

AWS Deployment Steps

Code Commit :

The screenshot shows the AWS CodeCommit service interface. The left sidebar has a 'Source' section under 'CodeCommit' with options like 'Getting started', 'Repositories', 'Approval rule templates', 'Artifacts', 'Build', 'Deploy', 'Pipeline', and 'Settings'. The main area shows a list of repositories with one entry: 'mfpe-auth' (description: 'authorization module', last modified: '44 minutes ago'). There are buttons for 'Clone URL' (HTTPS, SSH, HTTPS (GRC)), 'View repository', 'Delete repository', and a prominent orange 'Create repository' button.

The screenshot shows the 'Create repository' wizard. It starts with a summary step: 'Create a secure repository to store and share your code. Begin by typing a repository name and a description for your repository. Repository names are included in the URLs for that repository.' Below this, there's a 'Repository settings' section with fields for 'Repository name' (set to 'mfpe-rules'), 'Description - optional' (set to 'rules module'), and 'Tags' (an 'Add' button). At the bottom, there's an optional checkbox for 'Enable Amazon CodeGuru Reviewer for Java and Python - optional' with a note about getting recommendations for Java and Python code. The bottom right has 'Cancel' and 'Create' buttons.

HTTP Git Credentials:

The screenshot shows the AWS IAM SSH keys interface. On the left, a sidebar lists various IAM management options like Dashboard, Access management, and Users. The main area displays 'SSH keys for AWS CodeCommit' and 'HTTPS Git credentials for AWS CodeCommit'. Under 'HTTPS Git credentials for AWS CodeCommit', a table shows one credential entry:

User name	Status	Created
30intcde02517-at-941516478045	Active	2021-07-30 12:56 UTC+0530

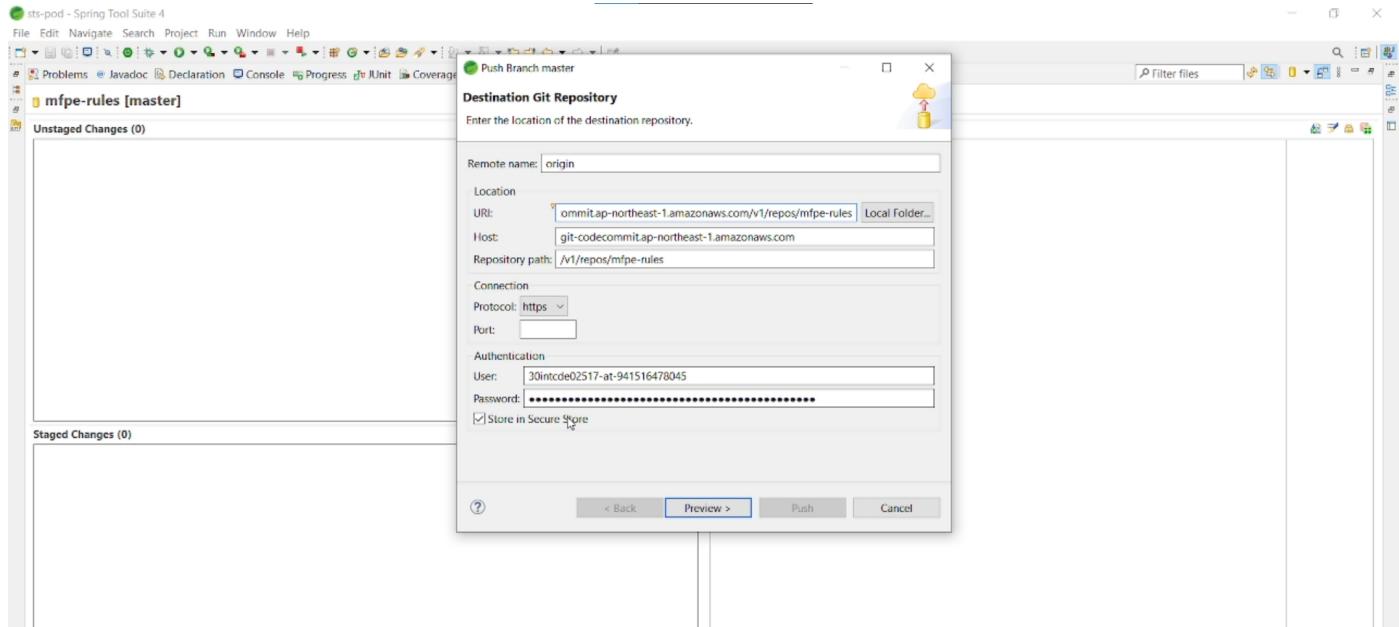
Below this, there's a section for 'Credentials for Amazon Keyspaces (for Apache Cassandra)' which is currently empty.

