

Installation Guide - CAT-IM AWS Marketplace

The purpose of this guide is to offer a comprehensive walkthrough for deploying CAT-IM via the AWS Marketplace and integrating it seamlessly with AWS native services.

1. Prerequisites
2. Provision RDS Postgres
 - 2.1 Create Cloudformation manifest
 - 2.2 Apply Cloudformation stack
- 3 Provision Cognito
 - 3.1 Create Cloudformation manifest
 - 3.2 Apply Cloudformation stack
 - 3.3 Create Cognito user
 - 3.4 Add Cognito user to Admin group
4. Install & Configure Temporal
5. Install & Configure CAT-IM

1. Prerequisites

- **EKS cluster:** operational cluster deployed onto an existing VPC/Subnets
- **Kubernetes Ingress Controller:** Configured in cluster. The IntellectEU recommended controller is Traefik
- **Helm:** Helm v3 installed on your local machine
- **SFTP Broker:** operational broker, for file service interactions
- **AWS CLI:** v2 installed. Required for CloudFormation deployments
- **Route53 Zone:** already configured, so we can set custom DNS aliases for our services
- **VPC:** An existing VPC configured with private/public subnets

2. Provision RDS Postgres

Deploy an RDS Postgres Instance. This can be through various methods (OpenTofu/Terraform, AWS CLI, etc), but for ease of using AWS native solutions, this section will provision an RDS cluster through a Cloudformation template.

2.1 Create Cloudformation manifest

Create a file (`rds-stack.yaml`) with the contents below:

```
1 AWSTemplateFormatVersion: 2010-09-09
2 Description: Provision an RDS PostgreSQL instance in an existing VPC and subnet group
3 Parameters:
4   VpcId:
5     Type: AWS::EC2::VPC::Id
```

```
6     Description: The ID of the existing VPC
7 VpcCidr:
8     Type: String
9     Description: CIDR block of the VPC (used as default for SG ingress rules)
10 SubnetIds:
11     Type: List<AWS::EC2::Subnet::Id>
12     Description: A list of existing Subnet IDs for the DB subnet group (must be in the same
VPC)
13 DBName:
14     Type: String
15     Default: message
16     Description: The name of the initial PostgreSQL database
17 DBUsername:
18     Type: String
19     Default: masteruser
20     Description: The master username for PostgreSQL
21 DBPassword:
22     Type: String
23     NoEcho: true
24     Description: The master user password for PostgreSQL
25 DBAllocatedStorage:
26     Type: Number
27     Default: 20
28     Description: The size of the database (GB)
29 DBInstanceClass:
30     Type: String
31     Default: db.t3.micro
32     AllowedValues:
33         - db.t3.micro
34         - db.t3.small
35         - db.t3.medium
36         - db.t3.large
37         - db.m5.large
38         - db.m5.xlarge
39     Description: The instance class for the RDS instance
40 DBEngineVersion:
41     Type: String
42     Default: "14.17"
43     Description: PostgreSQL engine version
44 HostedZoneId:
45     Type: String
46     Description: The Route53 Hosted Zone ID where you want to create the record
47 RecordName:
48     Type: String
49     Description: The DNS record name (e.g. db.example.com)
50
51 Resources:
52     DBSubnetGroup:
53         Type: AWS::RDS::DBSubnetGroup
54         Properties:
55             DBSubnetGroupDescription: Subnet group for PostgreSQL RDS instance
56             SubnetIds: !Ref SubnetIds
57     DBSecurityGroup:
58         Type: AWS::EC2::SecurityGroup
59         Properties:
60             GroupDescription: Security group for PostgreSQL RDS
61             VpcId: !Ref VpcId
62             SecurityGroupIngress:
```

```

63     - IpProtocol: tcp
64     FromPort: 5432
65     ToPort: 5432
66     CidrIp: !Ref VpcCidr
67 PostgresDB:
68   Type: AWS::RDS::DBInstance
69   Properties:
70     DBName: !Ref DBName
71     AllocatedStorage: !Ref DBAllocatedStorage
72     DBInstanceClass: !Ref DBInstanceClass
73     Engine: postgres
74     EngineVersion: !Ref DBEngineVersion
75     MasterUsername: !Ref DBUsername
76     MasterUserPassword: !Ref DBPassword
77     VPCSecurityGroups:
78       - !Ref DBSecurityGroup
79     DBSubnetGroupName: !Ref DBSubnetGroup
80     PubliclyAccessible: false
81     BackupRetentionPeriod: 7
82     DeletionProtection: false
83     StorageEncrypted: true
84     MultiAZ: false
85
86 DBRecordSet:
87   Type: AWS::Route53::RecordSet
88   Properties:
89     HostedZoneId: !Ref HostedZoneId
90     Name: !Ref RecordName
91     Type: CNAME
92     TTL: 300
93     ResourceRecords:
94       - !GetAtt PostgresDB.Endpoint.Address

