

# Outputs

In this lab, you're working with AWS CloudFormation to manage your infrastructure as code. The primary focus is on understanding and utilizing CloudFormation outputs, intrinsic functions, and conditions. Here's a brief summary of each section and its goal:

## 1. Output Section:

- **Goal:** Learn how to declare output values in CloudFormation templates and view these outputs in the CloudFormation console. Outputs can be imported into other stacks for creating cross-stack references.
- **Example:** Viewing the instance ID of an EC2 instance.

## 2. Intrinsic Function Get Attributes:

- **Goal:** Use the `GetAtt` intrinsic function to retrieve attributes like public DNS name and Availability Zone of resources.
- **Example:** Viewing the public DNS name and Availability Zone along with the instance ID.

## 3. Intrinsic Functions Sub and Exports:

- **Goal:** Learn to use the `Fn::Sub` function for string substitutions and the `Fn::Export` function for exporting values to be used in other stacks.
- **Example:** Exporting the security group and using it in another stack.

## 4. Intrinsic Function Import Value:

- **Goal:** Use the `Fn::ImportValue` function to import values from previously exported outputs of other stacks, promoting reusability and consistency.
- **Example:** Creating a new stack that references the security group and availability zone from a previous stack.

## 5. Using Conditions in Output:

- **Goal:** Implement conditions in the output section to dynamically generate outputs based on specific conditions.
- **Example:** Creating a development security group (SG) if the environment is set to "dev."

**End Goal:** The end goal is to understand how to effectively use CloudFormation to automate infrastructure management, ensuring reusable, consistent, and maintainable infrastructure deployments. By the end of the lab, you'll be proficient in defining outputs, using intrinsic functions for dynamic resource properties, and implementing conditions in CloudFormation templates.

## To begin with the Lab

### Instance ID as Output

1. The Output section declares output values that we can import into other stacks (to create cross-stack references). When using Nested stacks, we can see how the outputs of a nested stack are used in a Root stack. We can view outputs on the cloud formation console
2. Now in this lab we will use the Output function in our template to view the instance ID of our instance.

- Below you can see the code used in our template. You can get these templates from our GitHub account.

```

55  Outputs:
56    MyInstanceId:
57      Description: My instance id
58      Value: !Ref MyVMInstance
59

```

- Once you have seen the template, open your console, go to cloud formation, and click Create Stack. Now you need to choose the same options shown below, then just upload your template.

Create stack

**Prerequisite - Prepare template**

Prepare template  
Every stack is based on a template. A template is a JSON or YAML file that contains configuration information about the AWS resources you want to include in the stack.

Choose an existing template  
Upload or choose an existing template.

Use a sample template  
Choose from our sample template library.

Build from Application Composer  
Create a template using a visual builder.

**Specify template Info**  
A template is a JSON or YAML file that describes your stack's resources and properties.

Template source  
Selecting a template generates an Amazon S3 URL where it will be stored.

Amazon S3 URL  
Provide an Amazon S3 URL to your template.

Upload a template file  
Upload your template directly to the console.

Sync from Git - new  
Sync a template from your Git repository.

Upload a template file

- Then give your stack a name choose the environment as dev, also choose your key pair, after that move to the review page, and create your stack.

Specify stack details

**Provide a stack name**

Stack name  
  
Stack name must be 1 to 128 characters, start with a letter, and only contain alphanumeric characters. Character count: 12/128.

**Parameters**  
Parameters are defined in your template and allow you to input custom values when you create or update a stack.

MyEnvironmentName  
Select the environment

MyKeyName  
Select the Key Name

- Now in the events tab you can see that our stack has been created successfully.