A modal window titled 'Generate credentials' is displayed. It contains a success message: 'Your new credentials are available'. Below it, instructions say 'Save your user name and password now (or download a credentials file)'. It also notes that the password can only be viewed or downloaded once. The modal shows the generated credentials:

User name	30intcde02506-at-941516478045
Password	***** Show

At the bottom, there's a 'Download credentials' button and a 'Close' button.

Pushing to Git Repository:



Name	Description	Last modified	Clone URL
mfpe-customer	-	3 hours ago	HTTPS SSH HTTPS (GRC)
mfpe-transaction	-	3 hours ago	HTTPS SSH HTTPS (GRC)
mfpe-account	Account module	3 hours ago	HTTPS SSH HTTPS (GRC)
mfpe-rules	rules module	3 hours ago	HTTPS SSH HTTPS (GRC)
mfpe-auth	authorization module	8 hours ago	HTTPS SSH HTTPS (GRC)

Elastic Container Registry:

The screenshot shows the 'Create repository' page in the AWS ECR console. In the 'General settings' section, the 'Visibility settings' dropdown is set to 'Private'. The 'Repository name' field contains 'ij025-pod1-auth-ms-registry'. A note at the bottom states: 'Once a repository is created, the visibility setting of the repository can't be changed.'

Visibility settings [Info](#)
Choose the visibility setting for the repository.
 Private
Access is managed by IAM and repository policy permissions.
 Public
Publicly visible and accessible for image pulls.

Repository name
Provide a concise name. A developer should be able to identify the repository contents by the name.
941516478045.dkr.ecr.ap-northeast-1.amazonaws.com/

27 out of 256 characters maximum (2 minimum). The name must start with a letter and can only contain lowercase letters, numbers, hyphens, underscores, and forward slashes.

Tag immutability [Info](#)
Enable tag immutability to prevent image tags from being overwritten by subsequent image pushes using the same tag. Disable tag immutability to allow image tags to be overwritten.
 Disabled

Once a repository is created, the visibility setting of the repository can't be changed.

The screenshot shows the 'Private repositories' list page in the AWS ECR console. It displays five private repositories, each with its name, URI, creation date, tag immutability status, scan on push status, and encryption type (AES-256). The table includes columns for Repository name, URI, Created at, Tag immutability, Scan on push, and Encryption type.

Repository name	URI	Created at	Tag immutability	Scan on push	Encryption type
ij025-pod1-account-ms-registry	941516478045.dkr.ecr.ap-northeast-1.amazonaws.com/ij025-pod1-account-ms-registry	Jul 30, 2021 06:12:21 PM	Disabled	Disabled	AES-256
ij025-pod1-auth-ms-registry	941516478045.dkr.ecr.ap-northeast-1.amazonaws.com/ij025-pod1-auth-ms-registry	Jul 30, 2021 01:01:50 PM	Disabled	Disabled	AES-256
ij025-pod1-customer-ms-registry	941516478045.dkr.ecr.ap-northeast-1.amazonaws.com/ij025-pod1-customer-ms-registry	Jul 30, 2021 06:12:40 PM	Disabled	Disabled	AES-256
ij025-pod1-rules-ms-registry	941516478045.dkr.ecr.ap-northeast-1.amazonaws.com/ij025-pod1-rules-ms-registry	Jul 30, 2021 06:12:49 PM	Disabled	Disabled	AES-256
ij025-pod1-transaction-ms-registry	941516478045.dkr.ecr.ap-northeast-1.amazonaws.com/ij025-pod1-transaction-ms-registry	Jul 30, 2021 06:13:12 PM	Disabled	Disabled	AES-256

Code Build:

Salesforce Services ▾ [Alt+S] 30intcde02517 @ 9415-1647-8045 ▾ Tokyo ▾ Support ▾

Developer Tools > CodeBuild > Build projects > Create build project

Create build project

Project configuration

Project name: ij025-pod1-auth-ms-build
A project name must be 2 to 255 characters. It can include the letters A-Z and a-z, the numbers 0-9, and the special characters - and _.

