



Datomic Cloud Documentation

- [Home](#) >
- >
- Troubleshooting
- [Support](#)
- [Forum](#)

## What is Datomic?

- [Data Model](#)
- [Architecture](#)
- [Supported Operations](#)
- [Programming with Data and EDN](#)

## Local Dev and CI

### Cloud Setup

- [Start a System](#)
- [Configure Access](#)
- [Get Connected](#)

### Tutorial

- [Client API](#)
- [Assertion](#)
- [Read](#)
- [Accumulate](#)
- [Read Revisited](#)
- [Retract](#)
- [History](#)

### Client API

### Videos

- [AWS Setup](#)
- [Edn](#)
- [Datalog](#)
- [Datoms](#)
- [HTTP Direct](#)
- [CLI tools](#)

### Schema

- [Defining Attributes](#)
- [Schema Reference](#)
- [Changing Schema](#)
- [Data Modeling](#)
- [Schema Limits](#)

### Transactions

- [Processing Transactions](#)
- [Transaction Data Reference](#)
- [Transaction Functions](#)
- [ACID](#)
- [Client Synchronization](#)

### Query and Pull

- [Executing Queries](#)
- [Query Data Reference](#)
- [Pull](#)
- [Index-pull](#)
- [Raw Index Access](#)

### Time in Datomic

- [Log API](#)
- [Time Filters](#)

## [Ions](#)

- [Ions Tutorial](#)
- [Ions Reference](#)
- [Monitoring Ions](#)

## [Analytics Support](#)

- [Configuration](#)
- [Connecting](#)
- [Metaschema](#)
- [SQL CLI](#)
- [Troubleshooting](#)

## [Analytics Tools](#)

- [Metabase](#)
- [R](#)
- [Python](#)
- [Jupyter](#)
- [Superset](#)
- [JDBC](#)
- [Other Tools](#)

## [Operation](#)

- [Planning Your System](#)
- [Start a System](#)
- [AWS Account Setup](#)
- [Access Control](#)
- [CLI Tools](#)
- [Client Applications](#)
- [High Availability \(HA\)](#)
- [Howto](#)
- [Query Groups](#)
- [Monitoring](#)
- [Upgrading](#)
- [Scaling](#)
- [Deleting](#)
- [Splitting Stacks](#)

## [Tech Notes](#)

- [Turning Off Unused Resources](#)
- [Reserved Instances](#)
- [Lambda Provisioned Concurrency](#)

## [Best Practices](#)

## [Troubleshooting](#)

## [FAQ](#)

## [Examples](#)

## [Releases](#)

## [Glossary](#)

Hide All Examples

# Troubleshooting

- [Troubleshooting CloudFormation Templates](#)
- [Troubleshooting Compute Nodes](#)
- [Troubleshooting Solo Stacks](#)
- [Troubleshooting Clients](#)
- [Troubleshooting Socks Proxy](#)
- [Troubleshooting Ions](#)

Not finding what you need here? Ask on the [forum](#).

**IMPORTANT** - Datomic Cloud is configured with a set of well-tested default parameters for scaling and provisioning. You should **NOT** alter system settings (i.e. [storage provisioning and scaling](#), changes to network or instance configurations, alterations to the default CloudFormation templates) without first contacting [Datomic Support](#)

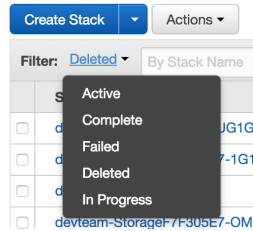
unless the changes are explicitly described in the Datomic Documentation.

## Troubleshooting CloudFormation Templates

### Failures in Nested Templates

If a nested template fails, the root template will fail also, and the console will by default show only the root template's `CREATE_FAILED` event. To see the important details in the nested template's events:

- Click on the Filter popup, and choose "Deleted"



- Select the nested template, and then choose the Events tab to see the cause of failure.

### Check for CloudFormation failure

To check your CloudFormation for errors, find your stack in the [CloudFormation](#)

window and click the checkbox at the start of its row.

If the status of the stack is "CREATE\_IN\_PROGRESS" the stack is still being created, and you should continue to wait for the stack to start. You can monitor it from the Cloudformation details window. The stack is done when it says "CREATE\_COMPLETE."

If the status of the stack is "CREATE\_FAILED" there was an error starting the stack. See the other topics on this page for help tracking down the source of the failure.