Stack-output																																																												
<input type="button" value="Stacks"/> <input type="button" value="Delete"/> <input type="button" value="Update"/> <input type="button" value="Stack actions ▾"/> <input type="button" value="Create stack ▾"/>																																																												
<input type="button" value="Stack info"/> <input type="button" value="Events"/> <input type="button" value="Resources"/> <input type="button" value="Outputs"/> <input type="button" value="Parameters"/> <input type="button" value="Template"/> <input type="button" value="Change sets"/> <input type="button" value="Git sync - new"/>																																																												
<b>Events (10)</b>																																																												
<input type="text" value="Search events"/> <input type="button" value="Detect root cause"/> <input type="button" value="C"/> <input type="button" value="@"/>																																																												
<table border="1"> <thead> <tr> <th>Timestamp</th><th>Logical ID</th><th>Status</th><th>Detailed status</th><th>Status reason</th></tr> </thead> <tbody> <tr> <td>2024-07-20 13:31:42 UTC+0530</td><td>Stack-output</td><td><span>CREATE_COMPLETE</span></td><td>-</td><td>-</td></tr> <tr> <td>2024-07-20 13:31:42 UTC+0530</td><td>MyVMInstance</td><td><span>CREATE_COMPLETE</span></td><td>-</td><td>-</td></tr> <tr> <td>2024-07-20 13:31:12 UTC+0530</td><td>Stack-output</td><td><span>CREATE_IN_PROGRESS</span></td><td><span>CONFIGURATION_COMPLETE</span></td><td>Eventual consistency check initiated</td></tr> <tr> <td>2024-07-20 13:31:12 UTC+0530</td><td>MyVMInstance</td><td><span>CREATE_IN_PROGRESS</span></td><td><span>CONFIGURATION_COMPLETE</span></td><td>Eventual consistency check initiated</td></tr> <tr> <td>2024-07-20 13:31:00 UTC+0530</td><td>MyVMInstance</td><td><span>CREATE_IN_PROGRESS</span></td><td>-</td><td>Resource creation Initiated</td></tr> <tr> <td>2024-07-20 13:30:58 UTC+0530</td><td>MyVMInstance</td><td><span>CREATE_IN_PROGRESS</span></td><td>-</td><td>-</td></tr> <tr> <td>2024-07-20 13:30:58 UTC+0530</td><td>MyDevGlobalSecurityGroup</td><td><span>CREATE_COMPLETE</span></td><td>-</td><td>-</td></tr> <tr> <td>2024-07-20 13:30:58 UTC+0530</td><td>MyDevGlobalSecurityGroup</td><td><span>CREATE_IN_PROGRESS</span></td><td>-</td><td>Resource creation Initiated</td></tr> <tr> <td>2024-07-20 13:30:55 UTC+0530</td><td>MyDevGlobalSecurityGroup</td><td><span>CREATE_IN_PROGRESS</span></td><td>-</td><td>-</td></tr> <tr> <td>2024-07-20 13:30:52 UTC+0530</td><td>Stack-output</td><td><span>CREATE_IN_PROGRESS</span></td><td>-</td><td>User Initiated</td></tr> </tbody> </table>						Timestamp	Logical ID	Status	Detailed status	Status reason	2024-07-20 13:31:42 UTC+0530	Stack-output	<span>CREATE_COMPLETE</span>	-	-	2024-07-20 13:31:42 UTC+0530	MyVMInstance	<span>CREATE_COMPLETE</span>	-	-	2024-07-20 13:31:12 UTC+0530	Stack-output	<span>CREATE_IN_PROGRESS</span>	<span>CONFIGURATION_COMPLETE</span>	Eventual consistency check initiated	2024-07-20 13:31:12 UTC+0530	MyVMInstance	<span>CREATE_IN_PROGRESS</span>	<span>CONFIGURATION_COMPLETE</span>	Eventual consistency check initiated	2024-07-20 13:31:00 UTC+0530	MyVMInstance	<span>CREATE_IN_PROGRESS</span>	-	Resource creation Initiated	2024-07-20 13:30:58 UTC+0530	MyVMInstance	<span>CREATE_IN_PROGRESS</span>	-	-	2024-07-20 13:30:58 UTC+0530	MyDevGlobalSecurityGroup	<span>CREATE_COMPLETE</span>	-	-	2024-07-20 13:30:58 UTC+0530	MyDevGlobalSecurityGroup	<span>CREATE_IN_PROGRESS</span>	-	Resource creation Initiated	2024-07-20 13:30:55 UTC+0530	MyDevGlobalSecurityGroup	<span>CREATE_IN_PROGRESS</span>	-	-	2024-07-20 13:30:52 UTC+0530	Stack-output	<span>CREATE_IN_PROGRESS</span>	-	User Initiated
Timestamp	Logical ID	Status	Detailed status	Status reason																																																								
2024-07-20 13:31:42 UTC+0530	Stack-output	<span>CREATE_COMPLETE</span>	-	-																																																								
2024-07-20 13:31:42 UTC+0530	MyVMInstance	<span>CREATE_COMPLETE</span>	-	-																																																								
2024-07-20 13:31:12 UTC+0530	Stack-output	<span>CREATE_IN_PROGRESS</span>	<span>CONFIGURATION_COMPLETE</span>	Eventual consistency check initiated																																																								
2024-07-20 13:31:12 UTC+0530	MyVMInstance	<span>CREATE_IN_PROGRESS</span>	<span>CONFIGURATION_COMPLETE</span>	Eventual consistency check initiated																																																								
2024-07-20 13:31:00 UTC+0530	MyVMInstance	<span>CREATE_IN_PROGRESS</span>	-	Resource creation Initiated																																																								
2024-07-20 13:30:58 UTC+0530	MyVMInstance	<span>CREATE_IN_PROGRESS</span>	-	-																																																								
2024-07-20 13:30:58 UTC+0530	MyDevGlobalSecurityGroup	<span>CREATE_COMPLETE</span>	-	-																																																								
2024-07-20 13:30:58 UTC+0530	MyDevGlobalSecurityGroup	<span>CREATE_IN_PROGRESS</span>	-	Resource creation Initiated																																																								
2024-07-20 13:30:55 UTC+0530	MyDevGlobalSecurityGroup	<span>CREATE_IN_PROGRESS</span>	-	-																																																								
2024-07-20 13:30:52 UTC+0530	Stack-output	<span>CREATE_IN_PROGRESS</span>	-	User Initiated																																																								

