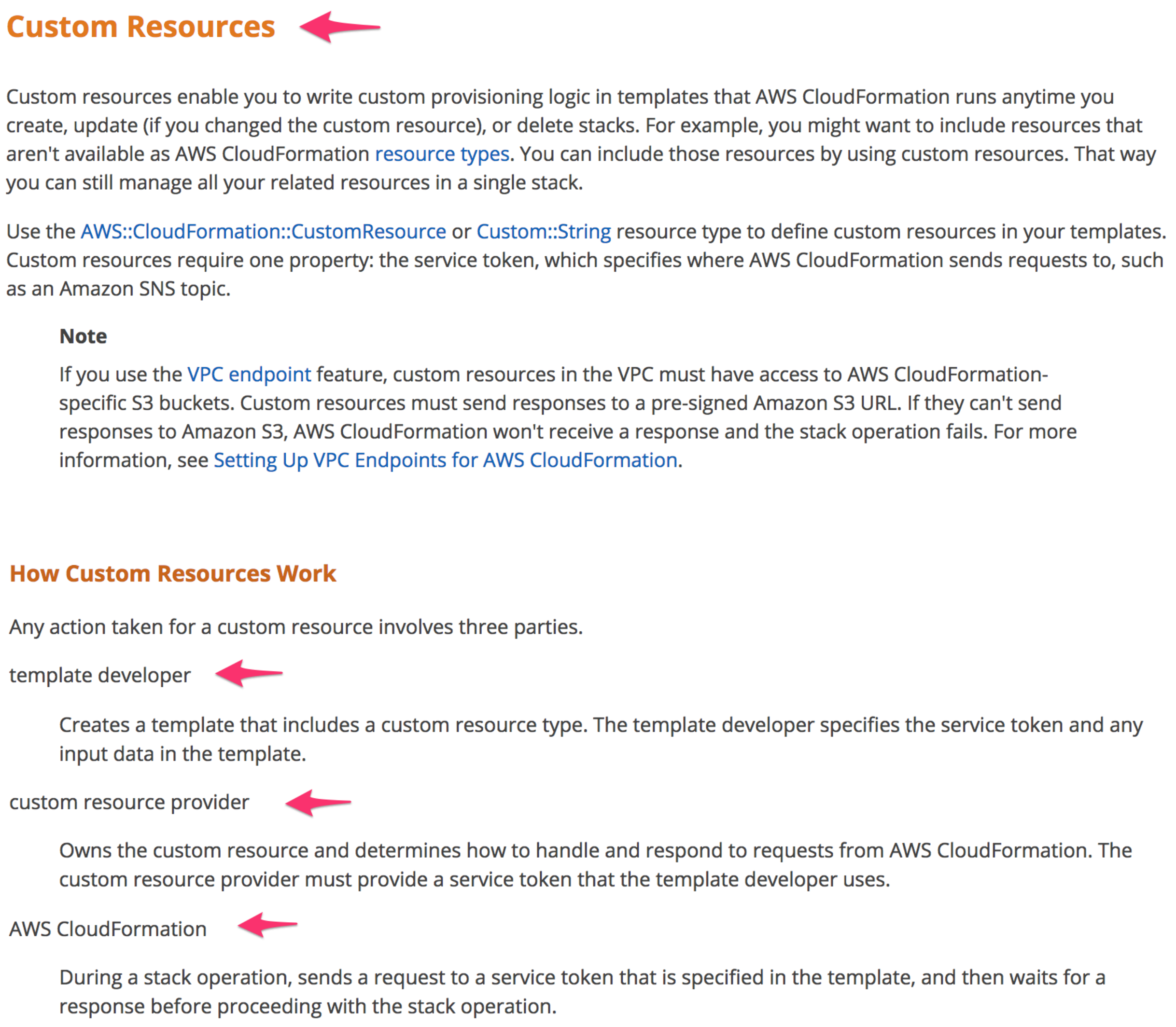
CloudFormation - Custom Resources

https://docs.aws.amazon.com/AWSCloudFormation/latest/UserGuide/walkthrough-custom-resources-lambda-lookup-amiids.html



Custom Resources

Custom resources enable you to write custom provisioning logic in templates that AWS CloudFormation runs anytime you create, update (if you changed the custom resource), or delete stacks. For example, you might want to include resources that aren't available as AWS CloudFormation resource types. You can include those resources by using custom resources. That way you can still manage all your related resources in a single stack.

Use the AWS::CloudFormation::CustomResource or Custom::String resource type to define custom resources in your templates. Custom resources require one property: the service token, which specifies where AWS CloudFormation sends requests to, such as an Amazon SNS topic.

> **Note**
>
> If you use the VPC endpoint feature, custom resources in the VPC must have access to AWS CloudFormation-specific S3 buckets. Custom resources must send responses to a pre-signed Amazon S3 URL. If they can't send responses to Amazon S3, AWS CloudFormation won't receive a response and the stack operation fails. For more information, see Setting Up VPC Endpoints for AWS CloudFormation.

How Custom Resources Work

Any action taken for a custom resource involves three parties.

template developer

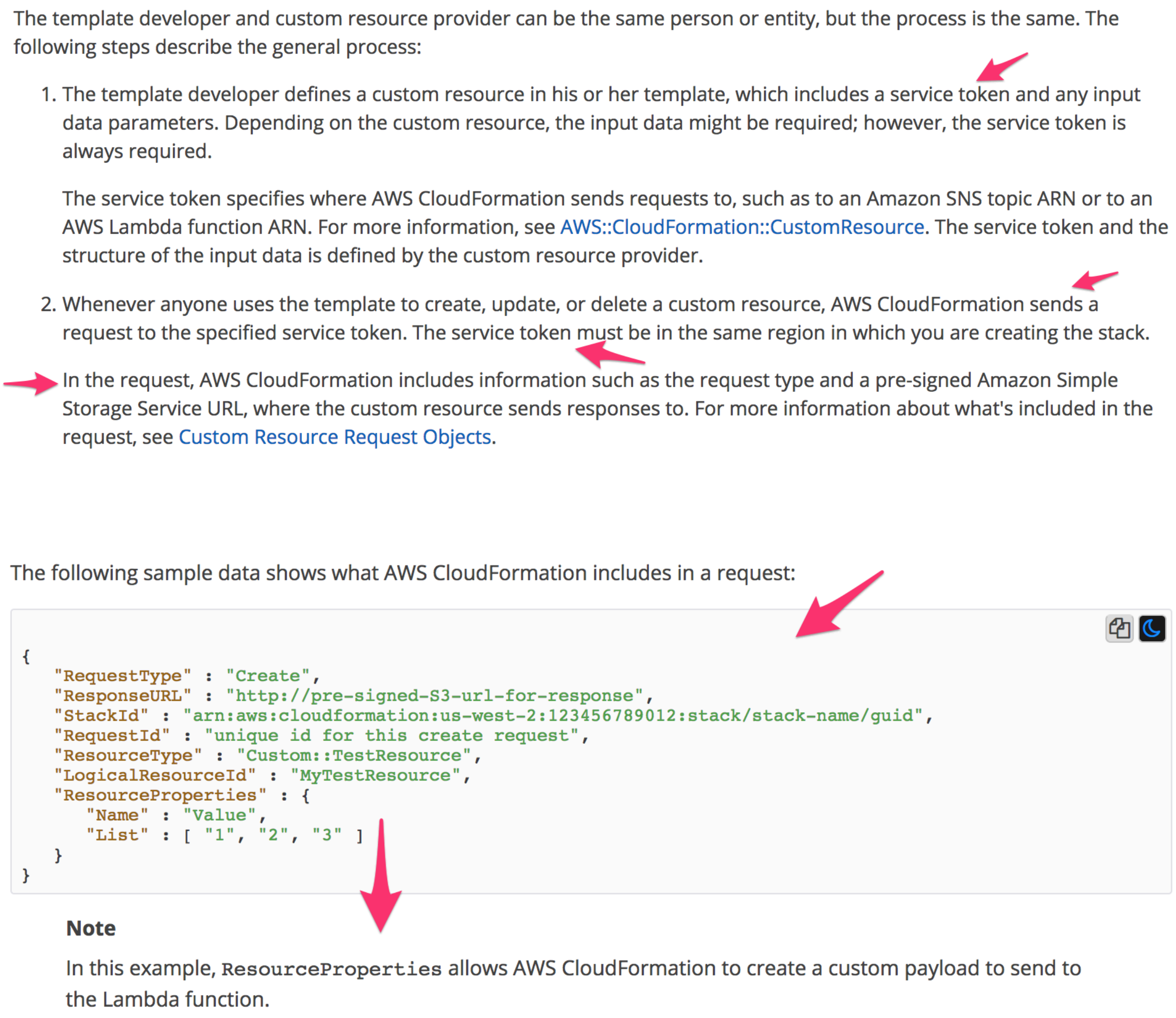
Creates a template that includes a custom resource type. The template developer specifies the service token and any input data in the template.

custom resource provider

Owns the custom resource and determines how to handle and respond to requests from AWS CloudFormation. The custom resource provider must provide a service token that the template developer uses.

