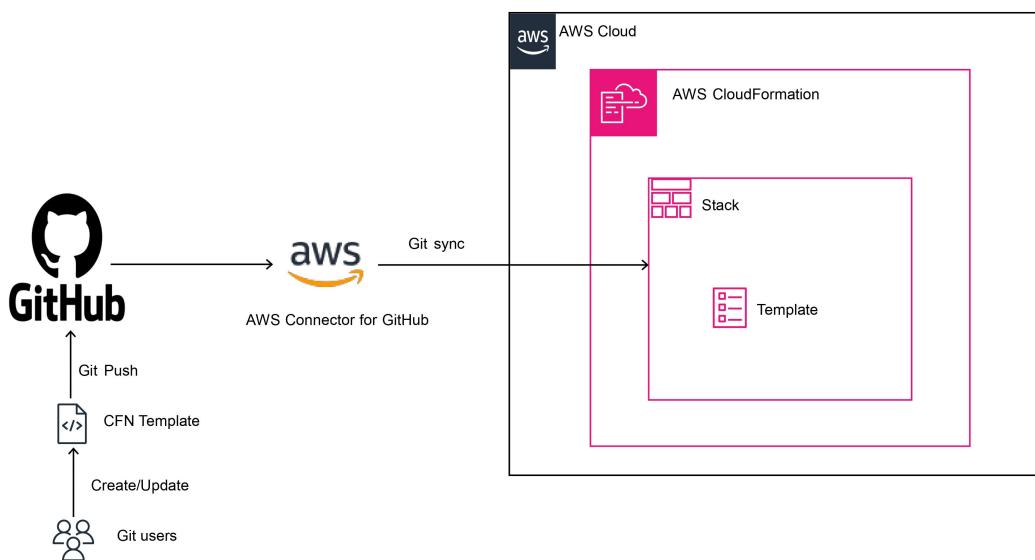


Automating Infrastructure Deployment with Gitsync and AWS CloudFormation



By
Mahendran Selvakumar
<https://devopstronaut.com/>



Create a Connection Between AWS and GitHub:

Please follow this link to establish the connection: Step-by-Step Guide to Connecting AWS with GitHub for Automated Workflow

1.Create a Repository on GitHub

The screenshot shows a GitHub repository page for 'cfgitsync'. At the top, there's a navigation bar with links for Code, Issues, Pull requests, Actions, Projects, Wiki, Security, Insights, and Settings. Below the navigation bar, the repository name 'cfgitsync' is displayed along with a 'Public' badge. A commit history is shown, with the most recent commit by 'Mahendran Selvakumar' titled 'Update ec2instance.yml' and a timestamp of '5 minutes ago'. Below the commit history, there's a file list with 'README.md' and 'ec2instance.yml'. The 'ec2instance.yml' file is selected.

2.Create an EC2 Instance with a CloudFormation Template and Upload It to Your Repository

The screenshot shows the same GitHub repository 'cfgitsync'. On the left, there's a sidebar with a 'Files' section containing 'README.md' and 'ec2instance.yml'. The main area shows the file 'ec2instance.yml' with its content displayed. The content of 'ec2instance.yml' is as follows:

```
AWSTemplateFormatVersion: '2010-09-09'
Description: Basic CloudFormation Template to create an EC2 Instance
Resources:
  CFInstance:
    Type: AWS::EC2::Instance
    Properties:
      InstanceType: t3.micro
      KeyName: cfinstance
      ImageId: ami-02af70169146bbdd3
      SubnetId: subnet-0f4e64ce0a813bf21
      SecurityGroupIds:
        - sg-086bbc46373ed734e
```

<https://www.linkedin.com/in/mahendran-selvakumar-36444a77/>



3. Create an IAM Role for CloudFormation: I've created it with administrator access, but restrict access when configuring the production environment

The screenshot shows the AWS IAM Roles page. The 'Cfaccess' role is selected. The 'Summary' tab is active, displaying creation date (August 08, 2024), last activity (8 hours ago), ARN (arn:aws:iam::851725583489:role/Cfaccess), and maximum session duration (1 hour). Below the summary, there are tabs for 'Permissions', 'Trust relationships', 'Tags', 'Access Advisor', and 'Revoke sessions'. The 'Permissions' tab is selected, showing one managed policy attached: 'AdministratorAccess' (AWS managed - job function). There are buttons for 'Edit', 'Delete', 'Simulate', 'Remove', and 'Add permissions'.