7. Also, if we go to the outputs section we can see that we got the key and the value.

CloudFormation > Stacks > Stack-output													
Stack-output													
<input type="button" value="Stack info"/> <input type="button" value="Events"/> <input type="button" value="Resources"/> <input type="button" value="Outputs"/> <input type="button" value="Parameters"/> <input type="button" value="Template"/> <input type="button" value="Change sets"/> <input type="button" value="Git sync - new"/>													
<b>Outputs (1)</b>													
<input type="text" value="Search outputs"/> <input type="button" value="C"/> <input type="button" value="@"/>													
<table border="1"> <thead> <tr> <th>Key</th><th>Value</th><th>Description</th><th>Export name</th></tr> </thead> <tbody> <tr> <td>MyInstanceld</td><td>i-0092e8eff9dec3852</td><td>My instance id</td><td>-</td></tr> </tbody> </table>						Key	Value	Description	Export name	MyInstanceld	i-0092e8eff9dec3852	My instance id	-
Key	Value	Description	Export name										
MyInstanceld	i-0092e8eff9dec3852	My instance id	-										

8. Now if we move to EC2 and match the instance ID we can see that they both are same.

Instances (1/1) Info																	
<input type="text" value="Find Instance by attribute or tag (case-sensitive)"/> <input type="button" value="All states ▾"/>																	
<input type="button" value="C"/> <input type="button" value="Connect"/> <input type="button" value="Instance state ▾"/> <input type="button" value="Actions ▾"/> <input type="button" value="Launch instances"/>																	
<input type="checkbox"/> <input type="button" value="Name"/> <input type="checkbox"/> <input type="button" value="Instance ID"/> <input type="checkbox"/> <input type="button" value="Instance state"/> <input type="checkbox"/> <input type="button" value="Instance type"/> <input type="checkbox"/> <input type="button" value="Status check"/> <input type="checkbox"/> <input type="button" value="Alarm status"/> <input type="checkbox"/> <input type="button" value="Availability Zone"/> <input type="checkbox"/> <input type="button" value="Pu"/>																	
