:doctype: book

include::attributes.txt[]

// Attributes [.topic] [#bootstrapping-env] = Bootstrap your environment for use with the {aws} CDK :info\_titleabbrev: Bootstrap your environment :keywords: {aws} CDK, {aws} Cloud Development Kit ({aws} CDK), {aws} account, {aws} Region, Bootstrapping, Bootstrap, Environment

== [abstract]

## Bootstrap your {aws} environment before using the {aws} Cloud Development Kit ({aws} CDK) to deploy CDK stacks into your environment.

// Content start

Bootstrap your {aws} environment to prepare it for {aws} Cloud Development Kit ({aws} CDK) stack deployments.

* For an introduction to environments, see xref:environments[Environments for the {aws} CDK].
* For an introduction to bootstrapping, see xref:bootstrapping[{aws} CDK bootstrapping].

[#bootstrapping-howto] == How to bootstrap your environment

You can use the {aws} CDK Command Line Interface ({aws} CDK CLI) or your preferred {aws} CloudFormation deployment tool to bootstrap your environment.

[#bootstrapping-howto-cli] *Use the CDK CLI*:: + You can use the CDK CLI cdk bootstrap command to bootstrap your environment. This is the method that we recommend if you don’t require significant modifications to bootstrapping. + *Bootstrap from any working directory*::: + To bootstrap from any working directory, provide the environment to bootstrap as a command line argument. The following is an example: + [source,bash,subs=“verbatim,attributes”] — $ cdk bootstrap <aws://123456789012/us-east-1> — + [TIP] – If you don’t have your {aws} account number, you can get it from the {aws} Management Console. You can also use the following {aws} CLI command to display your default account information, including your account number:

== [source,none,subs=“verbatim,attributes”]

## $ aws sts get-caller-identity

If you have named profiles in your {aws} config and credentials files, use the --profile option to retrieve account information for a specific profile. The following is an example:

== [source,none,subs=“verbatim,attributes”]

$ aws sts get-caller-identity –profile ++++++—-++++++

To display the default Region, use the aws configure get command:

== [source,none,subs=“verbatim,attributes”]

$ aws configure get region $ aws configure get region –profile ++++++—- – +++++++

When providing an argument, the aws:// prefix is optional. The following is valid: + [source,none,subs=“verbatim,attributes”] — $ cdk bootstrap <123456789012/us-east-1> — + To bootstrap multiple environments at the same time, provide multiple arguments: + [source,none,subs=“verbatim,attributes”] — $ cdk bootstrap <aws://123456789012/us-east-1> <aws://123456789012/us-east-2> —

*Bootstrap from the parent directory of a CDK project*::: + You can run cdk bootstrap from the parent directory of a CDK project containing a cdk.json file. If you don't provide an environment as an argument, the CDK CLI will obtain environment information from default sources, such as yourconfigandcredentialsfiles or any environment information specified for your CDK stack. + When you bootstrap from the parent directory of a CDK project, environments provided from command line arguments take precedence over other sources. + To bootstrap an environment that is specified in yourconfigandcredentialsfiles, use the–profile` option: + [source,none,subs=“verbatim,attributes”] — $ cdk bootstrap –profile ++++++—- +++++++

For more information on the cdk bootstrap command and supported options, see xref:ref-cli-cmd-bootstrap[cdk bootstrap].

[#bootstrapping-howto-cfn] *Use any {aws} CloudFormation tool*:: + You can copy the https://github.com/aws/aws-cdk-cli/blob/main/packages/aws-cdk/lib/api/bootstrap/bootstrap-template.yaml[bootstrap template] from the *aws-cdk-cli GitHub repository* or obtain the template with the cdk bootstrap --show-template command. Then, use any {aws} CloudFormation tool to deploy the template into your environment. + With this method, you can use {aws} CloudFormation StackSets or {aws} Control Tower. You can also use the {aws} CloudFormation console or the {aws} Command Line Interface ({aws} CLI). You can make modifications to your template before you deploy it. This method may be more flexible and suitable for large-scale deployments. + The following is an example of using the --show-template option to retrieve and save the bootstrap template to your local machine: + ==== [role=“tablist”] macOS/Linux:: + [source,none,subs=“verbatim,attributes”] — $ cdk bootstrap –show-template > bootstrap-template.yaml —

Windows:: On Windows, PowerShell must be used to preserve the encoding of the template. + [source,none,subs=“verbatim,attributes”] — powershell “cdk bootstrap –show-template | Out-File -encoding utf8 bootstrap-template.yaml” — ==== +

= [NOTE]

If CDK notices are appearing in your {aws} CloudFormation template output, provide the --no-notices option with your command.