```

2.2 Apply Cloudformation stack

With the use of AWS CLI, deploy the Cloudformation stack, ensuring you replace placeholders with relevant values:

```

1 aws cloudformation create-stack \
2   --stack-name rds-stack \
3   --template-body file://rds-stack.yaml \
4   --capabilities CAPABILITY_NAMED_IAM \
5   --parameters \
6     ParameterKey=VpcId,ParameterValue={REPLACEME} \
7     ParameterKey=SubnetIds,ParameterValue="{REPLACEME_WITH_COMMA_DELIMITED_SUBNET_IDS}" \
8     ParameterKey=VpcCidr,ParameterValue={REPLACEME} \
9     ParameterKey=DBUsername,ParameterValue={REPLACEME} \
10    ParameterKey=DBPassword,ParameterValue={REPLACEME} \
11    ParameterKey=HostedZoneId,ParameterValue={REPLACEME} \
12    ParameterKey=RecordName,ParameterValue={REPLACEME}

```

3 Provision Cognito

3.1 Create Cloudformation manifest

Create a file (`cognito-stack.yaml`) with the contents below:

```
1 AWSTemplateFormatVersion: 2010-09-09
2 Description: Cognito User Pool with domain, groups and Resource server with custom scopes
3
4 Parameters:
5   UserPoolName:
6     Type: String
7     Default: catim-pool
8     Description: The name of the Cognito User Pool
9   AppClientName:
10    Type: String
11    Default: catim-ui
12    Description: App client for OAuth2 access
13   DomainPrefix:
14     Type: String
15     Default: catim
16     Description: Unique prefix for Cognito hosted UI domain
17   CatimUiDomain:
18     Type: String
19     Default: catim-ui.catims.intellecteu.io
20     Description: The CATIM UI Dns
21
22 Resources:
23   CognitoTokenCustomizerLambda:
24     Type: AWS::Lambda::Function
25     Properties:
26       FunctionName: CognitoTokenCustomizer
27       Handler: index.handler
28       Runtime: nodejs18.x
29       Role: !GetAtt CognitoLambdaExecutionRole.Arn
30       Code:
31         ZipFile: |
32           exports.handler = function(event, context) {
33
34             const groups = event.request.groupConfiguration.groupsToOverride || [];
35
36             // Group → scopes mapping
37             const groupScopes = {
38               "Admin": [
39                 "Message:Repair:SendToQueue",
40                 "Settings:MessageApproval:Update",
41                 "Settings:MessageApproval:View",
42                 "Settings:View",
43                 "Message:Routing:View",
44                 "Message:Metadata:View",
45                 "Message:Routing:Export",
46                 "Message:Routing:Import",
47                 "Message:Template:View",
48                 "Message:Repair:Edit",
49                 "Message:Details:View",
50                 "Message:Dashboard:Export",
51                 "Message:Body:View",
```

```

52         "Message:Intervention:View",
53         "Message:Validation:Allow",
54         "Message:Validation:Block",
55         "Message:Validation:Details:View",
56         "Message:Export",
57         "Message:Create",
58         "Message:Create:Dashboard:View",
59         "Message:Create:SendToQueue",
60         "Message:Create:Route",
61         "Message:Template:Create",
62         "Message:Draft:Create",
63         "Message:Template:Update",
64         "Message:Template:Delete",
65         "Message:Draft:Update",
66         "Message:Draft:View",
67         "Message:Draft:Delete",
68         "Message:Validation:Approve:1",
69         "Message:Validation:Reject:1",
70         "Message:Validation:Approve:2",
71         "Message:Validation:Reject:2",
72         "Message:Validation:Approve:3",
73         "Message:Validation:Reject:3",
74         "Message:Repair:Route",
75         "Message:Repair:RepairStatus"
76     ],
77     "Viewer": [
78         "Message:Details:View",
79         "Message:Body:View",
80         "Message:Intervention:View",
81         "Message:Validation:Details:View",
82         "Message:Create:Dashboard:View",
83         "Message:Template:View"
84     ],
85     "Operator": [
86         "Message:Template:View",
87         "Message:Details:View",
88         "Message:Dashboard:Export",
89         "Message:Body:View",
90         "Message:Intervention:View",
91         "Message:Export",
92         "Message:Create",
93         "Message:Create:Dashboard:View",
94         "Message:Create:Route",
95         "Message:Template:Create",
96         "Message:Draft:Create",
97         "Message:Template:Update",
98         "Message:Template:Delete",
99         "Message:Draft:Update",
100        "Message:Draft:View",
101        "Message:Draft:Delete",
102        "Message:Repair:Route",
103        "Message:Repair:RepairStatus",
104        "Message:Repair:SendToQueue"
105    ],
106    "L1": [
107        "Message:Template:View",
108        "Message:Repair:Edit",
109        "Message:Details:View",