4. Go to AWS and search for CloudFormation

The screenshot shows the AWS CloudFormation search results page. The search bar contains 'cloudformation'. The results section is titled 'Search results for 'cloudformation'' and includes two main categories: 'Services' and 'Features'. Under 'Services', there is a card for 'CloudFormation' with the subtext 'Create and Manage Resources with Templates'. Under 'Features', there is a card for 'IaC Generator' with the subtext 'CloudFormation feature'. On the left sidebar, there are links for 'Welcome', 'Documentation', 'Marketplace', 'Blogs', 'Events', and 'Tutorials'.

5. click "Create stack"

The screenshot shows the AWS CloudFormation landing page. The main heading is 'AWS CloudFormation Model and provision all your cloud infrastructure'. It includes a subtext: 'AWS CloudFormation provides a common language to describe and provision all the infrastructure resources in your environment in a safe, repeatable way.' Below this is a 'How it works' section with a diagram. To the right, there is a 'Create a CloudFormation stack' section with a 'Create stack' button, and a 'Getting started' section with links to 'What is AWS CloudFormation?', 'Getting started with CloudFormation', 'Learn template basics', and 'Quick starts'.

<https://www.linkedin.com/in/mahendran-selvakumar-36444a77/>



6. Select 'Choose an existing template' and set the template source to 'Sync from Git,' then click 'Next'

CloudFormation > Stacks > Create stack

Step 1
Create stack

Step 2
Specify stack details

Step 3
Configure stack options

Step 4
Review and create

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.

Cancel **Next**

7. Provide the stack name and select 'Create the file using the following parameters and place it in my repository' for deployment file creation

CloudFormation > Stacks > Create stack

Step 1
Create stack

Step 2
Specify stack details

Step 3
Configure stack options

Step 4
Review and create

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: 10/128.

Stack deployment file Info

Your deployment file allows you to input custom values which will be used to update your stack.

Deployment file creation
You can create your own deployment file or CloudFormation can create it for you.

Create the file using the following parameters and place it in my repository.

I am providing my own file in my repository.

Template definition repository Info

Choose a Git repository that contains a stack template to sync with CloudFormation.

8. Since we already have a connection between AWS and the Git repository, select 'Choose a linked Git repository,' then choose the created repository and branch. Provide the deployment file path, select 'New IAM role,' and specify the role name



Template definition repository Info

Choose a Git repository that contains a stack template to sync with CloudFormation.

Choose a linked Git repository

Choose a Git repository that you have linked with CloudFormation.

Link a Git repository

Choose a Git repository to link with CloudFormation.

Repository

Choose the linked repository that contains your template.



Branch

CloudFormation will sync changes to your template from this branch.



Deployment file path

Provide a path for a file that will store deployment parameters for your template.

IAM role

This IAM role provides permissions for CloudFormation to update the stack from the Git repository. See [IAM role prerequisites here](#).

New IAM role

Create a role in your account.

Existing IAM role

Choose an existing role in your account.

Role name

Use only letters, numbers or hyphens. The maximum length is 100 characters.

9. Provide the CloudFormation (CFN) template for the stack and click 'Next'

Deployment file parameters

Add parameters from your deployment file to update your stack.

Template file path

Provide the path in your repository for the CloudFormation template for your stack.

Parameters

You can add 50 more parameters.

Tags - optional

Tags (key-value pairs) are used to apply metadata to AWS resources, which can help in organising, identifying and categorising those resources. You can add up to 50 unique tags for each stack.

No tags associated with the stack.

You can add 50 more tag(s)

10. Select the IAM role created for CloudFormation to use

CloudFormation > Stacks > Create stack

Step 1

Step 2

Step 3

Step 4

Configure stack options

Permissions - optional

Specify an existing AWS Identity and Access Management (IAM) service role that CloudFormation can assume.

IAM role

Choose the IAM role for CloudFormation to use for all operations performed on the stack.



⚠ AWS CloudFormation will use this role for Cfaccess permission to pass it. Ensure that this role grants the least privilege.

<https://www.linkedin.com/in/mahendran-selvakumar-36444a77/>



11.Click “Next”

Advanced options
You can set additional options for your stack, like notification options and a stack policy. [Learn more](#)

► **Stack policy - optional**
Defines the resources that you want to protect from unintentional updates during a stack update.

► **Rollback configuration - optional**
Specify alarms for CloudFormation to monitor when creating and updating the stack. If the operation breaches an alarm threshold, CloudFormation rolls it back.

