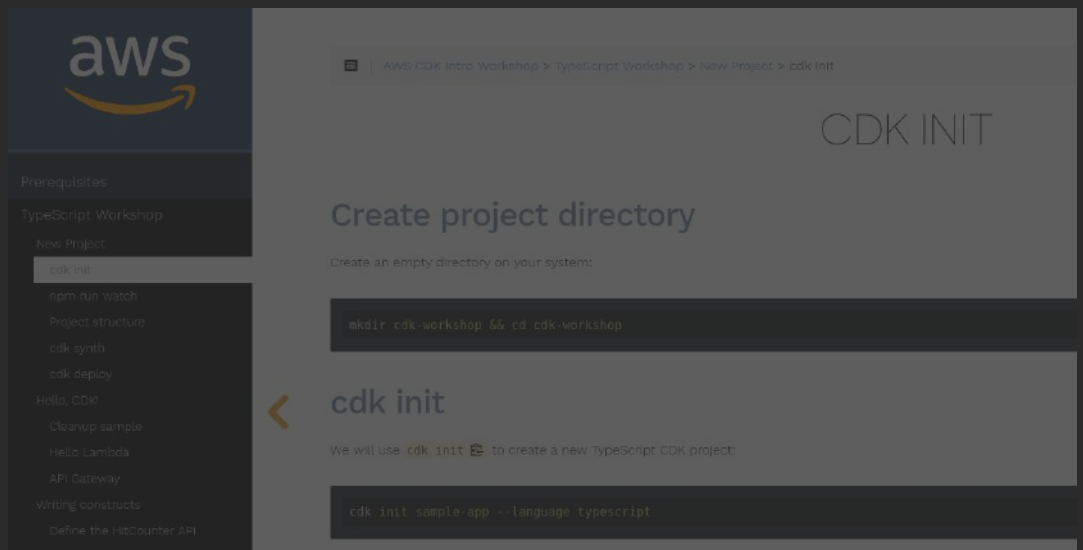
Additional Resources + Q&A

projen




<https://github.com/projen/projen>

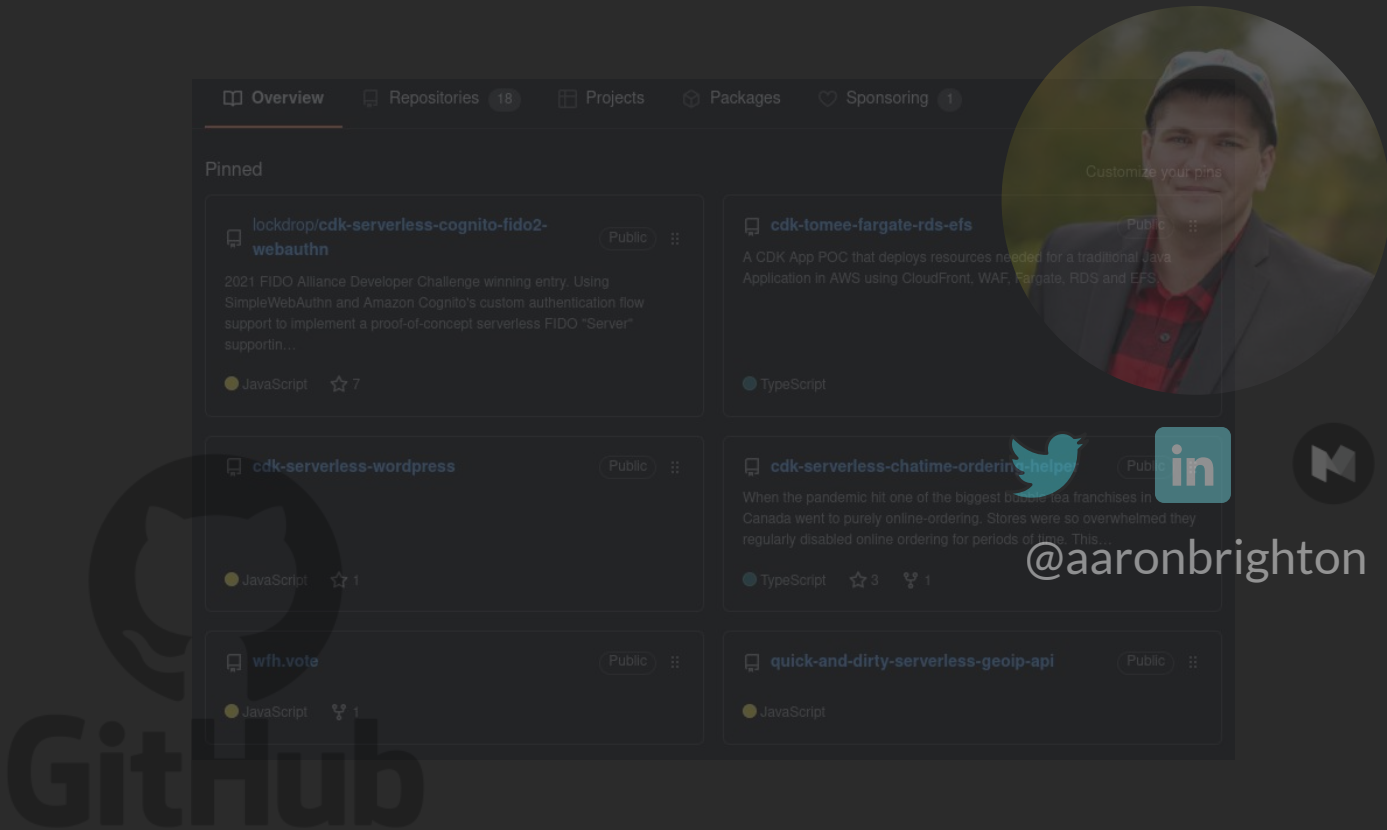
cdkworkshop.com



Much anticipated...