==== + To deploy this template using the CDK CLI, you can run the following: + [source,none,subs=“verbatim,attributes”] — $ cdk bootstrap –template +++<bootstrap-template.yaml>+++—- + The following is an example of using the {aws} CLI to deploy the template: + ==== [role=“tablist”] macOS/Linux:: + [source,none,subs=“verbatim,attributes”] —- aws cloudformation create-stack  –stack-name CDKToolkit  –template-body file://bootstrap-template.yaml  –capabilities CAPABILITY\_NAMED\_IAM  –region ++++++—-+++++++++</bootstrap-template.yaml>+++

Windows:: + [source,none,subs=“verbatim,attributes”] — aws cloudformation create-stack {caret} –stack-name CDKToolkit {caret} –template-body file://bootstrap-template.yaml {caret} –capabilities CAPABILITY\_NAMED\_IAM {caret} –region ++++++—- ==== +++++++

For information on using CloudFormation StackSets to bootstrap multiple environments, see https://aws.amazon.com/blogs/mt/bootstrapping-multiple-aws-accounts-for-aws-cdk-using-cloudformation-stacksets/[Bootstrapping multiple {aws} accounts for {aws} CDK using CloudFormation StackSets] in the *{aws} Cloud Operations & Migrations Blog*.

[#bootstrapping-env-when] == When to bootstrap your environment

You must bootstrap each {aws} environment before you deploy into the environment. We recommend that you proactively bootstrap each environment that you plan to use. You can do this before you plan on actually deploying CDK apps into the environment. By proactively bootstrapping your environments, you prevent potential future issues such as Amazon S3 bucket name conflicts or deploying CDK apps into environments that haven’t been bootstrapped.

It’s okay to bootstrap an environment more than once. If an environment has already been bootstrapped, the bootstrap stack will be upgraded if necessary. Otherwise, nothing will happen.

If you attempt to deploy a CDK stack into an environment that hasn`’t been bootstrapped, you will see an error like the following:

== [source,none,subs=“verbatim,attributes”]

$ cdk deploy

✨ Synthesis time: 2.02s

== ❌ Deployment failed: Error: BootstrapExampleStack: SSM parameter /cdk-bootstrap/hnb659fds/version not found. Has the environment been bootstrapped? Please run ‘cdk bootstrap’ (see https://docs.aws.amazon.com/cdk/latest/guide/bootstrapping.html)

[#bootstrapping-env-when-update] *Update your bootstrap stack*:: + Periodically, the CDK team will update the bootstrap template to a new version. When this happens, we recommend that you update your bootstrap stack. If you haven`’t customized the bootstrapping process, you can update your bootstrap stack by following the same steps that you took to originally bootstrap your environment. For more information, see xref:bootstrap-template-history[Bootstrap template version history].

[#bootstrapping-env-default] == Default resources created during bootstrapping

[#bootstrapping-env-roles] *IAM roles created during bootstrapping*:: + By default, bootstrapping provisions the following {aws} Identity and Access Management (IAM) roles in your environment: + –