► **Notification options - optional**
Specify a new or existing Amazon Simple Notification Service topic where notifications about stack events are sent.

► **Stack creation options - optional**
Specify the timeout and termination protection options for stack creation.

Cancel Previous Next

12.Click ‘submit’ to create the stack

Stack creation options

Timeout
-

Termination protection
Deactivated

⚠ To provision resources with your Git provider, AWS uses CodeStar Connections APIs, which are not in scope for compliance programmes. For more information, see [AWS Services in Scope by Compliance Program](#).

Git sync connects your stack to your Git repository by submitting a pull request. Once you create this configuration, merge the pull request into your Git repository.

Cancel Previous Submit

13. Once the stack is created, a banner will appear indicating that pull requests have been created in your Git sync repository

Pull requests have been created in your Git sync repository. Merge them to create the stack CFInstance with Git sync support.

[CloudFormation](#) > [Stacks](#) > CFInstance

Stacks (1)

Filter status Active View nested

Stacks

CFInstance
2024-08-10 00:03:30 UTC+0100
CREATE_IN_PROGRESS

CFInstance

Delete Update Stack actions Create stack

Outputs Parameters Template Changesets Git sync - new

⚠ To provision resources with your Git provider, AWS uses CodeStar Connections APIs, which are not in scope for compliance programmes. For more information, see [AWS Services in Scope by Compliance Program](#).

Git sync - new Info Edit Retry latest commit Disconnect

Configure and sync updates for your stack from the linked Git repository.

Repository cfgitsync	Deployment file path CFinstance-deploy.yml
Git sync Enabled	Repository provider GitHub

14. Go to the GitHub repository, and you will see the open pull requests which created by CloudFormation stack

<https://www.linkedin.com/in/mahendran-selvakumar-36444a77/>



devopstronaut / cfgitsync

Type ⌘ to search

Code Issues Pull requests 1 Actions Projects Wiki Security Insights Settings

Label issues and pull requests for new contributors

Now, GitHub will help potential first-time contributors [discover issues](#) labeled with [good first issue](#)

Filters is:pr is:open Labels 9 Milestones 0 New pull request

1 Open 9 Closed Author Label Projects Milestones Reviews Assignee Sort

Add AWS Cloudformation Deployment file for CFInstance stack #10 opened 1 minute ago by aws-connector-for-github bot

ProTip! Exclude your own issues with [-author:devopstronaut](#).

15.Click “Merge Pull request” to merge from aws created branch (ex:aws-sync-main-7ad76278)

devopstronaut / cfgitsync

Pull requests 1 Actions Projects Wiki Security Insights Settings

Add AWS Cloudformation Deployment file for CFInstance stack #10

Open aws-connector-... wants to merge 1 commit into main from aws-sync-main-7ad76278

Conversation 0 Commits 1 Checks 0 Files changed 1

aws-connector-for-github bot commented 3 minutes ago

This pull request commits a CloudFormation [stack deployment file](#) to your repository. AWS CloudFormation uses the `CFinstance-deploy.yml` file to locate, track, and automatically update a stack that belongs to the `CFInstance` template. When this pull request is merged, AWS CloudFormation tracks changes to this repository and applies updates to the stack and parameters that are defined in the file.

When this change is merged, AWS CloudFormation tracks and syncs changes from this repository to the stack defined in `CFinstance-deploy.yml`. Make sure to review it. You can disable syncing at any time in the AWS CloudFormation console.

How does it work?

Your `CFinstance-deploy.yml` file contains the path from the root of your Git repository to your CloudFormation template as well as the parameters and tags for your stack. When changes are committed to your repository, CloudFormation automatically updates the stack, its parameters, and tags.

How do I update the `CFinstance-deploy.yml` file?

Edit your `CFinstance-deploy.yml` file and commit the changes to your repository. Make sure that you commit the changes to the path and branch in the repository that are specified in the deployment file.

Add AWS Cloudformation Deployment file for CFInstance stack e90cb0e

This branch has no conflicts with the base branch

Merging can be performed automatically.

Merge pull request You can also [open this in GitHub Desktop](#) or view [command line instructions](#).

16.Click “Confirm Merge”



Merge pull request #10 from devopstronaut/aws-sync-main-7ad76278

Add AWS Cloudformation Deployment file for CFInstance stack

This commit will be authored by devopstronaut20@gmail.com

Confirm merge **Cancel**

Add a comment

17. Once merged, Git sync will start the provisioning process

The screenshot shows the AWS CloudFormation console with the 'CFInstance' stack selected. The 'Git sync - new' tab is active, displaying the configuration for syncing the stack with a GitHub repository named 'cfgitsync'. The provisioning status is shown as 'In progress'.