<input checked="" type="checkbox"/> <input type="button" value="i-0092e8eff9dec3852"/> <input checked="" type="checkbox"/> <input type="button" value="Running"/> <input checked="" type="checkbox"/> <input type="button" value="t2.micro"/> <input checked="" type="checkbox"/> <input type="button" value="Initializing"/> <input checked="" type="checkbox"/> <input type="button" value="View alarms +"/> <input checked="" type="checkbox"/> <input type="button" value="ap-southeast-1a"/> <input checked="" type="checkbox"/> <input type="button" value="ec"/>																	
<b>i-0092e8eff9dec3852</b>																	
<input type="button" value="Details"/> <input type="button" value="Status and alarms"/> <input type="button" value="Monitoring"/> <input type="button" value="Security"/> <input type="button" value="Networking"/> <input type="button" value="Storage"/> <input type="button" value="Tags"/>																	
<b>Instance summary</b> <input type="button" value="Info"/>																	
<table border="1"> <tr> <td>Instance ID</td><td><b>i-0092e8eff9dec3852</b></td><td>Public IPv4 address</td><td>54.169.181.7   <a href="#">open address</a></td><td>Private IPv4 addresses</td><td>172.31.22.193</td></tr> <tr> <td>IPv6 address</td><td>-</td><td>Instance state</td><td><span>Running</span></td><td>Public IPv4 DNS</td><td>ec2-54-169-181-7.ap-southeast-1.compute.amazonaws.com   <a href="#">open address</a></td></tr> </table>						Instance ID	<b>i-0092e8eff9dec3852</b>	Public IPv4 address	54.169.181.7   <a href="#">open address</a>	Private IPv4 addresses	172.31.22.193	IPv6 address	-	Instance state	<span>Running</span>	Public IPv4 DNS	ec2-54-169-181-7.ap-southeast-1.compute.amazonaws.com   <a href="#">open address</a>
Instance ID	<b>i-0092e8eff9dec3852</b>	Public IPv4 address	54.169.181.7   <a href="#">open address</a>	Private IPv4 addresses	172.31.22.193												
IPv6 address	-	Instance state	<span>Running</span>	Public IPv4 DNS	ec2-54-169-181-7.ap-southeast-1.compute.amazonaws.com   <a href="#">open address</a>												

## 😊 Intrinsic function Get Attributes

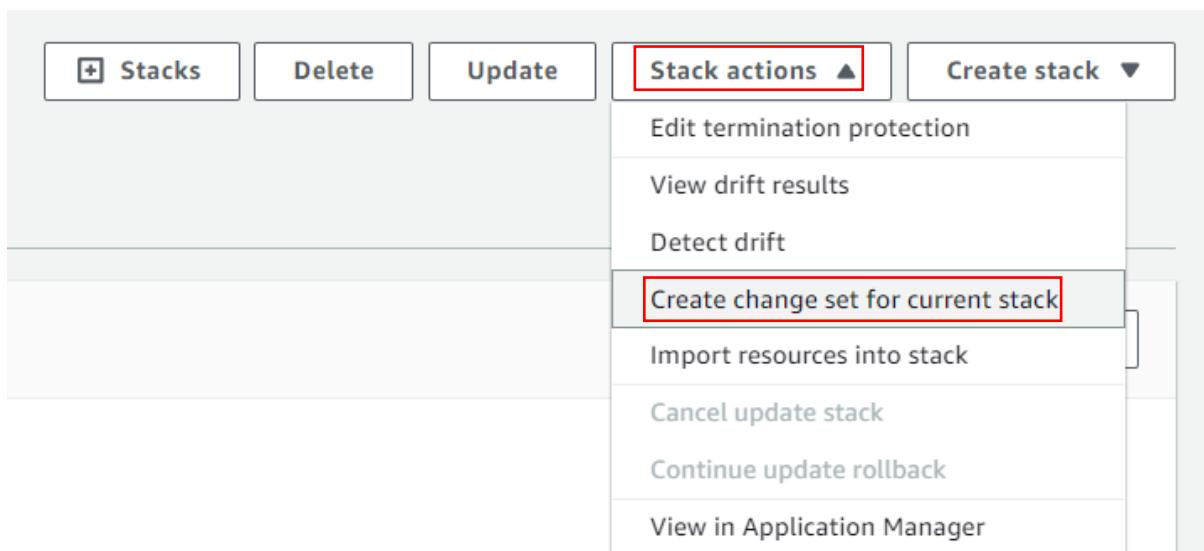
- Now we are going to use the Get Attributes function and we will be able to see the public DNS name and the Availability zone along with the instance ID.
- You can see the code which we are going to use in our template.

```

55 ~ Outputs:
56 ~   MyInstanceId:
57     Description: My instance id
58     Value: !Ref MyVMInstance
59 ~   MyDNSName:
60     Description: My Public DNS Name
61     Value: !GetAtt MyVMInstance.PublicDnsName
62 ~   MyInstanceAvailabilityZone:
63     Description: Instance Availability Zone
64     Value: !GetAtt MyVMInstance.AvailabilityZone
65
66