AWS CloudFormation

During a stack operation, sends a request to a service token that is specified in the template, and then waits for a response before proceeding with the stack operation.



The template developer and custom resource provider can be the same person or entity, but the process is the same. The following steps describe the general process:

1. The template developer defines a custom resource in his or her template, which includes a service token and any input data parameters. Depending on the custom resource, the input data might be required; however, the service token is always required.

   The service token specifies where AWS CloudFormation sends requests to, such as to an Amazon SNS topic ARN or to an AWS Lambda function ARN. For more information, see AWS::CloudFormation::CustomResource. The service token and the structure of the input data is defined by the custom resource provider.

2. Whenever anyone uses the template to create, update, or delete a custom resource, AWS CloudFormation sends a request to the specified service token. The service token must be in the same region in which you are creating the stack.

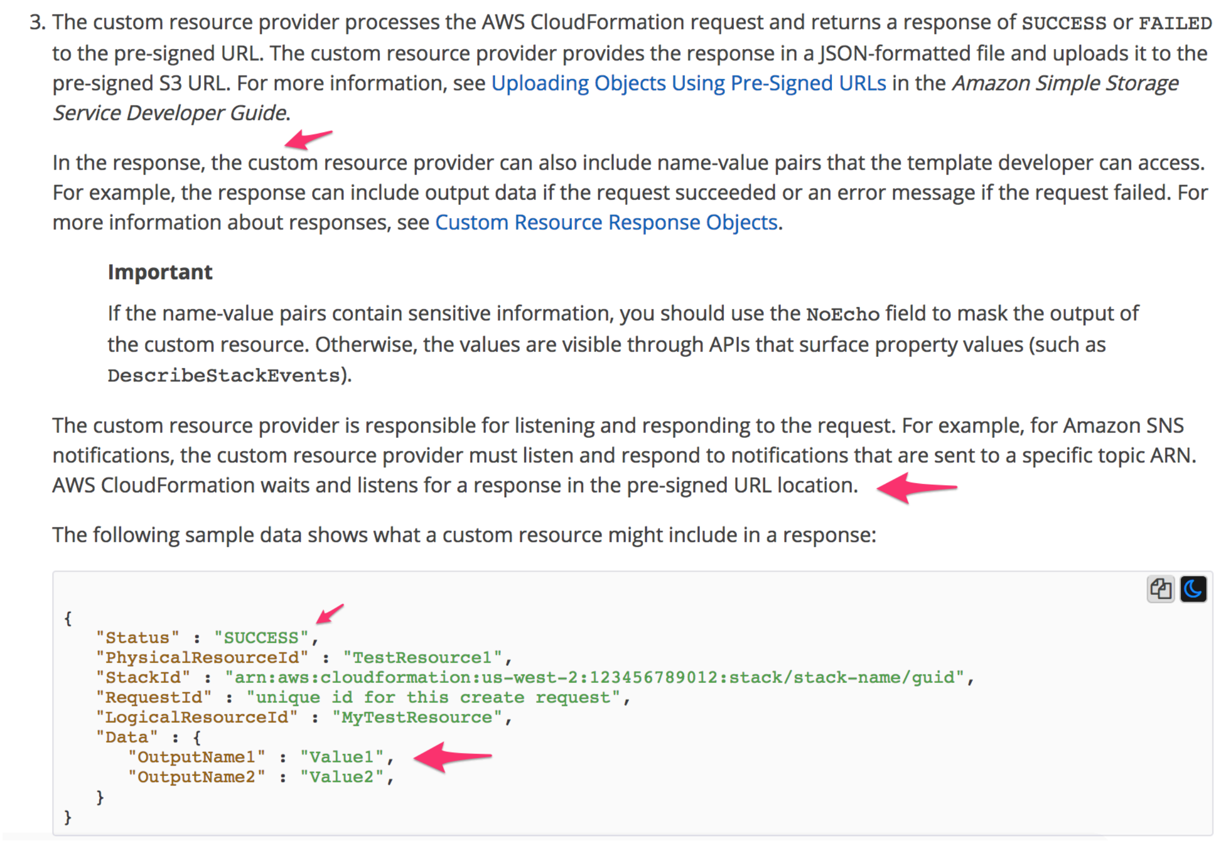
   In the request, AWS CloudFormation includes information such as the request type and a pre-signed Amazon Simple Storage Service URL, where the custom resource sends responses to. For more information about what's included in the request, see Custom Resource Request Objects.

The following sample data shows what AWS CloudFormation includes in a request:

```
{
   "RequestType" : "Create",
   "ResponseURL" : "http://pre-signed-S3-url-for-response",
   "StackId" : "arn:aws:cloudformation:us-west-2:123456789012:stack/stack-name/guid",
   "RequestId" : "unique id for this create request",
   "ResourceType" : "Custom::TestResource",
   "LogicalResourceId" : "MyTestResource",
   "ResourceProperties" : {
      "Name" : "Value",
      "List" : [ "1", "2", "3" ]
   }
}
```

Note

In this example, `ResourceProperties` allows AWS CloudFormation to create a custom payload to send to the Lambda function.



3. The custom resource provider processes the AWS CloudFormation request and returns a response of `SUCCESS` or `FAILED` to the pre-signed URL. The custom resource provider provides the response in a JSON-formatted file and uploads it to the pre-signed S3 URL. For more information, see Uploading Objects Using Pre-Signed URLs in the *Amazon Simple Storage Service Developer Guide*.

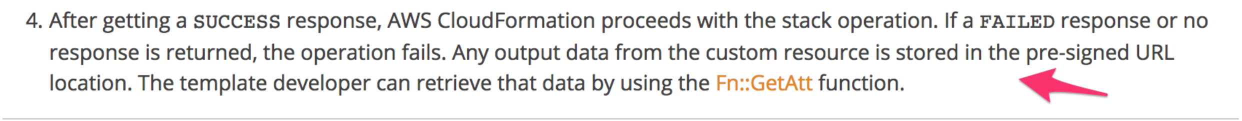
   In the response, the custom resource provider can also include name-value pairs that the template developer can access. For example, the response can include output data if the request succeeded or an error message if the request failed. For more information about responses, see Custom Resource Response Objects.

   > **Important**
   >
   > If the name-value pairs contain sensitive information, you should use the `NoEcho` field to mask the output of the custom resource. Otherwise, the values are visible through APIs that surface property values (such as `DescribeStackEvents`).