Description - optional: auth module

Build badge - optional: Enable build badge

Enable concurrent build limit - optional: Limit the number of allowed concurrent builds for this project.
 Restrict number of concurrent builds this project can start

► Additional configuration: tags

Source 1 - Primary

Source provider: AWS CodeCommit

Repository: mfpe-auth

Reference type: Choose the source version reference type that contains your source code.
 Branch
 Git tag
 Commit ID

Branch: Choose a branch that contains the code to build.
master

Commit ID - optional: Choose a commit ID. This can shorten the duration of your build.

Source version info: refs/heads/master
7386524a unit testing

► Additional configuration: Git clone depth, Git submodules

aws Services ▾ [Alt+S] 30intcde02517 @ 9415-1647-8045 ▾ Tokyo ▾ Support ▾

Developer Tools > CodeBuild > Build projects > ij025-pod1-auth-ms-build

ij025-pod1-auth-ms-build

Project created
You have successfully created the following project: ij025-pod1-auth-ms-build

Create a notification rule for this project

Notify Share Delete build project

Configuration

Source provider: AWS CodeCommit	Primary repository: mfpe-auth	Artifacts upload location: -	Build badge: Disabled
---------------------------------	-------------------------------	------------------------------	-----------------------

Build history Stop build View artifacts View logs Delete builds Retry build

Build history Status Source version Duration

Screenshot of the AWS CodeBuild build details page for build `ij025-pod1-auth-ms-build:0e980814-6d75-4e24-bb0b-7d631b2cb427`.

Build status:

Status	Initiator	Build ARN	Resolved source version
Succeeded	30intcde02517	arn:aws:codebuild:ap-northeast-1:941516478045:build/ij025-pod1-auth-ms-build:0e980814-6d75-4e24-bb0b-7d631b2cb427	5257b365c3647a6e332f0d71d1024f 6406d9483b

Logs: Showing the last 1000 lines of the build log. [View entire log](#) | [Tail logs](#)

Screenshot of the AWS CodeBuild build projects page.

Build projects:

Name	Source provider	Repository	Latest build status	Description	Last Modified
ij025-pod1-account-ms-build-1	AWS CodeCommit	mfpe-account	Succeeded	Account Service	6 hours ago
IJ025-POD1-Rules-MS-BUILD	AWS CodeCommit	mfpe-rules	Succeeded	Rules Module	6 hours ago
IJ025-Pod1-Customer-ms-build-2	AWS CodeCommit	mfpe-customer	Succeeded	Customer Module	7 hours ago
ij025-pod1-transaction-ms-build	AWS CodeCommit	mfpe-transaction	Succeeded	Transaction module	9 hours ago
ij025-pod1-auth-ms-build	AWS CodeCommit	mfpe-auth	Succeeded	auth module	2 days ago

Elastic Container Service - Cluster:

The screenshot shows the 'Configure cluster' step of the ECS cluster creation wizard. It includes fields for 'Cluster name*' (set to 'ij025-pod1'), 'Networking' (with an option to 'Create VPC' or use an existing one), 'Tags' (with 'Add key' and 'Add value' fields), and 'CloudWatch Container Insights' (with an option to 'Enable Container Insights'). At the bottom, there are buttons for '*Required', 'Cancel', 'Previous', and a prominent 'Create' button.

The screenshot shows the 'Launch status' page for the newly created cluster 'ij025-pod1'. It indicates that 1 of 1 container instances are launching. A success message states: 'ECS Cluster ij025-pod1 successfully created'. Navigation buttons 'Back' and 'View Cluster' are visible at the top.