* CloudFormationExecutionRole
* DeploymentActionRole
* FilePublishingRole
* ImagePublishingRole
* {blank}
* == LookupRole
* [#bootstrapping-env-roles-cfn] CloudFormationExecutionRole:::
* This IAM role is a CloudFormation service role that grants CloudFormation permission to perform stack deployments on your behalf. This role gives CloudFormation permission to perform {aws} API calls in your account, including deploying stacks.
* By using a service role, the permissions provisioned for the service role determine what actions can be performed on your CloudFormation resources. Without this service role, the security credentials you provide with the CDK CLI would determine what CloudFormation is allowed to do.
* [#bootstrapping-env-roles-deploy] DeploymentActionRole:::
* This IAM role grants permission to perform deployments into your environment. It is assumed by the CDK CLI during deployments.
* By using a role for deployments, you can perform cross-account deployments since the role can be assumed by {aws} identities in a different account.
* [#bootstrapping-env-roles-s3] FilePublishingRole:::
* This IAM role grants permission to perform actions against the bootstrapped Amazon Simple Storage Service (Amazon S3) bucket, including uploading and deleting assets. It is assumed by the CDK CLI during deployments.
* [#bootstrapping-env-roles-ecr] ImagePublishingRole:::
* This IAM role grants permission to perform actions against the bootstrapped Amazon Elastic Container Registry (Amazon ECR) repository. It is assumed by the CDK CLI during deployments.
* [#bootstrapping-env-roles-lookup] LookupRole:::
* This IAM role grants readOnly permission to look up xref:context[context values] from the {aws} environment. It is assumed by the CDK CLI when performing tasks such as template synthesis and deployments.

[#bootstrapping-env-default-id] *Resource IDs created during bootstrapping*:: + When you deploy the default bootstrap template, physical IDs for bootstrap resources are created using the following structure: cdk-<qualifier>-<description>-<account-ID>-<Region>. + –

* *Qualifier* – A nine character unique string value of hnb659fds. The actual value has no significance.
* *Description* – A short description of the resource. For example, container-assets.
* *Account ID* – The {aws} account ID of the environment.
* {blank}
* == *Region* – The {aws} Region of the environment.
* The following is an example physical ID of the Amazon S3 staging bucket created during bootstrapping: cdk-hnb659fds-assets-012345678910-us-west-1.

[#bootstrapping-env-permissions] == Permissions to use when bootstrapping your environment

When bootstrapping an {aws} environment, the IAM identity performing the bootstrapping must have at least the following permissions:

== [source,json,subs=“verbatim,attributes”]

{ “Version”: “2012-10-17”, “Statement”: [ { “Effect”: “Allow”, “Action”: [ “cloudformation:**”, ”ecr:**”, “ssm:**”, ”s3:**”, “iam:\_\_” ], “Resource”: “\_\_” } ] } —

Over time, the bootstrap stack, including the resources that are created and permissions they require, may change. With future changes, you may need to modify the permissions required to bootstrap an environment.

[#bootstrapping-env-customize] == Customize bootstrapping

If the default bootstrap template doesn`’t suit your needs, you can customize the bootstrapping of resources into your environment in the following ways:

* Use command line options with the cdk bootstrap command – This method is best for making small, specific changes that are supported through command line options.
* Modify the default bootstrap template and deploy it – This method is best for making complex changes or if you want complete control over the configuration of resources provisioned during bootstrapping.

For more information on customizing bootstrapping, see xref:bootstrapping-customizing[Customize {aws} CDK bootstrapping].

[#bootstrapping-env-pipelines] == Bootstrapping with CDK Pipelines

If you are using CDK Pipelines to deploy into another account’s environment, and you receive a message like the following:

== [source,none,subs=“verbatim,attributes”]

## Policy contains a statement with one or more invalid principals

This error message means that the appropriate IAM roles do not exist in the other environment. The most likely cause is that the environment has not been bootstrapped. Bootstrap the environment and try again.

[#bootstrapping-env-pipelines-protect] *Protecting your bootstrap stack from deletion*:: + If a bootstrap stack is deleted, the {aws} resources that were originally provisioned in the environment to support CDK deployments will also be deleted. This will cause the pipeline to stop working. If this happens, there is no general solution for recovery. + After your environment is bootstrapped, do not delete and recreate the environment’s bootstrap stack. Instead, try to update the bootstrap stack to a new version by running the cdk bootstrap command again. + To protect against accidental deletion of your bootstrap stack, we recommend that you provide the --termination-protection option with the cdk bootstrap command to enable termination protection. You can enable termination protection on new or existing bootstrap stacks. For instructions on enabling termination protection, see xref:bootstrapping-customizing-cli-protection[Enable termination protection for the bootstrap stack].