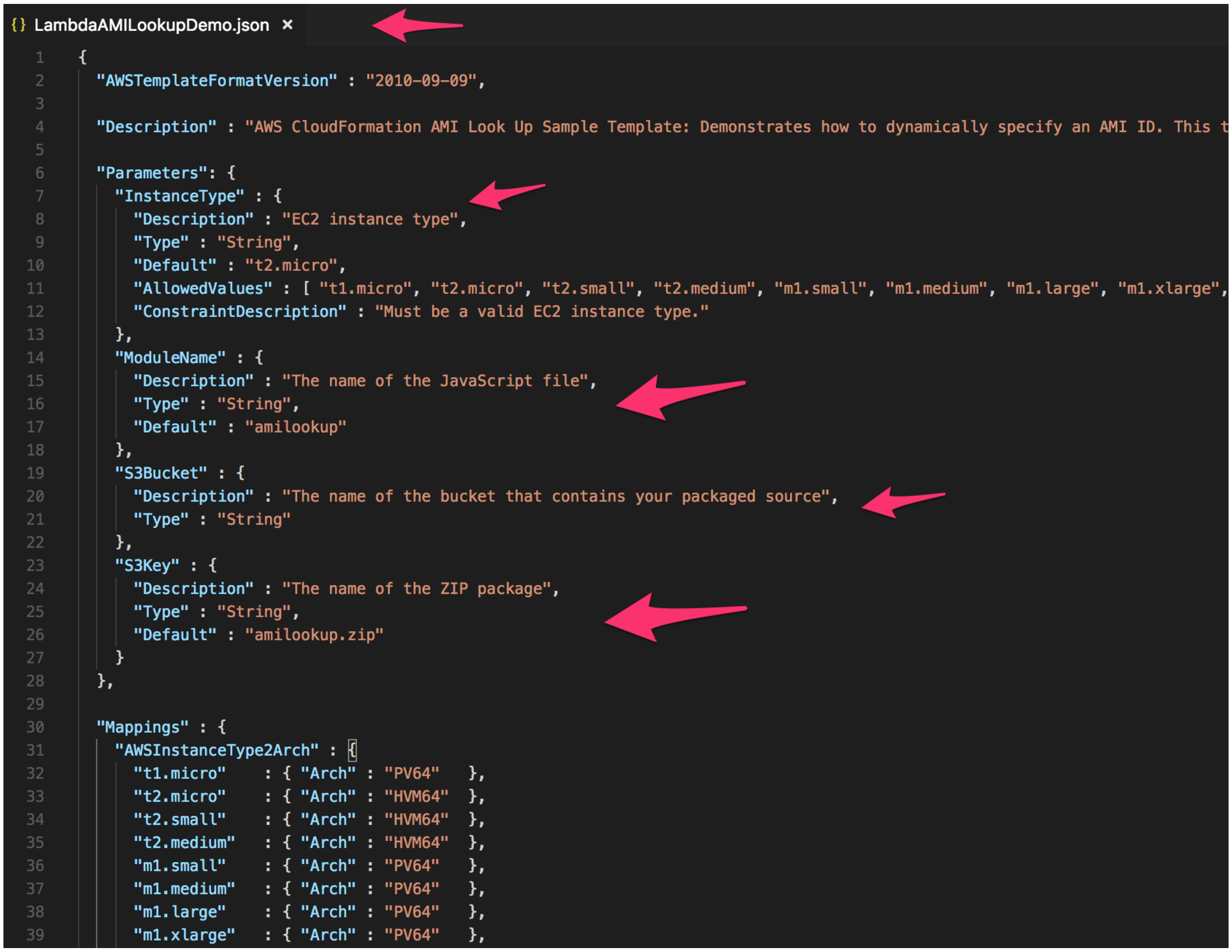
   The custom resource provider is responsible for listening and responding to the request. For example, for Amazon SNS notifications, the custom resource provider must listen and respond to notifications that are sent to a specific topic ARN. AWS CloudFormation waits and listens for a response in the pre-signed URL location.

   The following sample data shows what a custom resource might include in a response:

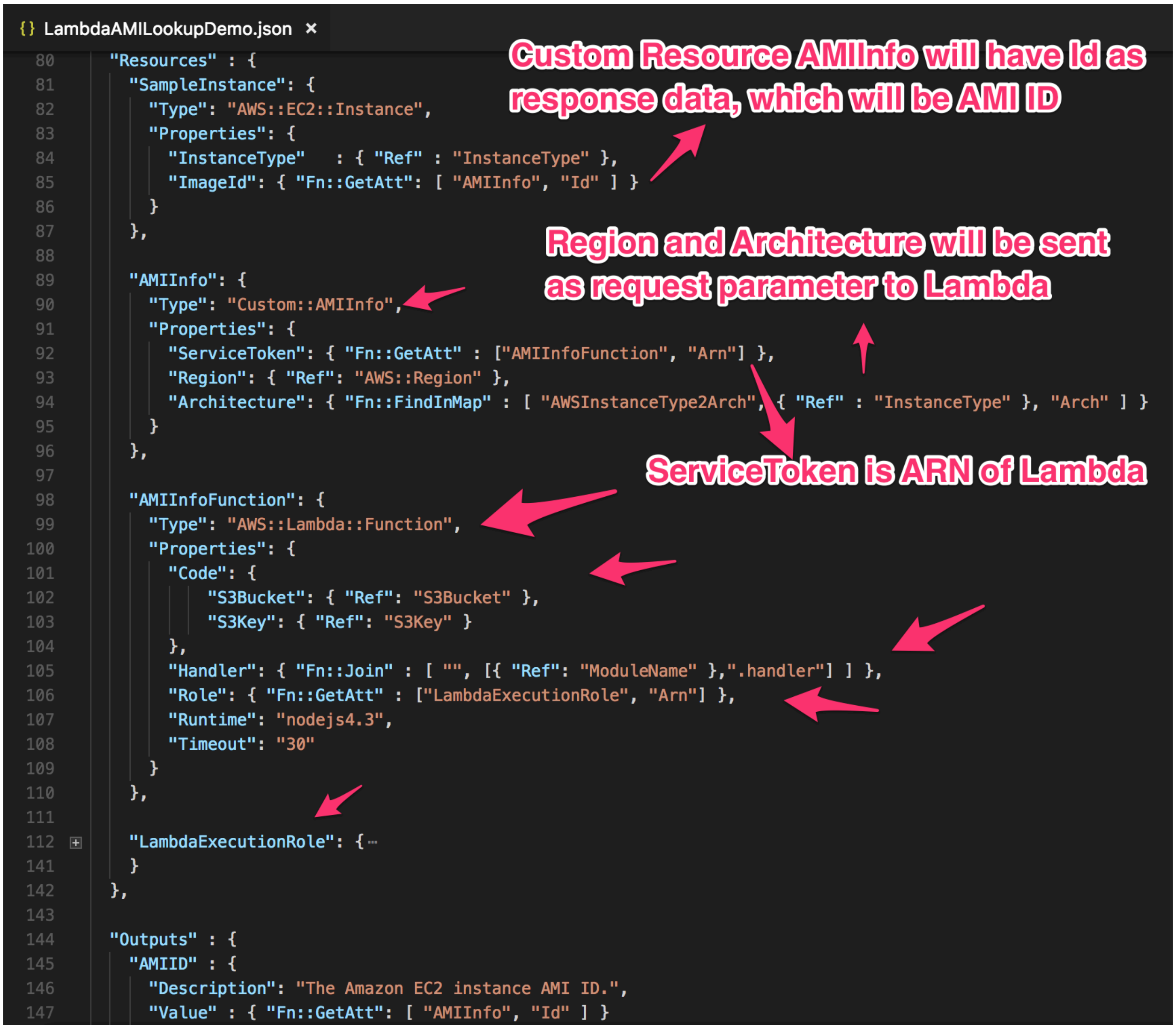
```
{
   "Status" : "SUCCESS",
   "PhysicalResourceId" : "TestResource1",
   "StackId" : "arn:aws:cloudformation:us-west-2:123456789012:stack/stack-name/guid",
   "RequestId" : "unique id for this create request",
   "LogicalResourceId" : "MyTestResource",
   "Data" : {
      "OutputName1" : "Value1",
      "OutputName2" : "Value2",
   }
}
```