```

```
110         "Message:Dashboard:Export",
111         "Message:Body:View",
112         "Message:Intervention:View",
113         "Message:Export",
114         "Message:Create",
115         "Message:Create:Dashboard:View",
116         "Message:Create:Route",
117         "Message:Template:Create",
118         "Message:Draft:Create",
119         "Message:Template:Update",
120         "Message:Template:Delete",
121         "Message:Draft:Update",
122         "Message:Draft:View",
123         "Message:Draft:Delete",
124         "Message:Repair:Route",
125         "Message:Repair:RepairStatus",
126         "Message:Repair:SendToQueue",
127         "Message:Validation:Approve:1",
128         "Message:Validation:Reject:1",
129         "Message:Create:SendToQueue",
130         "Message:Validation:Allow",
131         "Message:Validation:Block",
132         "Message:Validation:Details:View",
133         "Message:Routing:View",
134         "Message:Routing:Export",
135         "Message:Routing:Import"
136     ],
137     "L2": [
138         "Message:Template:View",
139         "Message:Details:View",
140         "Message:Dashboard:Export",
141         "Message:Body:View",
142         "Message:Intervention:View",
143         "Message:Export",
144         "Message:Create",
145         "Message:Create:Dashboard:View",
146         "Message:Create:Route",
147         "Message:Template:Create",
148         "Message:Draft:Create",
149         "Message:Template:Update",
150         "Message:Template:Delete",
151         "Message:Draft:Update",
152         "Message:Draft:View",
153         "Message:Draft:Delete",
154         "Message:Validation:Approve:2",
155         "Message:Validation:Reject:2",
156         "Message:Create:SendToQueue",
157         "Message:Validation:Allow",
158         "Message:Validation:Block",
159         "Message:Validation:Details:View",
160         "Message:Routing:View"
161     ],
162     "L3": [
163
164         "Message:Template:View",
165         "Message:Details:View",
166         "Message:Dashboard:Export",
167         "Message:Body:View",
```

```

168         "Message:Intervention:View",
169         "Message:Export",
170         "Message:Create",
171         "Message:Create:Dashboard:View",
172         "Message:Create:Route",
173         "Message:Template:Create",
174         "Message:Draft:Create",
175         "Message:Template:Update",
176         "Message:Template:Delete",
177         "Message:Draft:Update",
178         "Message:Draft:View",
179         "Message:Draft:Delete",
180         "Message:Validation:Approve:3",
181         "Message:Validation:Reject:3",
182         "Message:Create:SendToQueue",
183         "Message:Validation:Allow",
184         "Message:Validation:Block",
185         "Message:Validation:Details:View",
186         "Message:Routing:View"
187     ]
188 };
189
190 // Collect scopes for all groups the user belongs to
191 let scopesToAdd = [];
192 groups.forEach(group => {
193     if (groupScopes[group]) {
194         scopesToAdd = scopesToAdd.concat(groupScopes[group]);
195     }
196 });
197
198 // remove duplicates
199 scopesToAdd = [...new Set(scopesToAdd)];
200
201 event.response = {
202     "claimsAndScopeOverrideDetails": {
203         "accessTokenGeneration": {
204             "scopesToAdd": scopesToAdd,
205             "scopesToSuppress": [
206                 "phone_number",
207                 "aws.cognito.signin.user.admin"
208             ],
209             "claimsToAddOrOverride": {
210                 "preferred_username": event.request.userAttributes.email
211             }
212         }
213     }
214 };
215 // Return to Amazon Cognito
216 context.done(null, event);
217 };
218
219 CognitoLambdaExecutionRole:
220   Type: AWS::IAM::Role
221   Properties:
222     AssumeRolePolicyDocument:
223       Version: "2012-10-17"
224       Statement:
225         - Effect: Allow