[#bootstrap-template-history] == Bootstrap template version history

The bootstrap template is versioned and evolves over time with the {aws} CDK itself. If you provide your own bootstrap template, keep it up to date with the canonical default template. You want to make sure that your template continues to work with all CDK features.

= [NOTE]

Earlier versions of the bootstrap template created an {aws} KMS key in each bootstrapped environment by default. To avoid charges for the KMS key, re-bootstrap these environments using --no-bootstrap-customer-key. The current default is no KMS key, which helps avoid these charges.

====

This section contains a list of the changes made in each version.

[cols=“1,1,1”, options=“header”] |=== | Template version | {aws} CDK version | Changes

|=== | *1* | 1.40.0 | Initial version of template with Bucket, Key, Repository, and Roles. |===

|=== | *2* | 1.45.0 | Split asset publishing role into separate file and image publishing roles. |===

|=== | *3* | 1.46.0 | Add FileAssetKeyArn export to be able to add decrypt permissions to asset consumers. |===

|=== | *4* | 1.61.0 | {aws} KMS permissions are now implicit via Amazon S3 and no longer require FileAssetKeyArn. Add CdkBootstrapVersion SSM parameter so the bootstrap stack version can be verified without knowing the stack name. |===

|=== | *5* | 1.87.0 | Deployment role can read SSM parameter. |===

|=== | *6* | 1.108.0 | Add lookup role separate from deployment role. |===

|=== | *6* | 1.109.0 | Attach aws-cdk:bootstrap-role tag to deployment, file publishing, and image publishing roles. |===

|=== | *7* | 1.110.0 | Deployment role can no longer read Buckets in the target account directly. (However, this role is effectively an administrator, and could always use its {aws} CloudFormation permissions to make the bucket readable anyway). |===

|=== | *8* | 1.114.0 | The lookup role has full read-only permissions to the target environment, and has a aws-cdk:bootstrap-role tag as well. |===

|=== | *9* | 2.1.0 | Fixes Amazon S3 asset uploads from being rejected by commonly referenced encryption SCP. |===

|=== | *10* | 2.4.0 | Amazon ECR ScanOnPush is now enabled by default. |===

|=== | *11* | 2.18.0 | Adds policy allowing Lambda to pull from Amazon ECR repos so it survives re-bootstrapping. |===

|=== | *12* | 2.20.0 | Adds support for experimental cdk import. |===

|=== | *13* | 2.25.0 | Makes container images in bootstrap-created Amazon ECR repositories immutable. |===

|=== | *14* | 2.34.0 | Turns off Amazon ECR image scanning at the repository level by default to allow bootstrapping Regions that do not support image scanning. |===

|=== | *15* | 2.60.0 | KMS keys cannot be tagged. |===

|=== | *16* | 2.69.0 | Addresses Security Hub finding link:https://docs.aws.amazon.com/securityhub/latest/userguide/kms-controls.html#kms-2[KMS.2]. |===

|=== | *17* | 2.72.0 | Addresses Security Hub finding link:https://docs.aws.amazon.com/securityhub/latest/userguide/ecr-controls.html#ecr-3[ECR.3]. |===

|=== | *18* | 2.80.0 | Reverted changes made for version 16 as they don’t work in all partitions and are are not recommended. |===

|=== | *19* | 2.106.1 | Reverted changes made to version 18 where AccessControl property was removed from the template. (https://github.com/aws/aws-cdk/issues/27964[#27964]) |===

|=== | *20* | 2.119.0 | Add ssm:GetParameters action to the {aws} CloudFormation deploy IAM role. For more information, see link:https://github.com/aws/aws-cdk/pull/28336/files#diff-4fdac38426c4747aa17d515b01af4994d3d2f12c34f7b6655f24328259beb7bf[#28336]. |===

|=== | *21* | 2.149.0 | Add condition to the file publishing role. |===

|=== | *22* | 2.160.0 | Add sts:TagSession permissions to the trust policy of bootstrap IAM roles. |===