4. After getting a `SUCCESS` response, AWS CloudFormation proceeds with the stack operation. If a `FAILED` response or no response is returned, the operation fails. Any output data from the custom resource is stored in the pre-signed URL location. The template developer can retrieve that data by using the Fn::GetAtt function.



```
{
  "AWSTemplateFormatVersion" : "2010-09-09",

  "Description" : "AWS CloudFormation AMI Look Up Sample Template: Demonstrates how to dynamically specify an AMI ID. This t

  "Parameters": {
    "InstanceType" : {
      "Description" : "EC2 instance type",
      "Type" : "String",
      "Default" : "t2.micro",
      "AllowedValues" : [ "t1.micro", "t2.micro", "t2.small", "t2.medium", "m1.small", "m1.medium", "m1.large", "m1.xlarge",
      "ConstraintDescription" : "Must be a valid EC2 instance type."
    },
    "ModuleName" : {
      "Description" : "The name of the JavaScript file",
      "Type" : "String",
      "Default" : "amilookup"
    },
    "S3Bucket" : {
      "Description" : "The name of the bucket that contains your packaged source",
      "Type" : "String"
    },
    "S3Key" : {
      "Description" : "The name of the ZIP package",
      "Type" : "String",
      "Default" : "amilookup.zip"
    }
  },

  "Mappings" : {
    "AWSInstanceType2Arch" : {
      "t1.micro"    : { "Arch" : "PV64"   },
      "t2.micro"    : { "Arch" : "HVM64"  },
      "t2.small"    : { "Arch" : "HVM64"  },
      "t2.medium"   : { "Arch" : "HVM64"  },
      "m1.small"    : { "Arch" : "PV64"   },
      "m1.medium"   : { "Arch" : "PV64"   },
      "m1.large"    : { "Arch" : "PV64"   },
      "m1.xlarge"   : { "Arch" : "PV64"   },
```



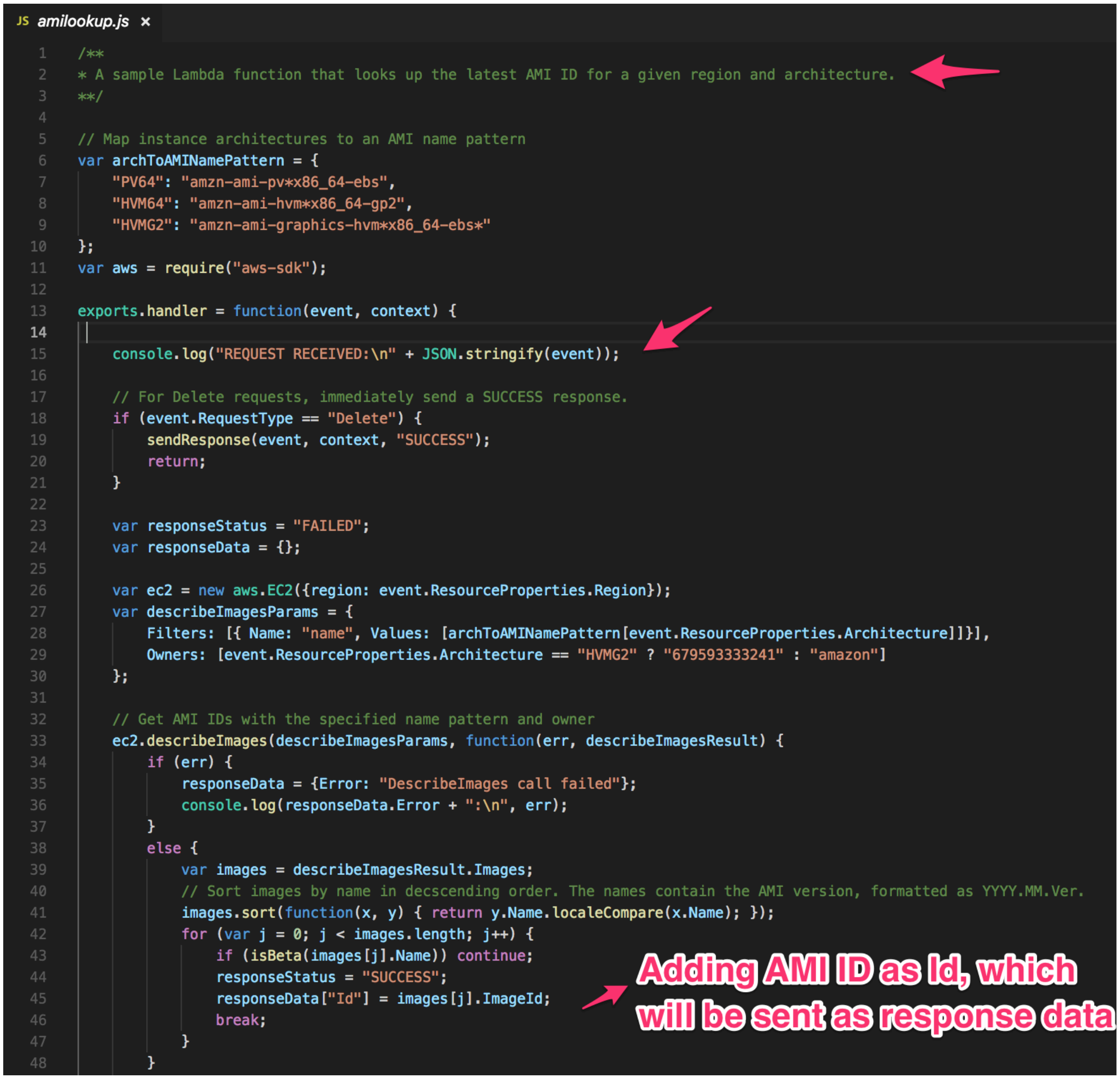
```
  "Resources" : {
    "SampleInstance": {
      "Type": "AWS::EC2::Instance",
      "Properties": {
        "InstanceType"   : { "Ref" : "InstanceType" },
        "ImageId": { "Fn::GetAtt": [ "AMIInfo", "Id" ] }
      }
    },

    "AMIInfo": {
      "Type": "Custom::AMIInfo",
      "Properties": {
        "ServiceToken": { "Fn::GetAtt" : ["AMIInfoFunction", "Arn"] },
        "Region": { "Ref": "AWS::Region" },
        "Architecture": { "Fn::FindInMap" : [ "AWSInstanceType2Arch", { "Ref" : "InstanceType" }, "Arch" ] }
      }
    },

    "AMIInfoFunction": {
      "Type": "AWS::Lambda::Function",
      "Properties": {
        "Code": {
            "S3Bucket": { "Ref": "S3Bucket" },
            "S3Key": { "Ref": "S3Key" }
        },
        "Handler": { "Fn::Join" : [ "", [{ "Ref": "ModuleName" },".handler"] ] },
        "Role": { "Fn::GetAtt" : ["LambdaExecutionRole", "Arn"] },
        "Runtime": "nodejs4.3",
        "Timeout": "30"
      }
    },

    "LambdaExecutionRole": {⋯
    }
  },

  "Outputs" : {
    "AMIID" : {
      "Description": "The Amazon EC2 instance AMI ID.",
      "Value" : { "Fn::GetAtt": [ "AMIInfo", "Id" ] }
```