```

- Now move to cloud formation and create a change set for the current stack. For that in your current stack expand stack actions and there you can find the option click on it.



- Now you need to choose the same option as shown below and upload your template.
- Then just move to the review page and click on submit.

## Create change set for Stack-output

### Prerequisite - Prepare template

#### Prepare template

Every stack is based on a template. A template is a JSON or YAML file that contains configuration information about the AWS resources you want to include in the stack.

Use existing template

Proceed with the template you are already using for this stack.

Replace existing template

Replace your existing template with a new template.

Edit in Application Composer

Edit your template in a visual builder.

### Specify template

A template is a JSON or YAML file that describes your stack's resources and properties.

#### Template source

Selecting a template generates an Amazon S3 URL where it will be stored.

Amazon S3 URL

Upload a template file

#### Upload a template file

Choose file

- Now you need to click on Execute change set. Also, you cannot see any changes because we are just adding the output not changing our resources.

Stack-outp-juxfiz94g1r-31rvmqq012k

Delete change set  Execute change set

Overview	
Change set ID arn:aws:cloudformation:ap-southeast-1:878893308172:changeSet/Stack-outp-juxfiz94g1r-31rvmqq012k/2c1f4054-4523-4d16-b6c2-caeeacf0641	Status <span style="color: green;">CREATE_COMPLETE</span>
Description Outputs practice	Status reason -
Created time 2024-07-20 13:41:08 UTC+0530	Execution status <span style="color: green;">AVAILABLE</span>

[Changes](#) [Input](#) [Template](#) [JSON changes](#) [Hook invocations](#)

**Changes (0)**  
Preview how proposed changes to a stack will impact running resources. Click on "View details" to preview the impact on property values for a resource.

Search changes							
Action	Logical ID	Resource type	Replacement	Property-level changes	Policy action	Physical ID	Module
No changes There are no resources to be changed							

- From the events tab you can see that our update has been completed successfully.

Stack-output

Stacks  Delete  Update  Stack actions  Create stack

[Stack info](#) [Events](#) [Resources](#) [Outputs](#) [Parameters](#) [Template](#) [Change sets](#) [Git sync - new](#)

**Events (13)**

Search events					
Timestamp	Logical ID	Status	Detailed status	Status reason	
2024-07-20 13:48:46 UTC+0530	Stack-output	<span style="color: green;">UPDATE_COMPLETE</span>	-	-	
2024-07-20 13:48:46 UTC+0530	Stack-output	<span style="color: blue;">UPDATE_COMPLETE_CLEANUP_IN_PROGRESS</span>	-	-	
2024-07-20 13:48:40 UTC+0530	Stack-output	<span style="color: blue;">UPDATE_IN_PROGRESS</span>	-	User Initiated	

- And if you go to the output section you will see the desired output that we have expected.