|=== | *23* | 2.161.0 | Add cloudformation:RollbackStack and cloudformation:ContinueUpdateRollback permissions to the trust policy of the deploy IAM role. This provides permissions for the cdk rollback command. |===

|=== | *24* | 2.165.0 | Change the duration of days that noncurrent objects in the bootstrap bucket will be retained, from 365 to 30 days. Since the new cdk gc command introduces the ability to delete objects in the bootstrap bucket, this new behavior ensures that deleted objects remain in the bootstrap bucket for 30 days instead of 365 days. For more information on this change, see aws-cdk PR https://github.com/aws/aws-cdk/pull/31949[#31949]. |===

|=== | *25* | 2.165.0 | Add support to the bootstrap bucket for the removal of incomplete multipart uploads. Incomplete multipart uploads will be deleted after 1 day. For more information on this change, see aws-cdk PR https://github.com/aws/aws-cdk/pull/31956[#31956]. |===

|=== | *26* | 2.1002.0 | Add two deletion-related policies (UpdateReplacePolicy and DeletionPolicy to the FileAssetsBucketEncryptionKey) resource. These policies ensure that the old {aws} KMS key resource will be properly deleted when the bootstrap stack is updated or deleted. For more information on this change, see aws-cdk-cli PR https://github.com/aws/aws-cdk-cli/pull/100[#100]. |===

|=== | *27* | 2.1003.0 | Add new Amazon ECR resource policy to grant Amazon EMR Serverless specific permissions for retrieving container images. For more information on this change, see aws-cdk-cli PR https://github.com/aws/aws-cdk-cli/pull/112[#112]. |===

[#bootstrapping-template] == Upgrade from legacy to modern bootstrap template

The {aws} CDK v1 supported two bootstrapping templates, legacy and modern. CDK v2 supports only the modern template. For reference, here are the high-level differences between these two templates.

[cols=“1h,1,1”, options=“header”] |=== | Feature | Legacy (v1 only) | Modern (v1 and v2)

|=== | Cross-account deployments | Not allowed | Allowed |===

|=== | {aws} CloudFormation Permissions | Deploys using current user’s permissions (determined by {aws} profile, environment variables, etc.) | Deploys using the permissions specified when the bootstrap stack was provisioned (for example, by using --trust) |===

|=== | Versioning | Only one version of bootstrap stack is available | Bootstrap stack is versioned; new resources can be added in future versions, and {aws} CDK apps can require a minimum version |===

|=== | Resources\* | Amazon S3 bucket | a |===

* Amazon S3 bucket
* {aws} KMS key
* IAM roles
* Amazon ECR repository
* SSM parameter for versioning

|=== | {aws} KMS key | IAM roles | Amazon ECR repository |===

[cols=2\*] |=== | // | SSM parameter for versioning |===

|=== | Resource naming | Automatically generated | Deterministic |===

|=== | Bucket encryption | Default key | {aws} managed key by default. You can customize to use a customer managed key. |===

* *We will add additional resources to the bootstrap template as needed.*

An environment that was bootstrapped using the legacy template must be upgraded to use the modern template for CDK v2 by re-bootstrapping. Re-deploy all {aws} CDK applications in the environment at least once before deleting the legacy bucket.

[#bootstrapping-securityhub] == Address Security Hub Findings

If you are using {aws} Security Hub, you may see findings reported on some of the resources created by the {aws} CDK bootstrapping process. Security Hub findings help you find resource configurations you should double-check for accuracy and safety. We have reviewed these specific resource configurations with {aws} Security and are confident they do not constitute a security problem.