Custom Resource AMIInfo will have Id as response data, which will be AMI ID

Region and Architecture will be sent as request parameter to Lambda

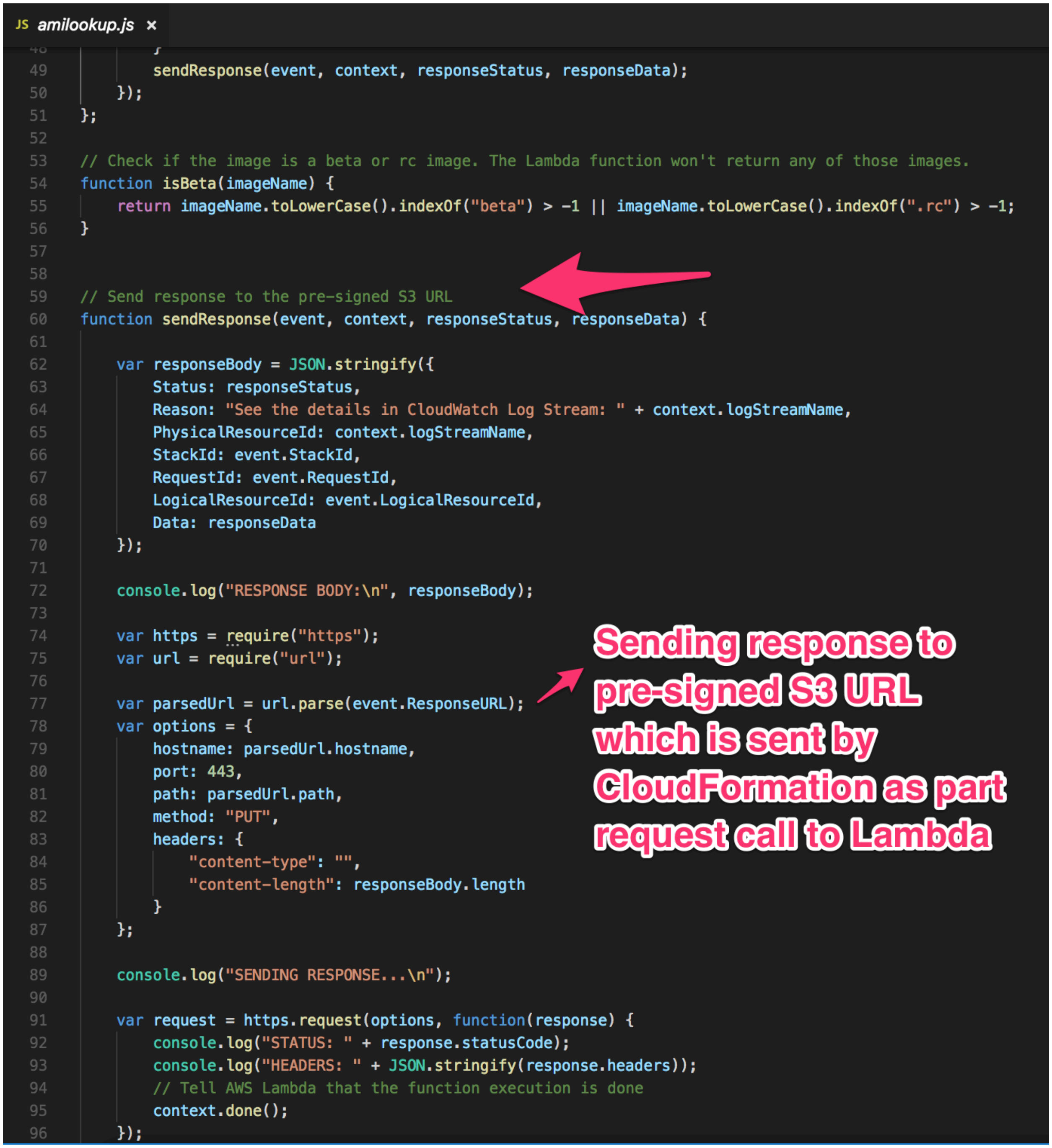
ServiceToken is ARN of Lambda

AMILookup Lambda Code



```javascript
/**
* A sample Lambda function that looks up the latest AMI ID for a given region and architecture.
**/

// Map instance architectures to an AMI name pattern
var archToAMINamePattern = {
    "PV64": "amzn-ami-pv*x86_64-ebs",
    "HVM64": "amzn-ami-hvm*x86_64-gp2",
    "HVMG2": "amzn-ami-graphics-hvm*x86_64-ebs*"
};
var aws = require("aws-sdk");

exports.handler = function(event, context) {

    console.log("REQUEST RECEIVED:\n" + JSON.stringify(event));

    // For Delete requests, immediately send a SUCCESS response.
    if (event.RequestType == "Delete") {
        sendResponse(event, context, "SUCCESS");
        return;
    }

    var responseStatus = "FAILED";
    var responseData = {};

    var ec2 = new aws.EC2({region: event.ResourceProperties.Region});
    var describeImagesParams = {
        Filters: [{ Name: "name", Values: [archToAMINamePattern[event.ResourceProperties.Architecture]]}],
        Owners: [event.ResourceProperties.Architecture == "HVMG2" ? "679593333241" : "amazon"]
    };

    // Get AMI IDs with the specified name pattern and owner
    ec2.describeImages(describeImagesParams, function(err, describeImagesResult) {
        if (err) {
            responseData = {Error: "DescribeImages call failed"};
            console.log(responseData.Error + ":\n", err);
        }
        else {
            var images = describeImagesResult.Images;
            // Sort images by name in decscending order. The names contain the AMI version, formatted as YYYY.MM.Ver.
            images.sort(function(x, y) { return y.Name.localeCompare(x.Name); });
            for (var j = 0; j < images.length; j++) {
                if (isBeta(images[j].Name)) continue;
                responseStatus = "SUCCESS";
                responseData["Id"] = images[j].ImageId;
                break;
            }
        }
```

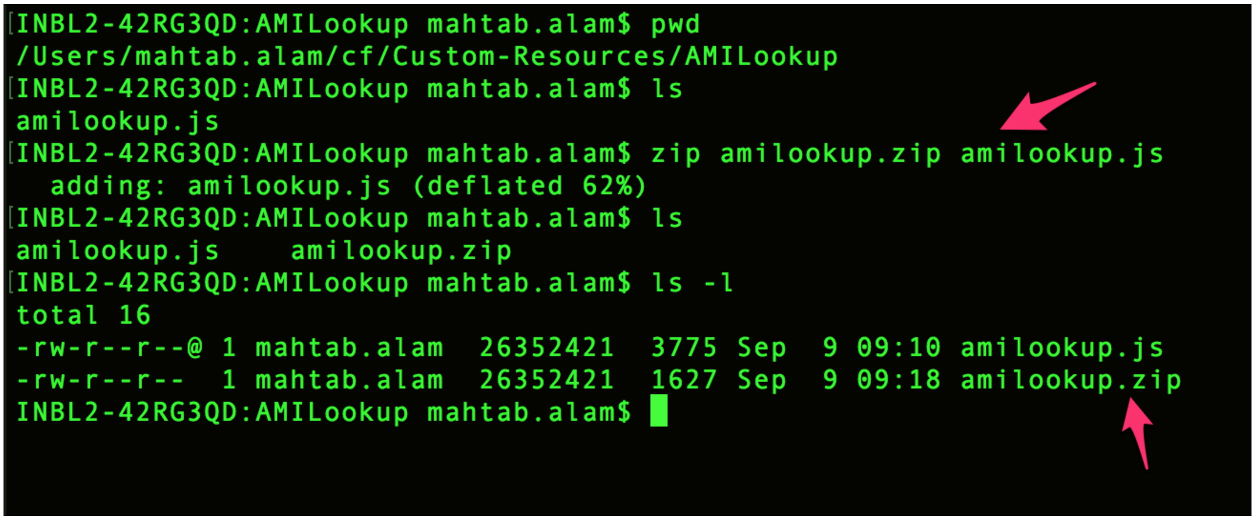
Adding AMI ID as Id, which will be sent as response data



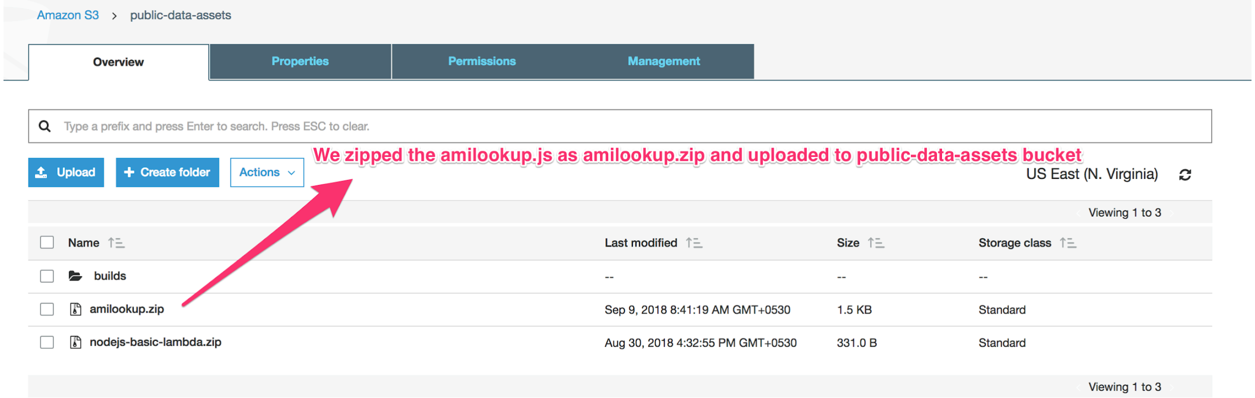
```
        }
        sendResponse(event, context, responseStatus, responseData);
    });
};

// Check if the image is a beta or rc image. The Lambda function won't return any of those images.
function isBeta(imageName) {
    return imageName.toLowerCase().indexOf("beta") > -1 || imageName.toLowerCase().indexOf(".rc") > -1;
}


// Send response to the pre-signed S3 URL
function sendResponse(event, context, responseStatus, responseData) {

    var responseBody = JSON.stringify({
        Status: responseStatus,
        Reason: "See the details in CloudWatch Log Stream: " + context.logStreamName,
        PhysicalResourceId: context.logStreamName,
        StackId: event.StackId,
        RequestId: event.RequestId,
        LogicalResourceId: event.LogicalResourceId,
        Data: responseData
    });

    console.log("RESPONSE BODY:\n", responseBody);

    var https = require("https");
    var url = require("url");

    var parsedUrl = url.parse(event.ResponseURL);
    var options = {
        hostname: parsedUrl.hostname,
        port: 443,
        path: parsedUrl.path,
        method: "PUT",
        headers: {
            "content-type": "",
            "content-length": responseBody.length
        }
    };

    console.log("SENDING RESPONSE...\n");

    var request = https.request(options, function(response) {
        console.log("STATUS: " + response.statusCode);
        console.log("HEADERS: " + JSON.stringify(response.headers));
        // Tell AWS Lambda that the function execution is done
        context.done();
    });
```