The screenshot shows the 'Clusters' management page. It lists the single cluster 'ij025-pod1' which has 'CloudWatch monitoring' and 'Default Monitoring' enabled. The cluster details show 0 services, 0 running tasks, and 0 pending tasks. Under the EC2 tab, it shows 0 services, 0 running tasks, 0 pending tasks, and 0 EC2 container instances. The CPUUtilization and MemoryUtilization metrics are listed as 'No data'. The page also includes a 'View' dropdown with 'list' and 'card' options, and a 'view all' link.

Load Balancer:

The screenshot shows the AWS Elastic Load Balancing service selection interface. At the top, there's a search bar and navigation links for services, marketplace, and support. Below the header, a message states: "Elastic Load Balancing supports four types of load balancers: Application Load Balancers, Network Load Balancers, Gateway Load Balancers, and Classic Load Balancers. Choose the load balancer type that meets your needs. Learn more about which load balancer is right for you." Three main options are displayed in cards:

- Application Load Balancer**: Handles HTTP and HTTPS traffic. A "Create" button is present.
- Network Load Balancer**: Handles TCP, TLS, and UDP traffic. A "Create" button is present.
- Gateway Load Balancer**: Handles IP traffic. A "Create" button is present.

A fourth option, **Classic Load Balancer**, is shown below the others. On the far right, there's a "Cancel" link.

Configure Load Balancer :

The screenshot shows the "Configure Load Balancer" step of the AWS wizard. The top navigation bar includes links for "Configure Security Settings", "Configure Security Groups", "Configure Routing", "Register Targets", and "Review". The "Configure Load Balancer" tab is selected.

Step 1: Configure Load Balancer

Basic Configuration

To configure your load balancer, provide a name, select a scheme, specify one or more listeners, and select a network. The default configuration is an Internet-facing load balancer in the selected network with a listener that receives HTTP traffic on port 80.

Fields shown:

- Name**: jj025-pod1-loadbalancer
- Scheme**: internet-facing
- IP address type**: ipv4

Listeners

A listener is a process that checks for connection requests, using the protocol and port that you configured.

Load Balancer Protocol	Load Balancer Port
HTTP	80

Add listener button

Availability Zones

Configuration details for availability zones are partially visible at the bottom of the screen.

Buttons: **Cancel** and **Next: Configure Security Settings**

Salesforce Services ▾ [Alt+S] 30Intcde02517 @ 9415-1647-8045 ▾ Tokyo ▾ Support ▾

1. Configure Load Balancer 2. Configure Security Settings 3. Configure Security Groups 4. Configure Routing 5. Register Targets 6. Review

Step 1: Configure Load Balancer

HTTP X

Add listener

Availability Zones

Specify the Availability Zones to enable for your load balancer. The load balancer routes traffic to the targets in these Availability Zones only. You can specify only one subnet per Availability Zone. You must specify subnets from at least two Availability Zones to increase the availability of your load balancer.

VPC	Subnet	IPv4 address	Status
vpc-7ca75a1a (172.31.0.0/16) (default)	subnet-8572cdcc	Assigned by AWS	
ap-northeast-1a	subnet-c7c1289d	Assigned by AWS	
ap-northeast-1c	subnet-c85adce3	Assigned by AWS	

Add-on services

Cancel Next: Configure Security Settings

Configure Routing:

1. Configure Load Balancer 2. Configure Security Settings 3. Configure Security Groups 4. Configure Routing 5. Register Targets 6. Review

Step 4: Configure Routing

Target group X

Name

Target type IP Instance Lambda function

Protocol X

Port X

Protocol version HTTP1 Send requests to targets using HTTP/1.1. Supported when the request protocol is HTTP/1.1 or HTTP/2.
 HTTP2 Send requests to targets using HTTP/2. Supported when the request protocol is HTTP/2 or gRPC, but gRPC-specific features are not available.
 gRPC Send requests to targets using gRPC. Supported when the request protocol is gRPC.