### Production Topology Create Failed

If the production topology fails to create in CloudFormation with a `CREATE_FAILED` on the `AWS::AutoScaling::AutoScalingGroup` for the `TxAutoScalingGroup` resource with the message:

You have requested more instances (2) than your current instance limit of 1 allows for the specified instance type. Please visit <http://>

You need to [request a limit increase](#) in your allowed number of running `i3.large` instances.

### Starting a compute stack without a storage stack.

Without a storage system, compute will fail immediately and rollback. The stack event message will report the following error, substitute `<system-name>` for the system name you specified when launching your compute stack.

No export named `<system-name>-FileSystemId` found. Rollback requested by user.

The screenshot shows the AWS CloudFormation console. At the top, there are navigation tabs for Services, Resource Groups, CloudFormation, and EC2. The CloudFormation tab is active, and the 'Stacks' section is selected. A filter is applied to 'test-compute'. The stack list shows one stack named 'test-compute' with a status of 'ROLLBACK\_COMPLETE'. Below the stack list, the 'Events' tab is selected, showing a list of events for the stack. The events table has columns for Time, Status, Type, Logical ID, and Status reason. The events are as follows:

Time	Status	Type	Logical ID	Status reason
11:43:39 UTC-0400	ROLLBACK_COMPLETE	AWS::CloudFormation::Stack	test-compute	
11:43:29 UTC-0400	ROLLBACK_IN_PROGRESS	AWS::CloudFormation::Stack	test-compute	No export named test-FileSystemServiceId found. Rollback requested by user.
11:43:27 UTC-0400	CREATE_IN_PROGRESS	AWS::CloudFormation::Stack	test-compute	User Initiated

## Running in EC2 Classic

Datomic Cloud is currently only supported in accounts that support only EC2-VPC. If you attempt to run Datomic Cloud in an EC2-Classic, the Cloudformation stack will fail. The stack event message will have an error with a Type of "Custom::ResourceCheck" and a Logical ID of "EnsureEc2Vpc."

See [Setting Up](#) for more information.

## Update Failed with "Export cannot be deleted"

A failed compute stack update to or past version 470-8654 with the error:

```
Export <SystemName>-VpcEndpointServiceName cannot be deleted as it is in use by <systemname>-vpc-endpoint
```

Indicates that you have previously launched a VPC Endpoint using the provided Endpoint CloudFormation template. Versions from 470-8654 on do not create the necessary services for the VPC Endpoint template. You will need to delete the VPC Endpoint stack from the [CloudFormation dashboard](#) and [recreate the endpoint](#).

## Troubleshooting Compute Nodes [↗](#)

### Check Alerts First

Datomic publishes a [metric](#) named `Alert` for any situation that might require operator intervention. You can create a Cloudwatch Alarm for the `Alert` metric, and then [search the logs](#) for specific details.

Reviewing alerts should be your first step when you are troubleshooting Datomic Cloud nodes.

### Check Memory Second

Running out of memory is the number one cause of both availability and performance problems.

Every Datomic node publishes a `JvmFreeMb` [metric](#). If the Minimum value for this metric dips below 15% of its initial startup value for an extended period, memory pressure is likely a problem.

- If you believe the problem is individual large queries, upgrade to larger EC2 instance sizes.
- If you believe the problem is a high volume of smaller queries, increase the number of instances in your compute group.

## Troubleshooting Solo Nodes

### Check Alerts and Logs First

As described [above](#), Datomic publishes a [metric](#) named `Alert` for any situation that might require operator intervention. You can create a Cloudwatch Alarm for the `Alert` metric, and then [search the logs](#) for specific details.

When troubleshooting a Solo system, you should ensure that the single compute node in your system is healthy and reporting [logs](#) and [metrics](#). If it is not, you should use the [EC2 Dashboard](#) to terminate your compute node and allow the AutoScaling group to replace it.

If you require an HA system with built-in monitoring and failover, you should move to the [Production Topology](#).

## Troubleshooting Client Errors

## Missing Connect Arguments

If you leave out a required argument.

```
CompilerException clojure.lang.ExceptionInfo: Expected string for :query-group {:cognitect.anomalies/category :cognitect.anomalies/inco
```

### ::cognitect.anomalies/forbidden

If you get a *forbidden* error when using the Client API, this means that your AWS credentials do not grant permission to this functionality.

- Make sure that you have an IAM policy that [authorizes](#) the operation you are trying to perform.
- Make sure that policy is associated with the identity you are running under.

### ExceptionInfo Forbidden to read keyfile

```
>d/create-database client {:db-name "test"})
ExceptionInfo Forbidden to read keyfile at s3://datomic-test-storageef7f305e7-1bpzuyaf5d-s3datomic-1wgl6uvtl9bei/datomic-test/datomic/ac
```

Ensure that you have sourced the right credentials with all necessary permissions.