Sending response to pre-signed S3 URL which is sent by CloudFormation as part request call to Lambda

Next let’s zip the amilookup.js into amilookup.zip and upload to S3 bucket



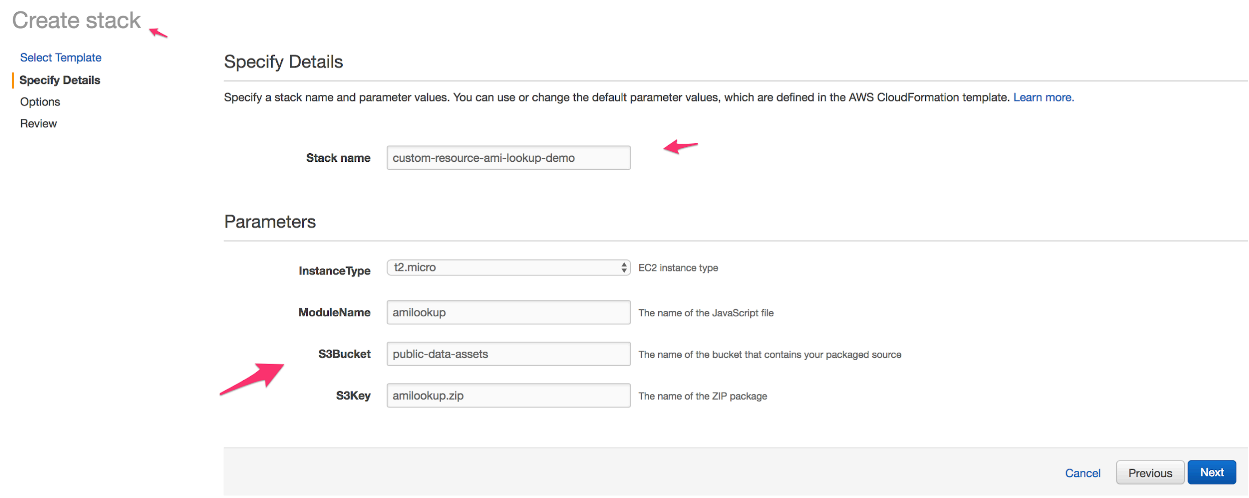
```
[INBL2-42RG3QD:AMILookup mahtab.alam$ pwd
/Users/mahtab.alam/cf/Custom-Resources/AMILookup
[INBL2-42RG3QD:AMILookup mahtab.alam$ ls
amilookup.js
[INBL2-42RG3QD:AMILookup mahtab.alam$ zip amilookup.zip amilookup.js
  adding: amilookup.js (deflated 62%)
[INBL2-42RG3QD:AMILookup mahtab.alam$ ls
amilookup.js    amilookup.zip
[INBL2-42RG3QD:AMILookup mahtab.alam$ ls -l
total 16
-rw-r--r--@ 1 mahtab.alam  26352421  3775 Sep  9 09:10 amilookup.js
-rw-r--r--  1 mahtab.alam  26352421  1627 Sep  9 09:18 amilookup.zip
INBL2-42RG3QD:AMILookup mahtab.alam$ 
```



We zipped the amilookup.js as amilookup.zip and uploaded to public-data-assets bucket

□ Name	Last modified	Size	Storage class
□ builds	--	--	--
□ amilookup.zip	Sep 9, 2018 8:41:19 AM GMT+0530	1.5 KB	Standard
□ nodejs-basic-lambda.zip	Aug 30, 2018 4:32:55 PM GMT+0530	331.0 B	Standard

Next let’s create stack



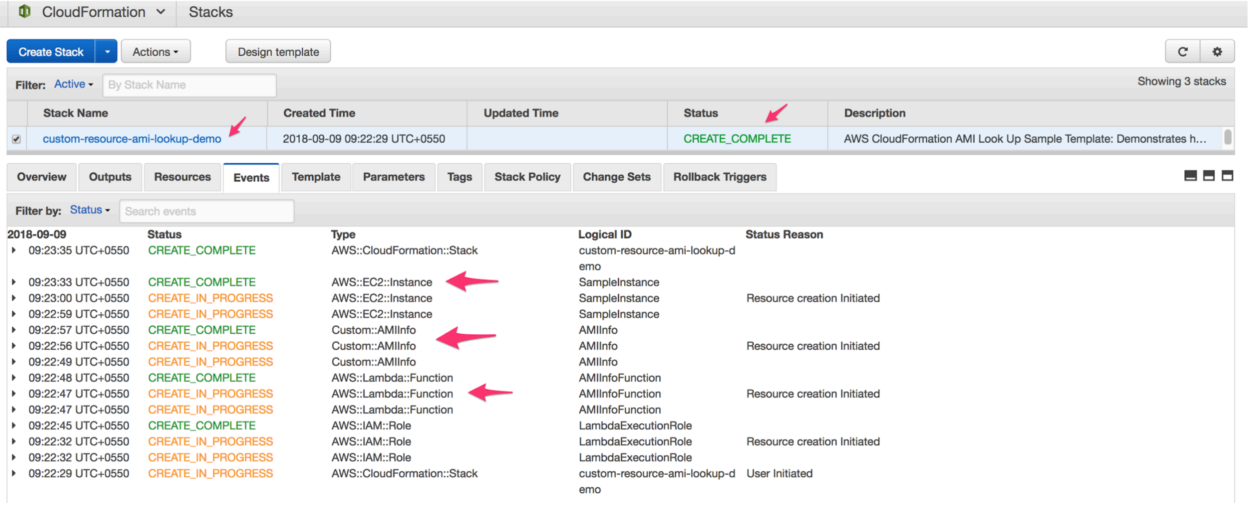
Specify Details

Specify a stack name and parameter values. You can use or change the default parameter values, which are defined in the AWS CloudFormation template. Learn more.