The screenshot shows the AWS CloudFormation Outputs page for a stack named 'Stack-output'. The page has tabs for Stack info, Events, Resources, Outputs (which is selected), Parameters, Template, Change sets, and Git sync - new. The Outputs section displays three entries:

Key	Value	Description	Export name
MyDNSName	<a href="#">ec2-54-169-181-7.ap-southeast-1.compute.amazonaws.com</a>	My Public DNS Name	-
MyInstanceAvailabilityZone	ap-southeast-1a	Instance Availability Zone	-
MyInstanceId	i-0092e8eff9dec3852	My instance id	-

## 😊 Intrinsic function Sub and Exports

- Now we are going to look at the substitute and export functions. Below you can see the code which we are going to use in our template.

```

55  Outputs:
56    MyInstanceId:
57      Description: My instance id
58      Value: !Ref MyVMInstance
59    MyDNSName:
60      Description: My Public DNS Name
61      Value: !GetAtt MyVMInstance.PublicDnsName
62    MyInstanceAvailabilityZone:
63      Description: Instance Availability Zone
64      Value: !GetAtt MyVMInstance.AvailabilityZone
65    Export:
66      Name: !Sub "${AWS::StackName}-InstanceAz"
67    MyDevGlobalSecurityGroup:
68      Description: My Dev SG
69      Value: !Ref MyDevGlobalSecurityGroup
70      Export:
71        Name: MyDevSSHGlobalSG
72
73

```

- Now we are going to cloud formation and this time we are going to update the stack.

The screenshot shows the AWS CloudFormation Stack actions menu. The menu items are: Delete, Update (which is highlighted with a red box), Stack actions ▾, and Create stack ▾.

- Then you need to choose the same options as shown below and upload your template.

**Update stack**

**Prerequisite - Prepare template**

**Prepare template**  
Every stack is based on a template. A template is a JSON or YAML file that contains configuration information about the AWS resources you want to include in the stack.

Use existing template  
Proceed with the template you are already using for this stack.

Replace existing template  
Replace your existing template with a new template.

Edit in Application Composer  
Edit your template in a visual builder.

**Specify template**  
A template is a JSON or YAML file that describes your stack's resources and properties.

**Template source**  
Selecting a template generates an Amazon S3 URL where it will be stored.

Amazon S3 URL

Upload a template file

Upload a template file

- After that just move to the review page and update your stack. And below you can see that our stack has been updated successfully.

**Stack-output**

**Events** (16)

Timestamp	Logical ID	Status	Detailed status	Status reason
2024-07-20 16:02:08 UTC+0530	Stack-output	UPDATE_COMPLETE	-	-
2024-07-20 16:02:07 UTC+0530	Stack-output	UPDATE_COMPLETE_CLEANUP_IN_PROGRESS	-	-
2024-07-20 16:02:01 UTC+0530	Stack-output	UPDATE_IN_PROGRESS	-	User Initiated

- Then in the output section you can see that we have our desired output and we can see the security group.

**CloudFormation > Stacks > Stack-output**

**Stack-output**

**Outputs** (4)

Key	Value	Description	Export name
MyDevGlobalSecurityGroup	Stack-output-MyDevGlobalSecurityGroup-cfmAVKyBINM	My Dev SG	MyDevSSHGlobalSG
MyDNSName	<a href="https://ec2-54-169-181-7.ap-southeast-1.compute.amazonaws.com">ec2-54-169-181-7.ap-southeast-1.compute.amazonaws.com</a>	My Public DNS Name	-
MyInstanceAvailabilityZone	ap-southeast-1a	Instance Availability Zone	Stack-output-InstanceAz
MyInstanceld	i-0092e8eff9dec3852	My instance id	-

## 💡 Intrinsic function Import Value

- In this lab we will create a new stack by referencing the Security group and availability zone value from our previous stack. We will use Fn::ImportValue intrinsic function to import those exports.

- Now go to cloud formation and create a new template, below you can see that our previous stack is also present because we will use this stack as reference to our new stack.

The screenshot shows the AWS CloudFormation Stacks page. At the top, there's a navigation bar with 'CloudFormation > Stacks'. Below it, a table lists one stack: 'Stack-output' with status 'UPDATE\_COMPLETE', created on '2024-07-20 13:30:52 UTC+0530', and a description 'Outputs practice'. There are buttons for 'Delete', 'Update', 'Stack actions', and 'Create stack'.

- Now choose the same values as shown below, upload your template and move to the next page.