Health checks

Protocol X

Cancel Previous Next: Register Targets

Load Balancer Created:

The screenshot shows the AWS Elastic Load Balancing (ELB) service in the AWS Management Console. On the left, the navigation pane includes 'EC2 Dashboard', 'Events', 'Tags', 'Limits', 'Instances' (with 'Instances New'), 'Instance Types', 'Launch Templates', 'Spot Requests', 'Savings Plans', 'Reserved Instances New', 'Dedicated Hosts', and 'Capacity Reservations'. Under 'Images', there is an 'AMIs' section. Under 'Elastic Block Store', there are 'Volumes' and 'Snapshots'. A 'Lifecycle Manager New' option is also present. The main content area is titled 'Create Load Balancer' and shows a table with one row of data. The table columns are 'Name', 'DNS name', 'State', 'VPC ID', 'Availability Zones', 'Type', and 'Actions'. The single row shows 'ij025-pod1-loadbalancer' as the Name, 'ij025-pod1-loadbalancer-491...' as the DNS name, 'Provisioning' as the State, 'vpc-7ca75a1a' as the VPC ID, 'ap-northeast-1a, ap-nor...' as the Availability Zones, 'application' as the Type, and an 'Actions' button. Below the table, a detailed view of the load balancer 'ij025-pod1-loadbalancer' is shown. It has tabs for 'Description', 'Listeners', 'Monitoring', 'Integrated services', and 'Tags'. The 'Basic Configuration' section shows 'Name' as 'ij025-pod1-loadbalancer' and 'ARN' as 'arn:aws:elasticloadbalancing:ap-northeast-1:941516478045:loadbalancer/app/ij025-pod1-loadbalancer/1ea1498101553390'. There is also a 'Edit' icon.

Task Definition:

Create new Task definition:

The screenshot shows the AWS Elastic Container Service (ECS) console. The left sidebar includes 'Amazon ECS Clusters', 'Task Definitions' (which is selected and highlighted in orange), 'Account Settings', 'Amazon EKS Clusters', 'Amazon ECR Repositories', 'AWS Marketplace Discover software', and 'Subscriptions'. The main content area is titled 'Task Definitions' and includes a 'Create new Task Definition' button, a 'Create new revision' button, and an 'Actions' dropdown menu. A status message 'Last updated on August 2, 2021 10:46:27 AM (0m ago)' is displayed. The table below shows one task definition entry. The table columns are 'Status' (ACTIVE), 'INACTIVE', 'Filter in this page', 'Task Definition', and 'Latest revision status'. The single row shows 'ACTIVE' as the Status, 'INACTIVE' as the Revision status, and 'No results' as the Latest revision status. There is also an 'Edit' icon.

Container:

The screenshot shows the AWS CloudWatch Metrics console with the 'Add container' dialog open. The dialog is titled 'Add container' and has a 'Standard' configuration selected. The 'Container name*' field contains 'auth-ms-container'. The 'Image*' field shows the ARN: '941516478045.dkr.ecr.ap-northeast-1.amazonaws.com/ij025-pod1-auth-ms-registry'. Below these fields are sections for 'Private repository authentication*', 'Memory Limits (MiB)', and 'Port mappings'. The 'Memory Limits (MiB)' section includes a note about hard and soft limits and a recommendation of 300-500 MiB for web applications. The 'Port mappings' section shows a single mapping with 'Container port' set to 1024 and 'Protocol' set to 'tcp'. At the bottom of the dialog are 'Cancel' and 'Add' buttons.

The screenshot shows the AWS CloudWatch Metrics console with the 'Launch Status' and 'Create Task Definition' sections. The 'Launch Status' section indicates 'Task definition status - 2 of 2 completed'. The 'Create Task Definition' section shows a log entry: 'ij025-pod1-auth-ms-task succeeded'. The 'Create CloudWatch Log Group' section shows a success message: 'CloudWatch Log Group created' and 'CloudWatch Log Group /ecs/ij025-pod1-auth-ms-task'. At the bottom right are 'Back' and 'View task definition' buttons.

The screenshot shows the AWS ECS Cluster details page for 'ij025-pod1'. The cluster ARN is arn:aws:ecs:ap-northeast-1:941516478045:cluster/ij025-pod1 and the status is ACTIVE. There are 0 registered container instances, 0 pending tasks, 0 running tasks, 0 active services, and 0 draining services. The 'Services' tab is selected, showing a table with columns: Service Name, Status, Service type..., Task Defin..., Desired tas..., Running ta..., Launch typ..., Platform ve... . The table is empty, displaying 'No results'. Other tabs include Tasks, ECS Instances, Metrics, Scheduled Tasks, Tags, and Capacity Providers.