18. Git sync has successfully provisioned the resources based on the changeset

The screenshot shows the AWS CloudFormation console with the 'CFInstance' stack selected. The 'Git sync - new' tab is active, displaying the configuration for syncing the stack with a GitHub repository named 'cfgitsync'. The provisioning status is shown as 'Succeeded'. Below, the 'Latest sync events' section lists three events:

Date	Commit ID	Event	Event type
2024-08-10 00:09:27 UTC+0100	88928c40	Changeset execution succeeded and stack updated successfully.	STACK_UPDATE_SUCCEEDED
2024-08-10 00:09:08 UTC+0100	88928c40	Waiting for changeset execution to finish.	CHANGESET_EXECUTION_IN_PROGRESS
2024-08-10 00:09:07 UTC+0100	88928c40	Changeset creation succeeded.	CHANGESET_CREATION_SUCCEEDED

<https://www.linkedin.com/in/mahendran-selvakumar-36444a77/>



19. Go to the EC2 console and verify that the instance has been created by the CloudFormation stack

The screenshot shows the AWS EC2 Instances page. The instance summary for 'i-09d47bc3b1fb54e47' is displayed. Key details include:

- Public IPv4 address: 16.16.120.184 [open address]
- Private IP4 addresses: 172.31.11.74
- Public IPv4 DNS: ec2-16-16-120-184.eu-north-1.compute.amazonaws.com [open address]
- Instance state: Running
- Instance type: t3.micro
- VPC ID: vpc-02aa3080a2b7a0b77
- Subnet ID: subnet-0f4e64ce0a813bf21
- Instance ARN: arn:aws:ec2:eu-north-1:851725583489:instance/i-09d47bc3b1fb54e47

The 'Tags' tab is selected, showing three tags:

Key	Value
aws:cloudformation:logical-id	CFinstance
aws:cloudformation:stack-name	CFinstance
aws:cloudformation:stack-id	arn:aws:cloudformation:eu-north-1:851725583489:stack/CFinstance/98219390-56a3-11ef-97de-06cce2b53075

20. An Amazon EventBridge rule has been created to monitor CloudFormation stack status changes

The screenshot shows the AWS EventBridge Rules page. The rule 'ManagedRuleForCfnGitSync' is listed. Key details include:

Rule name	Status	Event bus name	Type
ManagedRuleForCfnGitSync	Enabled	default	Managed
Description	Rule ARN	Event bus ARN	Managed by
Allows opted in CloudFormation stacks to be synchronized from Git.	arn:aws:events:eu-north-1:851725583489:rule/ManagedRuleForCfnGitSync	arn:aws:events:eu-north-1:851725583489:event-bus/default	cloudformation.sync.codeconnections.amazonaws.com

The 'Event pattern' tab is selected, showing the JSON pattern:

```
1 {  
2   "source": ["aws.cloudformation"],  
3   "detail-type": ["CloudFormation Stack Status Change"]  
4 }
```

A 'Copy' button is present below the event pattern.

21. A deployment file has been created in the GitHub repository



devopstronaut / cfgitsync

Code Issues Pull requests Actions Projects Wiki Security Insights Settings

Files

main Go to file ↗

CFinstance-deploy.yml README.md ec2instance.yml

cfgitsync / CFinstance-deploy.yml

sbx_user1051 Add AWS Cloudformation Deployment file for CFInstance stack

Code Blame 3 lines (3 loc) · 60 Bytes Code 55% faster with GitHub Copilot

```
1 template-file-path: ec2instance.yml
2 parameters: {}
3 tags: {}
```

22. Update the CloudFormation (CFN) template to include the creation of an Elastic IP and attach it to the instance

devopstronaut / cfgitsync

Issues Pull requests Actions Projects Wiki Security Insights Settings

ance-deploy.yml README.md instance.yml

cfgitsync / ec2instance.yml in main

Edit Preview Code 55% faster with GitHub Copilot

```
1 AWSTemplateFormatVersion: '2010-09-09'
2 Description: Basic CloudFormation Template to create an EC2 Instance
3 Resources:
4   CFInstance:
5     Type: AWS::EC2::Instance
6     Properties:
7       InstanceType: t3.micro
8       KeyName: cfinstance
9       ImageId: ami-02af70169146bbdd3
10      SubnetId: subnet-0f4e64ce0a813bf21
11      SecurityGroupIds:
12        - sg-086bbc46373ed734e
13   ElasticIP:
14     Type: AWS::EC2::EIP
15     Properties:
16       InstanceId: !Ref CFInstance
17
18
```

23. Commit the changes to the repository



ce.vml in main

Commit changes

Commit message

Update ec2instance.yml

Extended description

Add an optional extended description..