The screenshot shows the 'Create stack' wizard. On the first step, 'Prerequisite - Prepare template', there are three options: 'Choose an existing template' (selected), 'Use a sample template', and 'Build from Application Composer'.

- Then you need to give your stack a name and choose your key pair then choose the environment as dev.
- After that just move to the review page and create your stack.

The screenshot shows the 'Specify stack details' wizard. It has two main sections: 'Provide a stack name' where the stack name is 'Stack-output-2', and 'Parameters' where 'MyEnvironmentName' is set to 'dev' and 'MyKeyName' is set to 'CFKeyPair'.

- Below you can see that our stack has been created.

CloudFormation > Stacks > Stack-output-2					
Stack-output-2					
<a href="#">Stack info</a>   <a href="#">Events</a>   <a href="#">Resources</a>   <a href="#">Outputs</a>   <a href="#">Parameters</a>   <a href="#">Template</a>   <a href="#">Change sets</a>   <a href="#">Git sync - new</a>					
<b>Events (7)</b>					
<input type="text" value="Search events"/> <span style="float: right;">Detect root cause</span>					
Timestamp	Logical ID	Status	Detailed status	Status reason	
2024-07-20 16:22:05 UTC+0530	Stack-output-2	<span>CREATE_COMPLETE</span>	-	-	
2024-07-20 16:22:04 UTC+0530	MyVMinstance	<span>CREATE_COMPLETE</span>	-	-	
2024-07-20 16:21:55 UTC+0530	Stack-output-2	<span>CREATE_IN_PROGRESS</span>	<span>CONFIGURATION_COMPLETE</span>	Eventual consistency check initiated	
2024-07-20 16:21:55 UTC+0530	MyVMinstance	<span>CREATE_IN_PROGRESS</span>	<span>CONFIGURATION_COMPLETE</span>	Eventual consistency check initiated	
2024-07-20 16:21:43 UTC+0530	MyVMinstance	<span>CREATE_IN_PROGRESS</span>	-	Resource creation Initiated	
2024-07-20 16:21:42 UTC+0530	MyVMinstance	<span>CREATE_IN_PROGRESS</span>	-	-	
2024-07-20 16:21:39 UTC+0530	Stack-output-2	<span>CREATE_IN_PROGRESS</span>	-	User Initiated	

7. So, in the resources you can see it will have only my VM instance because the other security group instance, whatever we have used is. Just a reference. So, which means we have defined a security in resource and then that becomes a reference only for stack two. So, it will not be created as part of this respective stack. It's just only a reference.

CloudFormation > Stacks > Stack-output-2					
Stack-output-2					
<a href="#">Stack info</a>   <a href="#">Events</a>   <a href="#">Resources</a>   <a href="#">Outputs</a>   <a href="#">Parameters</a>   <a href="#">Template</a>   <a href="#">Change sets</a>   <a href="#">Git sync - new</a>					
<b>Resources (1)</b>					
<input type="text" value="Search resources"/> <span style="float: right;">C</span>					
Logical ID	Physical ID	Type	Status	Module	
MyVMinstance	i-0877b1d9ecaaa09cc	AWS::EC2::Instance	<span>CREATE_COMPLETE</span>	-	

8. Stated differently, this allows us to establish the standards inside a global stack that we can develop in a real-world scenario. After that, export those standards—for example, export the values—or anything else we require. For instance, we can build a global security group, export it from one stack, and then utilize the appropriate export value of that global security group in each stack that pertains to that company, making any necessary changes there. Thus, all corresponding stacks will receive those modifications if we update it concerning the global security group.
9. So, now if you make any changes to stack 1 then the changes will also take effect in stack 2 which we just created. Also, you can go to EC2 to check the instance.
10. Once you are done then delete both of your stacks.