```

```
226         Principal:
227             Service: lambda.amazonaws.com
228             Action: sts:AssumeRole
229     Policies:
230         - PolicyName: CognitoLambdaExecutionPolicy
231           PolicyDocument:
232             Version: "2012-10-17"
233             Statement:
234                 - Effect: Allow
235                   Action:
236                       - logs:CreateLogGroup
237                       - logs:CreateLogStream
238                       - logs:PutLogEvents
239                   Resource: "*"
240
241     CognitoUserPool:
242         Type: AWS::Cognito::UserPool
243         Properties:
244             UserPoolName: !Ref UserPoolName
245             AutoVerifiedAttributes:
246                 - email
247             LambdaConfig:
248                 PreTokenGeneration: !GetAtt CognitoTokenCustomizerLambda.Arn
249                 PreTokenGenerationConfig:
250                     LambdaArn: !GetAtt CognitoTokenCustomizerLambda.Arn
251                     LambdaVersion: V3_0
252             Schema:
253                 - Name: email
254                   AttributeDataType: String
255                   Required: true
256                   Mutable: true
257             Policies:
258                 PasswordPolicy:
259                     MinimumLength: 8
260                     RequireUppercase: true
261                     RequireLowercase: true
262                     RequireNumbers: true
263                     RequireSymbols: true
264
265     CognitoUserPoolClient:
266         Type: AWS::Cognito::UserPoolClient
267         Properties:
268             ClientName: !Ref AppClientName
269             UserPoolId: !Ref CognitoUserPool
270             GenerateSecret: false
271             AllowedOAuthFlows:
272                 - code
273                 - implicit
274             AllowedOAuthScopes:
275                 - openid
276                 - email
277                 - profile
278             AllowedOAuthFlowsUserPoolClient: true
279             CallbackURLs:
280                 - !Sub "https://${CatimUiDomain}/"
281             LogoutURLs:
282                 - !Sub "https://${CatimUiDomain}/logout"
283             SupportedIdentityProviders:
```



```
284     - COGNITO
285
286 CognitoGroupAdmin:
287   Type: AWS::Cognito::UserPoolGroup
288   Properties:
289     GroupName: Admin
290     UserPoolId: !Ref CognitoUserPool
291 CognitoGroupViewer:
292   Type: AWS::Cognito::UserPoolGroup
293   Properties:
294     GroupName: Viewer
295     UserPoolId: !Ref CognitoUserPool
296 CognitoGroupOperator:
297   Type: AWS::Cognito::UserPoolGroup
298   Properties:
299     GroupName: Operator
300     UserPoolId: !Ref CognitoUserPool
301 CognitoGroupL1Compliance:
302   Type: AWS::Cognito::UserPoolGroup
303   Properties:
304     GroupName: L1
305     UserPoolId: !Ref CognitoUserPool
306 CognitoGroupL2Compliance:
307   Type: AWS::Cognito::UserPoolGroup
308   Properties:
309     GroupName: L2
310     UserPoolId: !Ref CognitoUserPool
311 CognitoGroupL3Compliance:
312   Type: AWS::Cognito::UserPoolGroup
313   Properties:
314     GroupName: L3
315     UserPoolId: !Ref CognitoUserPool
316
317 CognitoUserPoolDomain:
318   Type: AWS::Cognito::UserPoolDomain
319   Properties:
320     Domain: !Ref DomainPrefix
321     UserPoolId: !Ref CognitoUserPool
322
323 CognitoResourceServer:
324   Type: AWS::Cognito::UserPoolResourceServer
325   Properties:
326     Identifier: message-api
327     Name: "Message API Resource Server"
328     UserPoolId: !Ref CognitoUserPool
329     Scopes:
330       - ScopeName: Message:Repair:SendToQueue
331         ScopeDescription: Grants access to the Send to button when repairing a message,
allowing users to select a queue.
332       - ScopeName: Message:Repair:AnyStatus
333         ScopeDescription: Grants access to the repair button on message details on any
status.
334       - ScopeName: Settings:MessageApproval:Update
335         ScopeDescription: Grants access to edit/change all configs/toggles in the Message
Approval tab, inside Settings.
336       - ScopeName: Settings:MessageApproval:View
337         ScopeDescription: Grants access to consult all configs/toggles in the Message
Approval tab, inside Settings.
```

```
338     - ScopeName: Settings:View
339       ScopeDescription: Grants access to the settings dashboard.
340     - ScopeName: Message:Routing:View
341       ScopeDescription: Grants access to the Routing dashboard.
342     - ScopeName: Message:Metadata:View
343       ScopeDescription: Grants access to view the message metadata.
344     - ScopeName: Message:Routing:Export
345       ScopeDescription: Grants access to the Export button on the Routing dashboard.
346     - ScopeName: Message:Routing:Import
347       ScopeDescription: Grants access to the Import button on the Routing dashboard.
348     - ScopeName: Message:Template:View
349       ScopeDescription: Grants access to view available PUBLIC templates in the Message
350 templates dashboard, and their details.
351     - ScopeName: Message:Repair:Edit
352       ScopeDescription: Grants access to edit the Application Header and Body on
353 message repair.
354     - ScopeName: Message:Details:View
355       ScopeDescription: Grants access to view the message dashboard and details (the