Create new Service for the task:

The screenshot shows the 'Create service' step in the AWS ECS wizard. It displays the 'Launch Status' section, which shows 'ECS Service status - 1 of 1 completed'. Below it are links for 'Configure Task Networking', 'Create Load Balancer', and 'Create Service'. The 'Create Service' link is highlighted. A message box indicates that the service has been created: 'Service created. Tasks will start momentarily. View: ij025-pod1-auth-ms-service'. The 'View Service' button is visible at the bottom right.

Additional integrations you can connect to your ECS service

Code Pipeline

Setup a CI/CD process from your service. You can build from source or have an ECR repository as the source for your deployment.

[Create a pipeline](#)

[Back](#) [View Service](#)

ECS Service status - 3 of 3 completed

Configure Task Networking

Create Load Balancer

Target Group: ij025-pod1-transaction-ms

✓ **Target Group created**
Target Group created. Waiting to create listener/rule. View: [ij025-pod1-transaction-ms](#)

Rule: 80:HTTP /transaction-ms/*:5

✓ **Rule created**
Rule created. Waiting to create service. View in load balancer: [ij025-pod1-loadbalancer](#)

Create Service

Create service: ij025-pod1-transaction-ms-service

✓ **Service created**
Service created. Tasks will start momentarily. View: [ij025-pod1-transaction-ms-service](#)

Services for 5 Microservices have been created:

New ECS Experience
Tell us what you think

Amazon ECS

Clusters

Task Definitions

Account Settings

Amazon EKS

Clusters

Amazon ECR

Repositories

AWS Marketplace

Discover software

Subscriptions

Clusters

An Amazon ECS cluster is a regional grouping of one or more container instances on which you can run task requests. Each account receives a default cluster the first time you use the Amazon ECS service. Clusters may contain more than one Amazon EC2 instance type.

For more information, see the [ECS documentation](#).

[Create Cluster](#) [Get Started](#)

View list card view all ↻

ij025-pod1 > CloudWatch monitoring Default Monitoring

FARGATE

5 Services	5 Running tasks	0 Pending tasks
------------	-----------------	-----------------

EC2

0 Services	0 Running tasks	0 Pending tasks	No data CPUUtilization	No data MemoryUtilization	0 EC2 container instances
------------	-----------------	-----------------	------------------------	---------------------------	---------------------------

Sales Services ▾ Search for services, features, marketplace products, and docs [Alt+S] 30intcde02517 @ 9415-1647-8045 ▾ Tokyo ▾ Support ▾

Clusters

Task Definitions
Account Settings
Amazon EKS
Clusters
Amazon ECR
Repositories
AWS Marketplace
Discover software
Subscriptions

Get a detailed view of the resources on your cluster.

Cluster ARN arn:aws:ecs:ap-northeast-1:941516478045:cluster/ij025-pod1
Status ACTIVE

Registered container instances 0

Pending tasks count 0 Fargate, 0 EC2, 0 External
Running tasks count 5 Fargate, 0 EC2, 0 External
Active service count 5 Fargate, 0 EC2, 0 External
Draining service count 0 Fargate, 0 EC2, 0 External

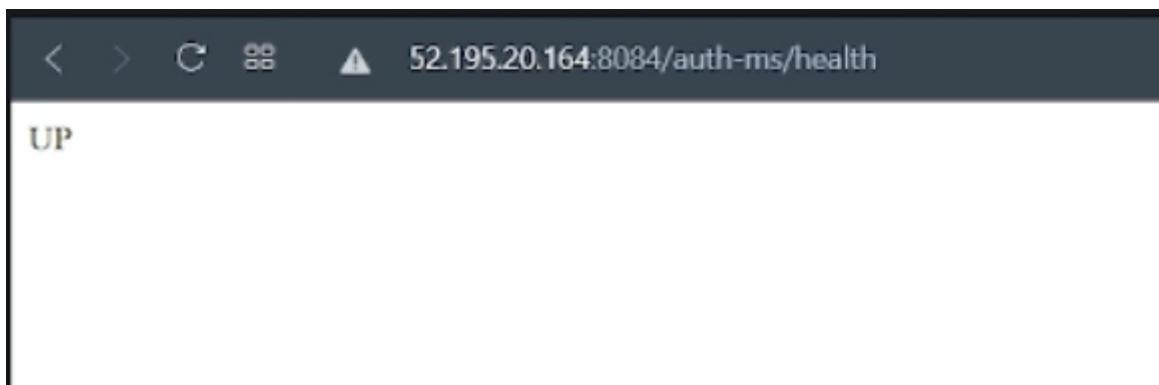
Services Tasks ECS Instances Metrics Scheduled Tasks Tags Capacity Providers