 @mattbonig, @hoegertn,
@SathyaBhat, @nideveloper



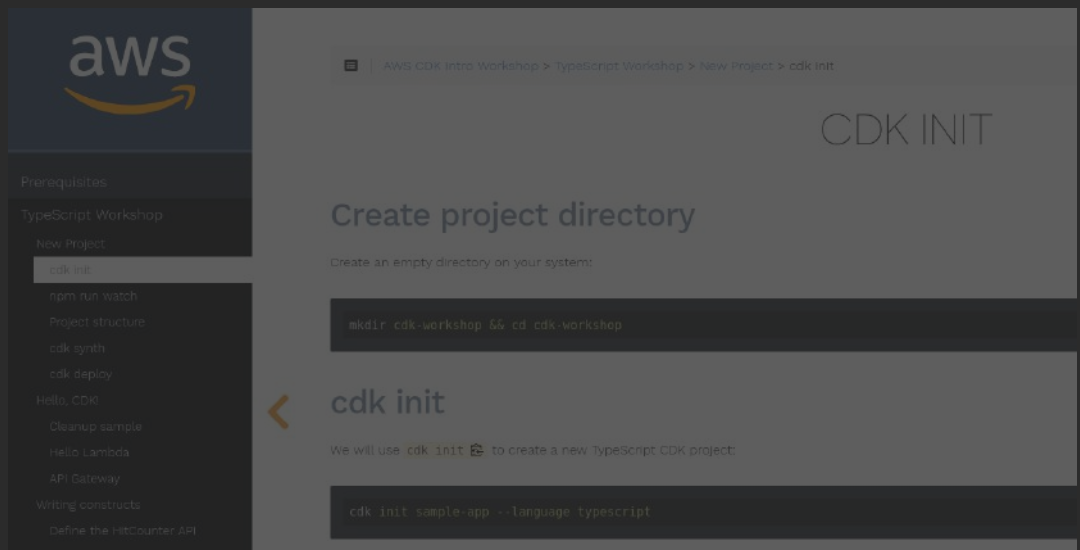
Additional Resources + Q&A

projen

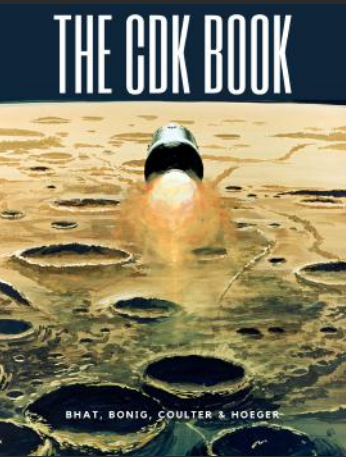



<https://github.com/projen/projen>

cdkworkshop.com



Much anticipated...



 @mattbonig, @hoegertn,
@SathyaBhat, @nideveloper



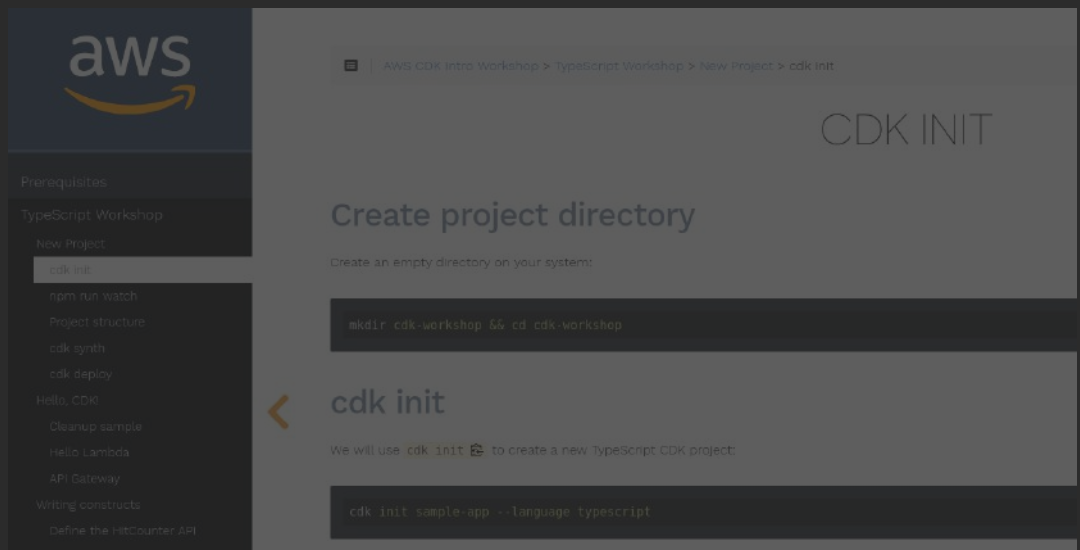
Additional Resources + Q&A

projen

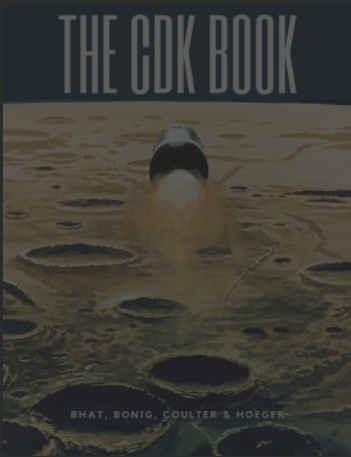



<https://github.com/projen/projen>

cdkworkshop.com



Much anticipated...



 @mattbonig, @hoegertn,
@SathyaBhat, @nideveloper