356 first part on the message details tab).
357     - ScopeName: Message:Dashboard:Export
358       ScopeDescription: Allows access to the export button on the messages dashboard.
359     - ScopeName: Message:Body:View
360       ScopeDescription: Grants access to view the message body (the second part is on
361 the message details tab).
362     - ScopeName: Message:Intervention:View
363       ScopeDescription: Grants access to the interventions tab.
364     - ScopeName: Message:Validation:Allow
365       ScopeDescription: Grants access to the Allow button on the message validations
366 tab.
367     - ScopeName: Message:Validation:Block
368       ScopeDescription: Grants access to the Block button on the message validations
369 tab.
370     - ScopeName: Message:Validation:Details:View
371       ScopeDescription: Grants access to view validations dashboard and details (Info).
372     - ScopeName: Message:Export
373       ScopeDescription: Allows access to the export button on the message details for
374 the body and/or details.
375     - ScopeName: Message:Create
376       ScopeDescription: Grants access to the Select button on the New FIN/MX message
377 dashboard for creating a message.
378     - ScopeName: Message:Create:Dashboard:View
379       ScopeDescription: Grants access to visualize the New FIN/MX message dashboard.
380     - ScopeName: Message:Create:SendToQueue
381       ScopeDescription: Grants access to the Send to button, allowing users to select a
382 queue and send messages for processing.
383     - ScopeName: Message:Create:Route
384       ScopeDescription: Grants access to the Route button, routing the message through
385 a pre-defined workflow.
386     - ScopeName: Message:Template:Create
387       ScopeDescription: Grants access to the Template button on message creation for
388 saving the message as a template.
389     - ScopeName: Message:Draft:Create
390       ScopeDescription: Grants access to the Draft button on message creation for
391 saving the message as a draft.
392     - ScopeName: Message:Template:Update
393       ScopeDescription: Grants access to the Pen button on the Message templates
394 dashboard to update a template.
395     - ScopeName: Message:Template>Delete
```

```

383     ScopeDescription: Grants access to the Trash button on the Message templates
dashboard to delete a template.
384     - ScopeName: Message:Draft:Update
385     ScopeDescription: Grants access to the Pen button on the Message drafts dashboard
to update a draft.
386     - ScopeName: Message:Draft:View
387     ScopeDescription: Grants access to view available drafts in the Message drafts
dashboard.
388     - ScopeName: Message:Draft:Delete
389     ScopeDescription: Grants access to the Trash button on the Message drafts
dashboard to delete a draft.
390     - ScopeName: Message:Validation:Approve:1
391     ScopeDescription: Provides access to the Approve button on the message
validations tab, allowing decisions to be made at the First level.
392     - ScopeName: Message:Validation:Reject:1
393     ScopeDescription: Provides access to the Reject button on the message validations
tab, allowing decisions to be made at the First level.
394     - ScopeName: Message:Validation:Approve:2
395     ScopeDescription: Provides access to the Approve button on the message
validations tab, allowing decisions to be made at the Second level.
396     - ScopeName: Message:Validation:Reject:2
397     ScopeDescription: Provides access to the Reject button on the message validations
tab, allowing decisions to be made at the Second level.
398     - ScopeName: Message:Validation:Approve:3
399     ScopeDescription: Provides access to the Approve button on the message
validations tab, allowing decisions to be made at the Third level.
400     - ScopeName: Message:Validation:Reject:3
401     ScopeDescription: Provides access to the Reject button on the message validations
tab, allowing decisions to be made at the Third level.
402     - ScopeName: Message:Repair:Route
403     ScopeDescription: Grants access to the Route button when repairing a message,
routing the message through a pre-defined workflow.
404     - ScopeName: Message:Repair:Metadata:Update
405     ScopeDescription: Grants access to edit the message metadata during the repair.
406     - ScopeName: Message:Repair:RepairStatus
407     ScopeDescription: Grants access to the repair button on message details when
messages are on the Repair status.
408
409     LambdaInvokePermission:
410         Type: AWS::Lambda::Permission
411         Properties:
412             Action: lambda:InvokeFunction
413             FunctionName: !GetAtt CognitoTokenCustomizerLambda.Arn
414             Principal: cognito-idp.amazonaws.com
415             SourceArn: !Sub arn:${AWS::Partition}:cognito-
idp:${AWS::Region}:${AWS::AccountId}:userpool/${CognitoUserPool}
416
417     Outputs:
418         UserPoolId:
419             Description: Cognito User Pool ID
420             Value: !Ref CognitoUserPool
421
422         UserPoolClientId:
423             Description: App Client ID
424             Value: !Ref CognitoUserPoolClient
425
426         UserPoolDomain:
427             Description: Cognito Hosted UI domain URL