Commit directly to the main branch
 Create a new branch for this commit and start a pull request
[Learn more about pull requests](#)

Cancel **Commit changes**

24. After committing the changes, Git sync will automatically start provisioning the resources based on the latest commit

CloudFormation > Stacks > CFInstance

CFInstance

Stacks (1)

Filter status Active View nested < 1 >

Stacks

CFInstance 2024-08-10 00:03:30 UTC+0100 UPDATE_COMPLETE

Stack info Events Resources Outputs Parameters Template Changesets **Git sync - new**

To provision resources with your Git provider, AWS uses CodeStar Connections APIs, which are not in scope for compliance programmes. For more information, see [AWS Services in Scope by Compliance Program](#).

Git sync - new Info

Configure and sync updates for your stack from the linked Git repository.

Repository cfgitsync	Deployment file path CFInstance-deploy.yml	Git sync <input checked="" type="radio"/> Enabled
Repository provider GitHub	Repository sync status <input checked="" type="radio"/> Succeeded	Provisioning status <input type="radio"/> In progress
Branch main	Repository sync status message Starting syncs for commit a6b433dc4a849fc8c9f711ef88bce3da61e7a3a	

Latest sync events (1)

Date	Commit ID	Event	Event type
2024-08-10 00:25:58 UTC+0100	a6b433dc	Clone started	<input type="radio"/> CLONE_STARTED

25. Provisioning is complete with the latest commit. You can cross-check the commit ID to verify



Stack info | Events | Resources | Outputs | Parameters | Template | Changesets | **Git sync - new**

To provision resources with your Git provider, AWS uses CodeStar Connections APIs, which are not in scope for compliance programmes. For more information, see [AWS Services in Scope by Compliance Program](#).

Git sync - new [Info](#)

Configure and sync updates for your stack from the linked Git repository.

[Edit](#)

[Retry latest commit](#)

[Disconnect](#)

[C](#)

Repository	Deployment file path	Git sync
cfgsync	CFinstance-deploy.yml	Enabled
Repository provider	Repository sync status	Provisioning status
GitHub	Succeeded	Succeeded

Branch

main

Repository sync status message

Starting syncs for commit

a6b433dc4a849fca8c9f711ef88bce3da61e7a3a

Latest sync events (6)			
Date	Commit ID	Event	Event type
2024-08-10 00:26:40 UTC+0100	a6b433dc	Changeset execution succeeded and stack updated successfully.	STACK_UPDATE_SUCCEEDED
2024-08-10 00:26:17 UTC+0100	a6b433dc	Waiting for changeset execution to finish.	CHANGESET_EXECUTION_IN_PROGRESS
2024-08-10 00:26:16 UTC+0100	a6b433dc	Changeset creation succeeded.	CHANGESET_CREATION_SUCCEEDED
2024-08-10 00:26:00 UTC+0100	a6b433dc	Creating a changeset.	CHANGESET_CREATION_IN_PROGRESS
2024-08-10 00:25:59 UTC+0100	a6b433dc	Uploaded CloudFormation template.	TEMPLATE_BUNDLED

Validation:

26. The Elastic IP has been created by the CloudFormation stack

EC2 > Elastic IP addresses > 13.51.51.4	Actions Associate Elastic IP address																																
13.51.51.4																																	
Summary																																	
<table><tr><td>Allocated IPv4 address</td><td>Type</td><td>Allocation ID</td><td>Reverse DNS record</td></tr><tr><td>13.51.51.4</td><td>Public IP</td><td>epalloc-0219b41cfdd88bf1f</td><td>-</td></tr><tr><td>Association ID</td><td>Scope</td><td>Associated instance ID</td><td>Private IP address</td></tr><tr><td>eipassoc-0ca66df218b021b56</td><td>VPC</td><td>i-09e47bc3b1fb54e47</td><td>172.31.11.74</td></tr><tr><td>Network interface ID</td><td>Network interface owner account ID</td><td>Public DNS</td><td>NAT Gateway ID</td></tr><tr><td>eni-01807610e4909564e</td><td>851725583489</td><td>ec2-13-51-51-4.eu-north-1.compute.amazonaws.com</td><td>-</td></tr><tr><td>Address pool</td><td>Network border group</td><td></td><td></td></tr><tr><td>Amazon</td><td>eu-north-1</td><td></td><td></td></tr></table>		Allocated IPv4 address	Type	Allocation ID	Reverse DNS record	13.51.51.4	Public IP	epalloc-0219b41cfdd88bf1f	-	Association ID	Scope	Associated instance ID	Private IP address	eipassoc-0ca66df218b021b56	VPC	i-09e47bc3b1fb54e47	172.31.11.74	Network interface ID	Network interface owner account ID	Public DNS	NAT Gateway ID	eni-01807610e4909564e	851725583489	ec2-13-51-51-4.eu-north-1.compute.amazonaws.com	-	Address pool	Network border group			Amazon	eu-north-1		
Allocated IPv4 address	Type	Allocation ID	Reverse DNS record																														
13.51.51.4	Public IP	epalloc-0219b41cfdd88bf1f	-																														
Association ID	Scope	Associated instance ID	Private IP address																														
eipassoc-0ca66df218b021b56	VPC	i-09e47bc3b1fb54e47	172.31.11.74																														
Network interface ID	Network interface owner account ID	Public DNS	NAT Gateway ID																														
eni-01807610e4909564e	851725583489	ec2-13-51-51-4.eu-north-1.compute.amazonaws.com	-																														
Address pool	Network border group																																
Amazon	eu-north-1																																
Tags(3)																																	
<table><tr><td>Key</td><td>Value</td></tr><tr><td>aws:cloudformation:stack-id</td><td>arn:aws:cloudformation:eu-north-1:851725583489:stack/CFInstance/98219390-56a5-11ef-97de-06cce2b53075</td></tr><tr><td>aws:cloudformation:stack-name</td><td>CFInstance</td></tr><tr><td>aws:cloudformation:logical-id</td><td>ElasticIP</td></tr></table>		Key	Value	aws:cloudformation:stack-id	arn:aws:cloudformation:eu-north-1:851725583489:stack/CFInstance/98219390-56a5-11ef-97de-06cce2b53075	aws:cloudformation:stack-name	CFInstance	aws:cloudformation:logical-id	ElasticIP																								
Key	Value																																
aws:cloudformation:stack-id	arn:aws:cloudformation:eu-north-1:851725583489:stack/CFInstance/98219390-56a5-11ef-97de-06cce2b53075																																
aws:cloudformation:stack-name	CFInstance																																
aws:cloudformation:logical-id	ElasticIP																																

<https://www.linkedin.com/in/mahendran-selvakumar-36444a77/>

27. The Elastic IP has been attached to the EC2 instance

Instance summary for i-09d47bc3b1fb54e47

Updated less than a minute ago

Instance ID	i-09d47bc3b1fb54e47	Public IPv4 address	13.51.51.4 open address	Private IPv4 addresses	172.31.11.74
IPv6 address	-	Instance state	Running	Public IPv4 DNS	ec2-13-51-51-4.eu-north-1.compute.amazonaws.com open address
Hostname type	IP name: ip-172-31-11-74.eu-north-1.compute.internal	Private IP DNS name (IPv4 only)	ip-172-31-11-74.eu-north-1.compute.internal	Elastic IP addresses	13.51.51.4 [Public IP]
Answer private resource DNS name	-	Instance type	t3.micro	AWS Compute Optimizer finding	Opt-in to AWS Compute Optimizer for recommendations. Learn more
Auto-assigned IP address	-	VPC ID	vpc-02aa3080a2b7a0b77	Subnet ID	subnet-0f4e64ce0a813bf21
IAM Role	-	Instance ARN	arn:aws:ec2:eu-north-1:851725583489:instance/i-09d47bc3b1fb54e47	Auto Scaling Group name	-
IMDSv2	Required				

Tags

Key	Value
aws:cloudformation:logical-id	CFInstance
aws:cloudformation:stack-name	CFInstance
aws:cloudformation:stack-id	arn:aws:cloudformation:eu-north-1:851725583489:stack/CFInstance/98219390-56a3-11ef-97de-06cce2b53075

Keep Learning, Keep Deploying!!!

Feel free to reach out to me, if you have any other queries or suggestions

Stay connected on LinkedIn <https://www.linkedin.com/in/mahendran-selvakumar-36444a77>