Create Update Delete Actions ▾ Last updated on August 2, 2021 9:29:49 PM (0m ago)

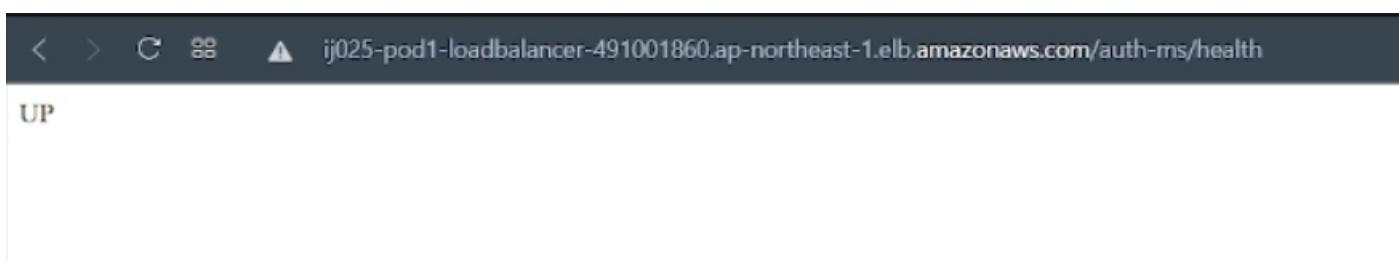
Filter in this page	Launch type	ALL	Service type	ALL	< 1-5 >						
<input type="checkbox"/> Service Name	Status	Service typ...	Task Definit...	Desired tas...	Running ta...	Launch typ...	Platform ve...				
<input type="checkbox"/> ij025-Pod1-Customer-ms-service	ACTIVE	REPLICA	ij025-Pod1-...	1	1	FARGATE	LATEST(1.4...				
<input type="checkbox"/> ij025-pod1-transaction-ms-service	ACTIVE	REPLICA	ij025-pod1-tr...	1	1	FARGATE	LATEST(1.4...				
<input type="checkbox"/> ij025-pod1-auth-ms-service	ACTIVE	REPLICA	ij025-pod1-a...	1	1	FARGATE	LATEST(1.4...				
<input type="checkbox"/> ij025-Pod1-Rules-ms-service	ACTIVE	REPLICA	ij025-Pod1-...	1	1	FARGATE	LATEST(1.4...				
<input type="checkbox"/> ij025-pod1-account-ms-service	ACTIVE	REPLICA	ij025-pod1-a...	1	1	FARGATE	LATEST(1.4...				

Endpoints have been health checked:

With Public IP:



With DNS:



DNS Links tested with postman:

Authorization-ms:

A Postman screenshot showing a successful login request to the Authorization microservice. The request is a POST to `http://ij025-pod1-loadbalancer-491001860.ap-northeast-1.elb.amazonaws.com/auth-ms/login`. The body contains a JSON payload with fields `userid`, `password`, and `role`. The response status is `202 Accepted`, and the response body includes a generated `authToken`.

```
POST http://ij025-pod1-loadbalancer-491001860.ap-northeast-1.elb.amazonaws.com/auth-ms/login
Content-Type: application/json
{
  "userid": "EMPLOYEE101",
  "password": "emp",
  "role": "EMPLOYEE"
}

Body
Status: 202 Accepted
{
  "userid": "EMPLOYEE101",
  "authToken": "eyJhbGciOiJIUzI1NiJ9eyJzdWIiOiJFTVBMT1lFRTEwMSIsImV4cCI6MTYyNz...",
  "role": "EMPLOYEE"
}
```

Customer-ms:

A Postman screenshot showing a successful deletion of a customer from the Customer microservice. The request is a DELETE to `http://ij025-pod1-loadbalancer-491001860.ap-northeast-1.elb.amazonaws.com/customer/deleteCustomer/CUSTOMER101`. The response status is `200 OK`, and the response body indicates the message `"CUSTOMER DELETED"`.

```
DELETE http://ij025-pod1-loadbalancer-491001860.ap-northeast-1.elb.amazonaws.com/customer/deleteCustomer/CUSTOMER101
Content-Type: application/json
Authorization: eyJhbGciOiJIUzI1NiJ9eyJzdWIiOiJFTVBMT1lFRTEwMSIsImV4cCI6MTYyNz...
Body
Status: 200 OK
{
  "message": "CUSTOMER DELETED"
}
```

Account-ms:

A Postman screenshot showing a successful retrieval of an account from the Account microservice. The request is a GET to `http://ij025-pod1-loadbalancer-491001860.ap-northeast-1.elb.amazonaws.com/account-ms/getAccount/100000002`. The response status is `200 OK`, and the response body provides detailed account information.

```
GET http://ij025-pod1-loadbalancer-491001860.ap-northeast-1.elb.amazonaws.com/account-ms/getAccount/100000002
Content-Type: application/json
Authorization: eyJhbGciOiJIUzI1NiJ9eyJzdWIiOiJFTVBMT1lFRTEwMSIsImV4cCI6MTYyNz...
Body
Status: 200 OK
{
  "accountId": 100000002,
  "customerId": "CUSTOMER102",
  "currentBalance": 2000,
  "accountType": "Current",
  "openingDate": "2021-10-08T18:30:00.000+00:00",
  "ownerName": "Albert Einstein",
  "transactions": null
}
```

Transaction-ms:

The screenshot shows a Postman request for a GET operation. The URL is `http://ij025-pod1-loadbalancer-491001860.ap-northeast-1.elb.amazonaws.com/transaction-ms/getAllTransByAcctId/1000000001`. The Headers tab is selected, containing two entries: Content-Type (application/json) and Authorization (a long token). The Body tab is selected, showing a JSON response with one item in an array. The item has fields: id (1), sourceAccountId (1000000001), sourceOwnerName ("Isaac Newton"), targetAccountId (1000000002), targetOwnerName ("Albert Einstein"), amount (100), initiationDate ("2019-04-01T10:30:00"), and reference ("transfer").

```
1 [ 2 { 3   "id": 1, 4   "sourceAccountId": 1000000001, 5   "sourceOwnerName": "Isaac Newton", 6   "targetAccountId": 1000000002, 7   "targetOwnerName": "Albert Einstein", 8   "amount": 100, 9   "initiationDate": "2019-04-01T10:30:00", 10  "reference": "transfer" 11 } 12 ]
```

Rules-ms:

The screenshot shows a Postman request for a POST operation. The URL is `http://ij025-pod1-loadbalancer-491001860.ap-northeast-1.elb.amazonaws.com/rules/serviceCharges`. The Headers tab is selected, containing two entries: Content-Type (application/json) and Authorization (a long token). The Body tab is selected, showing a JSON response with two items in an array. Each item represents a bank account with fields: accountId (1000000001 or 1000000002), customerId (CUSTOMER101 or CUSTOMER102), currentBalance (80000 or 2000), accountType ("Savings" or "Current"), openingDate ("2021-10-08T18:30:00.000+00:00"), ownerName ("Isaac Newton" or "Albert Einstein"), and transactions (null).

```
1 [ 2 { 3   "accountId": 1000000001, 4   "customerId": "CUSTOMER101", 5   "currentBalance": 80000, 6   "accountType": "Savings", 7   "openingDate": "2021-10-08T18:30:00.000+00:00", 8   "ownerName": "Isaac Newton", 9   "transactions": null 10 }, 11 { 12   "accountId": 1000000002, 13   "customerId": "CUSTOMER102", 14   "currentBalance": 2000, 15   "accountType": "Current", 16   "openingDate": "2021-10-08T18:30:00.000+00:00", 17   "ownerName": "Albert Einstein", 18   "transactions": null 19 } 20 ]
```