```

```

428     Value: !Sub "https://${DomainPrefix}.auth.${AWS::Region}.amazoncognito.com"
429
430     ResourceServerIdentifier:
431         Description: Resource Server Identifier
432         Value: !Ref CognitoResourceServer

```

3.2 Apply Cloudformation stack

With the use of AWS CLI, deploy the Cloudformation stack, ensuring you replace placeholders with relevant values:

 Replace placeholder for CatimUiDomain, ensuring it is the URL you will associated with the CATIM Frontend

```

1 aws cloudformation create-stack \
2   --stack-name CognitoStack \
3   --template-body file://cognito-stack.yaml \
4   --capabilities CAPABILITY_NAMED_IAM CAPABILITY_AUTO_EXPAND \
5   --parameters \
6     ParameterKey=CatimUiDomain,ParameterValue={REPLACEME}

```

3.3 Create Cognito user

1. Login to the AWS console and navigate to the Cognito User Pool you just created
2. Under *User Management > Users*, click *Create User*
3. Set a password (you will be asked to change it upon your first login)
4. After user creation, click on the user you created
5. Edit *User attributes* and add the *Optional attribute preferred_username* set to the user email

3.4 Add Cognito user to Admin group

1. In your Cognito user pool, navigate to *User Management > Groups > Admin*
2. In *Group Members*, add the new user you just created

4. Install & Configure Temporal

1. Add the temporalio Helm repo:

```
1 helm repo add temporalio https://temporalio.github.io/helm-charts
```

2. Add variables to your bash session, replacing placeholders:

```

1 WEB_URL=temporal.{REPLACEME} # the Ingress for your Temporal-web service
2 DB_HOST={REPLACEME} # The hostname for your Postgres RDS
3 DB_USERNAME={REPLACEME} # The RDS username
4 DB_PASSWORD={REPLACEME} # The RDS user password