## 💡 Using Conditions in Output

1. This time we will use conditions in the output section, and we will say that if we are using dev as our environment then create a dev SG.
2. Below is the code which we are going to use in our template.

```

59 ~ Outputs:
60 ~   MyInstanceId:
61     Description: My instance id
62     Value: !Ref MyVMInstance
63 ~   MyDNSName:
64     Description: My Public DNS Name
65     Value: !GetAtt MyVMInstance.PublicDnsName
66 ~   MyInstanceAvailabilityZone:
67     Description: Instance Availability Zone
68     Value: !GetAtt MyVMInstance.AvailabilityZone
69 ~   Export:
70     Name: !Sub "${AWS::StackName}-InstanceAz"
71 ~   MyDevGlobalSecurityGroup:
72     Description: My Dev SG
73     Value: !Ref MyDevGlobalSecurityGroup
74     Condition: CreateDevSecurityGroup
75 ~   Export:
76     Name: MyDevSSHGlobalSG
77
78

```

3. Now move to cloud formation and create your stack. Then upload your template and move to the next page, here you need to give your stack a name and then choose your environment as dev then choose your key pair.
4. After that just move to the review page and create your stack

Specify stack details

**Provide a stack name**

Stack name  
Stack-condition-output

Stack name must be 1 to 128 characters, start with a letter, and only contain alphanumeric characters. Character count: 22/128.

**Parameters**

Parameters are defined in your template and allow you to input custom values when you create or update a stack.

MyEnvironmentName Select the environment	dev
MyKeyName Select the Key Name	CFKeyPair

Cancel Previous Next

5. Below you can see that our stack got created successfully.

Stack-condition-output					
Stack info		Events	Resources	Outputs	Parameters
Events (10)					
<input type="text"/> Search events <span style="float: right;">Detect root cause</span> <span style="float: right;">C</span> <span style="float: right;">@</span>					
Timestamp	Logical ID	Status	Detailed status	Status reason	
2024-07-20 16:40:24 UTC+0530	Stack-condition-output	CREATE_COMPLETE	-	-	
2024-07-20 16:40:21 UTC+0530	MyVMinstance	CREATE_COMPLETE	-	-	
2024-07-20 16:40:01 UTC+0530	Stack-condition-output	CREATE_IN_PROGRESS	CONFIGURATION_COMPLETE	Eventual consistency check initiated	
2024-07-20 16:40:01 UTC+0530	MyVMinstance	CREATE_IN_PROGRESS	CONFIGURATION_COMPLETE	Eventual consistency check initiated	
2024-07-20 16:39:50 UTC+0530	MyVMinstance	CREATE_IN_PROGRESS	-	Resource creation Initiated	
2024-07-20 16:39:48 UTC+0530	MyVMinstance	CREATE_IN_PROGRESS	-	-	
2024-07-20 16:39:47 UTC+0530	MyDevGlobalSecurityGroup	CREATE_COMPLETE	-	-	
2024-07-20 16:39:47 UTC+0530	MyDevGlobalSecurityGroup	CREATE_IN_PROGRESS	-	Resource creation Initiated	
2024-07-20 16:39:44 UTC+0530	MyDevGlobalSecurityGroup	CREATE_IN_PROGRESS	-	-	
2024-07-20 16:39:42 UTC+0530	Stack-condition-output	CREATE_IN_PROGRESS	-	User Initiated	

6. From the output section you can see that the SG is dev SG which we'd expected. You can go to EC2 to verify it.

CloudFormation > Stacks > Stack-condition-output					
Stack-condition-output					
Stack info		Events	Resources	Outputs	Parameters
Outputs (4)					
<input type="text"/> Search outputs <span style="float: right;">C</span> <span style="float: right;">&lt;</span> <span style="float: right;">1</span> <span style="float: right;">&gt;</span> <span style="float: right;">@</span>					
Key	Value	Description	Export name		
MyDevGlobalSecurityGroup	Stack-condition-output-MyDevGlobalSecurityGroup-vSLVEWCyF0nm	My Dev SG	MyDevSSHGlobalSG		
MyDNSName	<a href="#">ec2-13-229-197-228.ap-southeast-1.compute.amazonaws.com</a>	My Public DNS Name	-		
MyInstanceAvailabilityZone	ap-southeast-1a	Instance Availability Zone	Stack-condition-output-InstanceAz		
MyInstanceId	i-05d49db5f64dc46fc	My instance id	-		