Stack name: custom-resource-ami-lookup-demo

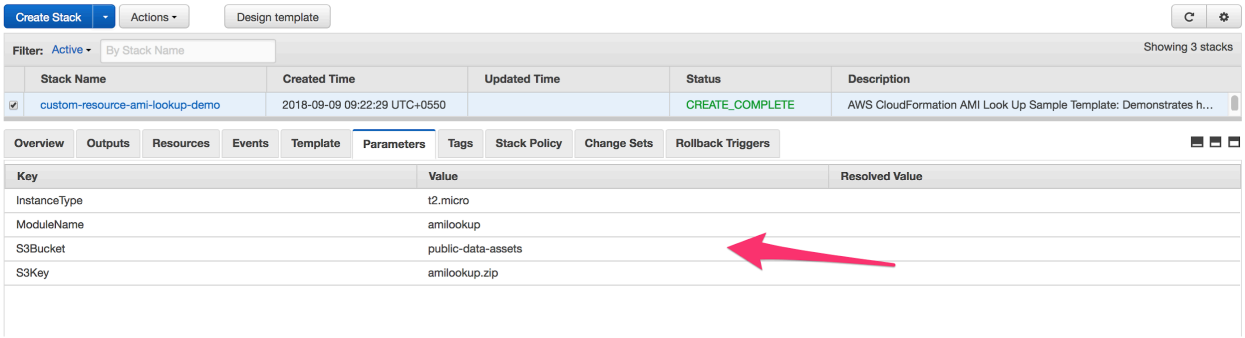
Parameters

InstanceType	t2.micro	EC2 instance type
ModuleName	amilookup	The name of the JavaScript file
S3Bucket	public-data-assets	The name of the bucket that contains your packaged source
S3Key	amilookup.zip	The name of the ZIP package



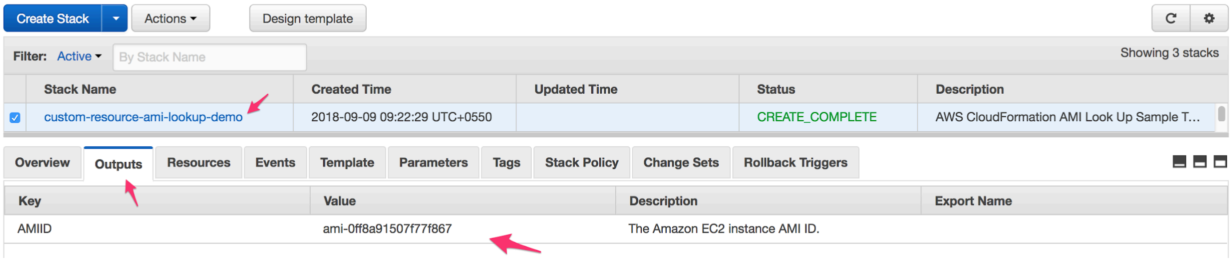
	Stack Name	Created Time	Updated Time	Status	Description
☑	custom-resource-ami-lookup-demo	2018-09-09 09:22:29 UTC+0550		CREATE_COMPLETE	AWS CloudFormation AMI Look Up Sample Template: Demonstrates h...

2018-09-09	Status	Type	Logical ID	Status Reason
09:23:35 UTC+0550	CREATE_COMPLETE	AWS::CloudFormation::Stack	custom-resource-ami-lookup-demo	
09:23:33 UTC+0550	CREATE_COMPLETE	AWS::EC2::Instance	SampleInstance	
09:23:00 UTC+0550	CREATE_IN_PROGRESS	AWS::EC2::Instance	SampleInstance	Resource creation Initiated
09:22:59 UTC+0550	CREATE_IN_PROGRESS	AWS::EC2::Instance	SampleInstance	
09:22:57 UTC+0550	CREATE_COMPLETE	Custom::AMIInfo	AMIInfo	
09:22:56 UTC+0550	CREATE_IN_PROGRESS	Custom::AMIInfo	AMIInfo	Resource creation Initiated
09:22:49 UTC+0550	CREATE_IN_PROGRESS	Custom::AMIInfo	AMIInfo	
09:22:48 UTC+0550	CREATE_COMPLETE	AWS::Lambda::Function	AMIInfoFunction	
09:22:47 UTC+0550	CREATE_IN_PROGRESS	AWS::Lambda::Function	AMIInfoFunction	Resource creation Initiated
09:22:47 UTC+0550	CREATE_IN_PROGRESS	AWS::Lambda::Function	AMIInfoFunction	
09:22:45 UTC+0550	CREATE_COMPLETE	AWS::IAM::Role	LambdaExecutionRole	
09:22:32 UTC+0550	CREATE_IN_PROGRESS	AWS::IAM::Role	LambdaExecutionRole	Resource creation Initiated
09:22:32 UTC+0550	CREATE_IN_PROGRESS	AWS::IAM::Role	LambdaExecutionRole	
09:22:29 UTC+0550	CREATE_IN_PROGRESS	AWS::CloudFormation::Stack	custom-resource-ami-lookup-demo	User Initiated



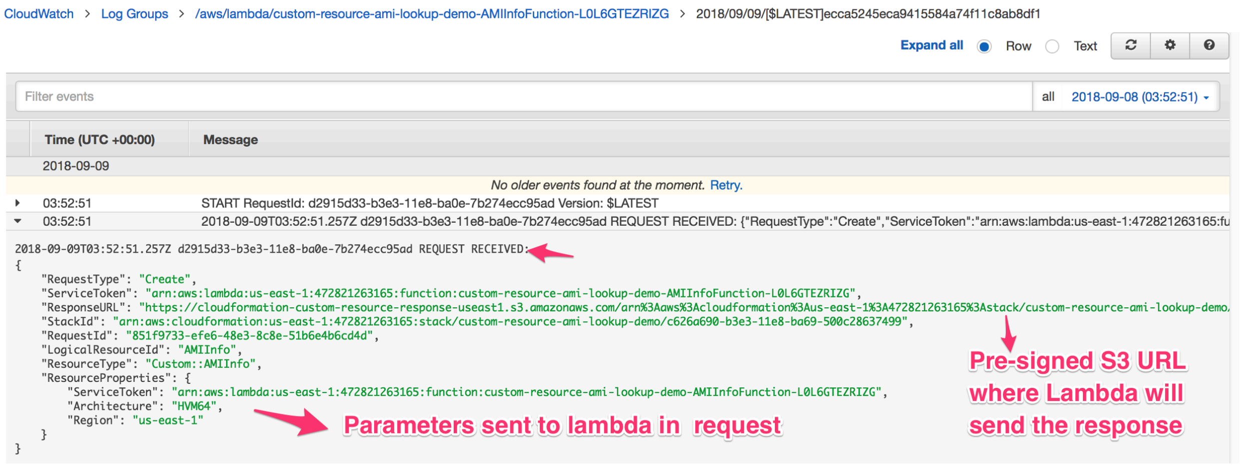
	Stack Name	Created Time	Updated Time	Status	Description
☑	custom-resource-ami-lookup-demo	2018-09-09 09:22:29 UTC+0550		CREATE_COMPLETE	AWS CloudFormation AMI Look Up Sample Template: Demonstrates h...

Key	Value	Resolved Value
InstanceType	t2.micro	
ModuleName	amilookup	
S3Bucket	public-data-assets	
S3Key	amilookup.zip	



	Stack Name	Created Time	Updated Time	Status	Description
☑	custom-resource-ami-lookup-demo	2018-09-09 09:22:29 UTC+0550		CREATE_COMPLETE	AWS CloudFormation AMI Look Up Sample T...

Key	Value	Description	Export Name
AMIID	ami-0ff8a91507f77f867	The Amazon EC2 instance AMI ID.	



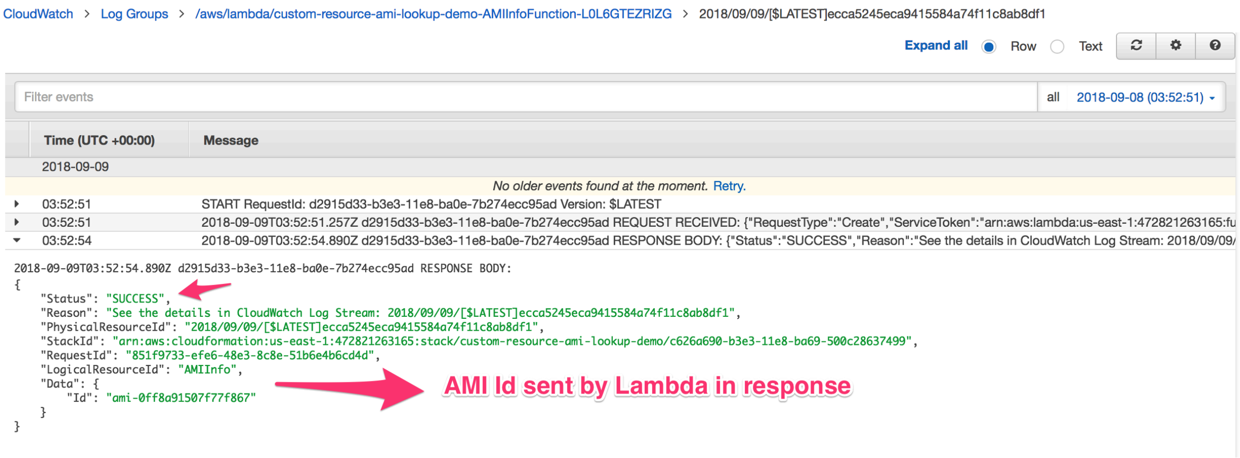
CloudWatch > Log Groups > /aws/lambda/custom-resource-ami-lookup-demo-AMIInfoFunction-L0L6GTEZRIZG > 2018/09/09/[$LATEST]ecca5245eca9415584a74f11c8ab8df1
Expand all ● Row ○ Text
Filter events
all 2018-09-08 (03:52:51)