```

3. Apply the Temporal helm values:

```
1 helm install -n catim --create-namespace temporal temporalio/temporal --version 0.62.0 -f
2 - <<EOF
3 web:
4   image:
5     repository: 709825985650.dkr.ecr.us-east-1.amazonaws.com/intellecteu/catim
6     tag: temporalio-ui-2.37.1
7     pullPolicy: IfNotPresent
8   ingress:
9     enabled: true
10    hosts:
11      - $WEB_URL
12  additionalEnv:
13    - name: TEMPORAL_CSRF_COOKIE_INSECURE
14      value: "True"
15 server:
16   image:
17     repository: 709825985650.dkr.ecr.us-east-1.amazonaws.com/intellecteu/catim
18     tag: temporalio-server-1.27.2
19     pullPolicy: IfNotPresent
20  internalFrontend:
21    enabled: true
22  config:
23    namespaces:
24      create: true
25      namespace:
26        - name: default
27          retention: 3d
28  persistence:
29    default:
30      driver: sql
31      sql:
32        driver: postgres12
33        host: $DB_HOST
34        port: 5432
35        database: temporal
36        user: $DB_USERNAME
37        password: $DB_PASSWORD
38        maxConns: 20
39        maxConnLifetime: 1h
40        tls:
41          enabled: true
42          enableHostVerification: false
43  visibility:
44    driver: sql
45    sql:
46      driver: postgres12
47      host: $DB_HOST
48      port: 5432
49      database: temporal_visibility
50      user: $DB_USERNAME
51      password: $DB_PASSWORD
52      maxConns: 20
53      maxConnLifetime: 1h
54      tls:
55        enabled: true
56        enableHostVerification: false
57  admintools:
58    image:
```

```

58     repository: 709825985650.dkr.ecr.us-east-1.amazonaws.com/intellecteu/catim
59     tag: temporalio-admintools-1.28.1-tctl-1.18.4-cli-1.4.1
60     pullPolicy: IfNotPresent
61 cassandra:
62     enabled: false
63 mysql:
64     enabled: false
65 postgresql:
66     enabled: false
67 prometheus:
68     enabled: false
69 grafana:
70     enabled: false
71 elasticsearch:
72     enabled: false
73 EOF

```

5. Install & Configure CAT-IM

1. Add variables to your bash session, replacing placeholders:

```

1 DOMAIN={REPLACEME} # the base custom domain/subdomain for your EKS cluster
2 DB_HOSTNAME={REPLACEME} # the RDS Postgresql DNS hostname
3 DB_USERNAME={REPLACEME} # The RDS Postgresql username
4 DB_PASSWORD={REPLACEME} # The RDS Postgresql password
5 OIDC_CLIENTID={REPLACEME} # The Cognito App Client ID
6 OIDC_AUTHORITY_URL=https://cognito-idp.
  {REPLACEME_AWSREGION}.amazonaws.com/{REPLACEME_USERPOOLID}
7 OIDC_AUTH_URL=https://{REPLACEME_COGNITODOMAINNAME}.auth.
  {REPLACEME_AWSREGION}.amazoncognito.com/oauth2/authorize
8 SFTP_HOSTNAME={REPLACEME}
9 SFTP_USERNAME={REPLACEME}
10 SFTP_PASSWORD={REPLACEME}
11 INGRESS_CLASS={REPLACEME}

```

2. Login to ECR registry:

```

1 aws ecr get-login-password \
2     --region us-east-1 | helm registry login \
3     --username AWS \
4     --password-stdin 709825985650.dkr.ecr.us-east-1.amazonaws.com

```

3. Apply the CAT-IM helm values:

```

1 helm upgrade --install catim oci://709825985650.dkr.ecr.us-east-
  1.amazonaws.com/intellecteu/catim --namespace catim --version 4.0.12 \
2     --set global.baseHostname=$DOMAIN \
3     --set global.postgresql.auth.hostname=$DB_HOSTNAME \
4     --set global.postgresql.auth.username=$DB_USERNAME \
5     --set global.postgresql.auth.password=$DB_PASSWORD \
6     --set global.oidc.clientId=$OIDC_CLIENTID \
7     --set global.oidc.authorityUrl=$OIDC_AUTHORITY_URL \
8     --set global.oidc.authUrl=$OIDC_AUTH_URL \
9     --set global.sftp.hostname=$SFTP_HOSTNAME \
10    --set global.sftp.username=$SFTP_USERNAME \
11    --set global.sftp.password=$SFTP_PASSWORD \

```

```
12 --set global.ingressClassName=$INGRESS_CLASS
```