[#bootstrapping-securityhub-kms2] *[KMS.2] IAM principals should not have IAM inline policies that allow decryption actions on all KMS keys*:: + The *deploy role* (DeploymentActionRole) grants permission to read encrypted data, which is necessary for cross-account deployments with CDK Pipelines. Policies in this role do not grant permission to all data. It only grants permission to read encrypted data from Amazon S3 and {aws} KMS, and only when those resources allow it through their bucket or key policy. + The following is a snippet of these two statements in the *deploy role* from the bootstrap template: + [source,yaml,subs=“verbatim,attributes”] — DeploymentActionRole: Type: {aws}::IAM::Role Properties: … Policies: - PolicyDocument: Statement: … - Sid: PipelineCrossAccountArtifactsBucket Effect: Allow Action: - s3:GetObject *- s3:GetBucket* - s3:List *- s3:Abort* - s3:DeleteObject *- s3:PutObject* Resource: “*” Condition: StringNotEquals: s3:ResourceAccount: Ref: {aws}::AccountId - Sid: PipelineCrossAccountArtifactsKey Effect: Allow Action: - kms:Decrypt - kms:DescribeKey - kms:Encrypt - kms:ReEncrypt* - kms:GenerateDataKey\* Resource: “\_” Condition: StringEquals: kms:ViaService: Fn::Sub: s3.${{aws}::Region}.amazonaws.com … — + [#bootstrapping-securityhub-kms2-why] \*Why does Security Hub flag this?\_::: + The policies contain a Resource: \\* combined with a Condition clause. Security Hub flags the \* wildcard. This wildcard is used because at the time the account is bootstrapped, the {aws} KMS key created by CDK Pipelines for the CodePipeline artifact bucket does not exist yet, and therefore, can't be referenced on the bootstrap template by ARN. In addition, Security Hub does not consider theConditionclause when raising this flag. ThisConditionrestrictsResource: *+ to requests made from the same {aws} account of the {aws} KMS key. These requests must come from Amazon S3 in the same {aws} Region as the {aws} KMS key. + [#bootstrapping-securityhub-kms2-decide] \*Do I need to fix this finding?\*::: + As long as you have not modified the {aws} KMS key on your bootstrap template to be overly permissive, the \_deploy role\_ does not allow more access than it needs. Therefore, it is not necessary to fix this finding. + [#bootstrapping-securityhub-kms2-fix] \*What if I want to fix this finding?\*::: + How you fix this finding depends on whether or not you will be using CDK Pipelines for cross-account deployments. + \*To fix the Security Hub finding and use CDK Pipelines for cross-account deployments\*:::: + . If you have not done so, deploy the CDK bootstrap stack using the +cdk bootstrapcommand. . If you have not done so, create and deploy your CDK [.noloc]```Pipeline```+. For instructions, see xref:cdk-pipeline[Continuous integration and delivery (CI/CD) using CDK Pipelines]. . Obtain the {aws} KMS key ARN of the CodePipeline artifact bucket. This resource is created during pipeline creation. . Obtain a copy of the CDK bootstrap template to modify it. The following is an example, using the {aws} CDK CLI: + [source,bash,subs="verbatim,attributes"] ---- $ cdk bootstrap --show-template > bootstrap-template.yaml ---- . Modify the template by replacing +Resource:* of thePipelineCrossAccountArtifactsKey` statement with your ARN value. . Deploy the template to update your bootstrap stack. The following is an example, using the CDK CLI: + [source,bash,subs=“verbatim,attributes”] — $ cdk bootstrap aws://++++++/++++++–template bootstrap-template.yaml —- +++++++++++++

*To fix the Security Hub finding if you’re not using CDK Pipelines for cross-account deployments*:::: + . Obtain a copy of the CDK bootstrap template to modify it. The following is an example, using the CDK CLI: + [source,bash,subs=“verbatim,attributes”] — $ cdk bootstrap –show-template > bootstrap-template.yaml — . Delete the PipelineCrossAccountArtifactsBucket and PipelineCrossAccountArtifactsKey statements from the template. . Deploy the template to update your bootstrap stack. The following is an example, using the CDK CLI: + [source,bash,subs=“verbatim,attributes”] — $ cdk bootstrap aws://++++++/++++++–template bootstrap-template.yaml —-++++++++++++

[#bootstrapping-env-considerations] == Considerations

Since bootstrapping provisions resources in your environment, you may incur {aws} charges when those resources are used with the {aws} CDK.

include::bootstrapping-customizing.adoc[leveloffset=+1]

include::customize-permissions-boundaries.adoc[leveloffset=+1]

include::bootstrapping-troubleshoot.adoc[leveloffset=+1]