Time (UTC +00:00)	Message
2018-09-09	
	No older events found at the moment. Retry.
03:52:51	START RequestId: d2915d33-b3e3-11e8-ba0e-7b274ecc95ad Version: $LATEST
03:52:51	2018-09-09T03:52:51.257Z d2915d33-b3e3-11e8-ba0e-7b274ecc95ad REQUEST RECEIVED: {"RequestType":"Create","ServiceToken":"arn:aws:lambda:us-east-1:472821263165:fu

```
2018-09-09T03:52:51.257Z d2915d33-b3e3-11e8-ba0e-7b274ecc95ad REQUEST RECEIVED:
{
    "RequestType": "Create",
    "ServiceToken": "arn:aws:lambda:us-east-1:472821263165:function:custom-resource-ami-lookup-demo-AMIInfoFunction-L0L6GTEZRIZG",
    "ResponseURL": "https://cloudformation-custom-resource-response-useast1.s3.amazonaws.com/arn%3Aaws%3Acloudformation%3Aus-east-1%3A472821263165%3Astack/custom-resource-ami-lookup-demo/
    "StackId": "arn:aws:cloudformation:us-east-1:472821263165:stack/custom-resource-ami-lookup-demo/c626a690-b3e3-11e8-ba69-500c28637499",
    "RequestId": "851f9733-efe6-48e3-8c8e-51b6e4b6cd4d",
    "LogicalResourceId": "AMIInfo",
    "ResourceType": "Custom::AMIInfo",
    "ResourceProperties": {
        "ServiceToken": "arn:aws:lambda:us-east-1:472821263165:function:custom-resource-ami-lookup-demo-AMIInfoFunction-L0L6GTEZRIZG",
        "Architecture": "HVM64",
        "Region": "us-east-1"
    }
}
```

Parameters sent to lambda in request

Pre-signed S3 URL where Lambda will send the response



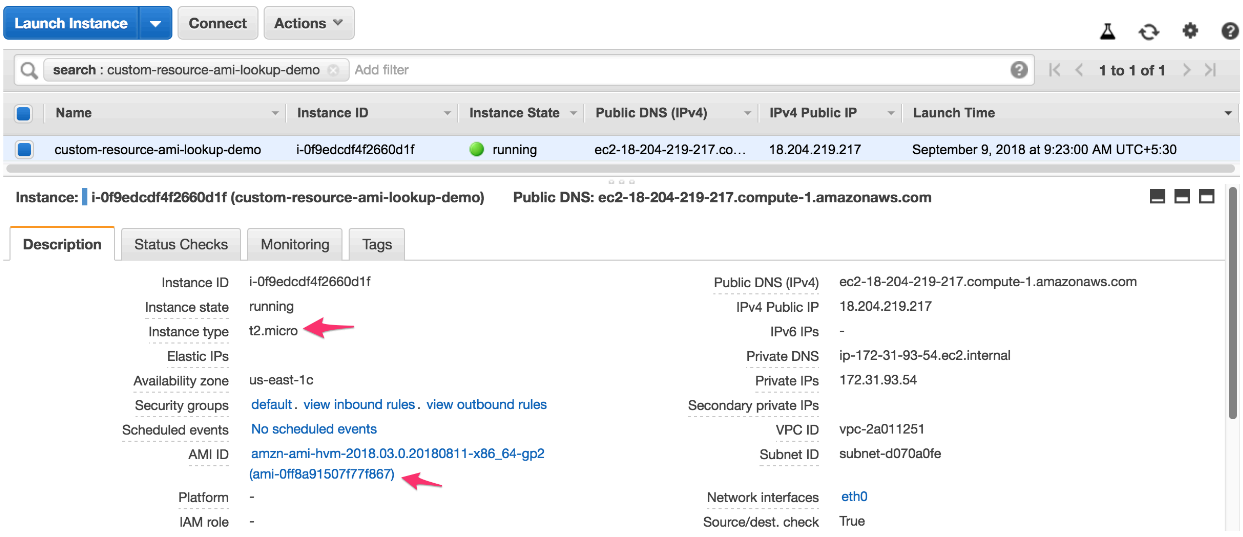
CloudWatch > Log Groups > /aws/lambda/custom-resource-ami-lookup-demo-AMIInfoFunction-L0L6GTEZRIZG > 2018/09/09/[$LATEST]ecca5245eca9415584a74f11c8ab8df1
Expand all ● Row ○ Text
Filter events
all 2018-09-08 (03:52:51)

Time (UTC +00:00)	Message
2018-09-09	
	No older events found at the moment. Retry.
03:52:51	START RequestId: d2915d33-b3e3-11e8-ba0e-7b274ecc95ad Version: $LATEST
03:52:51	2018-09-09T03:52:51.257Z d2915d33-b3e3-11e8-ba0e-7b274ecc95ad REQUEST RECEIVED: {"RequestType":"Create","ServiceToken":"arn:aws:lambda:us-east-1:472821263165:fu
03:52:54	2018-09-09T03:52:54.890Z d2915d33-b3e3-11e8-ba0e-7b274ecc95ad RESPONSE BODY: {"Status":"SUCCESS","Reason":"See the details in CloudWatch Log Stream: 2018/09/09/

```
2018-09-09T03:52:54.890Z d2915d33-b3e3-11e8-ba0e-7b274ecc95ad RESPONSE BODY:
{
    "Status": "SUCCESS",
    "Reason": "See the details in CloudWatch Log Stream: 2018/09/09/[$LATEST]ecca5245eca9415584a74f11c8ab8df1",
    "PhysicalResourceId": "2018/09/09/[$LATEST]ecca5245eca9415584a74f11c8ab8df1",
    "StackId": "arn:aws:cloudformation:us-east-1:472821263165:stack/custom-resource-ami-lookup-demo/c626a690-b3e3-11e8-ba69-500c28637499",
    "RequestId": "851f9733-efe6-48e3-8c8e-51b6e4b6cd4d",
    "LogicalResourceId": "AMIInfo",
    "Data": {
        "Id": "ami-0ff8a91507f77f867"
    }
}
```

AMI Id sent by Lambda in response

Let’s verify that our EC2 Instance was launched using this AMI ID ami-0ff8a91507f77f867



Name	Instance ID	Instance State	Public DNS (IPv4)	IPv4 Public IP	Launch Time
custom-resource-ami-lookup-demo	i-0f9edcdf4f2660d1f	● running	ec2-18-204-219-217.co…	18.204.219.217	September 9, 2018 at 9:23:00 AM UTC+5:30

Instance: i-0f9edcdf4f2660d1f (custom-resource-ami-lookup-demo) Public DNS: ec2-18-204-219-217.compute-1.amazonaws.com

Instance ID	i-0f9edcdf4f2660d1f	Public DNS (IPv4)	ec2-18-204-219-217.compute-1.amazonaws.com
Instance state	running	IPv4 Public IP	18.204.219.217
Instance type	t2.micro	IPv6 IPs	-
Elastic IPs		Private DNS	ip-172-31-93-54.ec2.internal
Availability zone	us-east-1c	Private IPs	172.31.93.54
Security groups	default. view inbound rules. view outbound rules	Secondary private IPs	
Scheduled events	No scheduled events	VPC ID	vpc-2a011251
AMI ID	amzn-ami-hvm-2018.03.0.20180811-x86_64-gp2 (ami-0ff8a91507f77f867)	Subnet ID	subnet-d070a0fe
Platform	-	Network interfaces	eth0
IAM role	-	Source/dest. check	True