## Connection Failure

The client error:

```
Exception in thread "main" clojure.lang.ExceptionInfo: Unable to connect to system:
{:cognitect.anomalies/category :cognitect.anomalies/fault, :cognitect.anomalies/message "SOCKS4 tunnel failed, connection closed", :cognit
:cause "SOCKS4 tunnel failed, connection closed"
:via
 [{:type java.io.IOException
  :message "SOCKS4 tunnel failed, connection closed"
  :at [org.eclipse.jetty.client.Socks4Proxy$Socks4ProxyConnection onFillable "Socks4Proxy.java" 166]}] ...
```



Accompanied by the following message from your SOCKS Proxy:

```
debug1: channel 2: new [dynamic-tcpip]
channel 2: open failed: administratively prohibited: open failed
```

Indicates that the client was unable to reach the Datomic system through the proxy. Check your configuration `:endpoint` carefully and use the [[[getting-started/connecting.html](#)][Access Gateway Connection Test]] to ensure your proxy is configured correctly.

### java.lang.NoClassDefFoundError: javax/xml/bind/DatatypeConverter

This issue is resolved on [client-cloud 0.8.63](#)

On `com.datomic/client-cloud 0.8.56` and Java 9+ you'll run into this error when connecting:

```
CompilerException clojure.lang.ExceptionInfo: java.lang.NoClassDefFoundError: javax/xml/bind/DatatypeConverter #:cognitect.anomalies{:c
```

As of Java 9 the `jaxb` module was removed from the default JDK. As such, you must explicitly require the `javax.xml.bind/jaxb-api` `{:mvn/version "2.3.0"}` dep to resolve this error.

## Jetty Dependency Conflict

The following error will occur if your project has a transient dependency on a more recent version of Apache Jetty than the version used by Datomic:

```
#error {
:cause "org.eclipse.jetty.util.thread.Invocable$InvocationType"
:via
 [{:type java.lang.NoClassDefFoundError
  :message "org/eclipse/jetty/util/thread/Invocable$InvocationType"
  :at [org.eclipse.jetty.io.ManagedSelector <init> "ManagedSelector.java" 79]}]
{:type java.lang.ClassNotFoundException
 :message "org.eclipse.jetty.util.thread.Invocable$InvocationType"
 :at [java.net.URLClassLoader findClass "URLClassLoader.java" 381]}]
...}
```



The conflict can be resolved by excluding the version of Jetty used by Datomic:

```
[com.datomic/client-cloud "0.8.54"
 :exclusions [org.eclipse.jetty/jetty-client
```

```
org.eclipse.jetty/jetty-http
org.eclipse.jetty/jetty-util]]
```



## Loading database error when connecting

The "Loading database" message:

```
#:cognitect.anomalies{:category :cognitect.anomalies/unavailable,
                      :message "Loading database"}
```

indicates that the Datomic node you've reached has not yet loaded the requested database into memory. This error can occur when you call `create-database`, followed immediately by `connect`, as well as following an instance restart. You should consider this a retry-able error and wrap the connection attempt with retry logic.

### ::cognitect.anomalies/busy

If you get a *busy* response from the Client API, your request rate has temporarily exceeded the capacity of a node, and has already been through an exponential backoff and retry implemented by the client. At this point you have three options:

- When transacting continue to retry the request at the application level with you own exponential backoff. The [Mbrainz Importer](#) example project demonstrates a [batch import with retry](#).
- When querying expand the capacity of your system, by [upgrading](#) from [Solo](#) to [Production](#) or increasing the number of instances in your AutoScaling Group.
- Give up on completing the request.

## Server type mismatch in datomic client

Ions support the *synchronous* Datomic Client API

The error:

```
{
  "Msg": "IonLambdaException",
  "Ex": {
    "Cause": ":server-type must be :cloud or :peer-server",
    "Via": [
      {
        "Type": "clojure.lang.ExceptionInfo",
        "Message": ":server-type must be :cloud or :peer-server",
        "Data": {
          "CognitectAnomaliesCategory": "CognitectAnomaliesIncorrect",
          "CognitectAnomaliesMessage": ":server-type must be :cloud or :peer-server"
        },
        "At": [
          "clojure.core$ex_info",
          "invokeStatic",
          "core.clj",
          4739
        ]
      }
    ]
  }
}
```



Indicates an attempt to use the async API with the server-type `:ion`

You should use the synchronous Client API in your Ion(s).

## Troubleshooting Socks Proxy Errors

### Unsupported AWS CLI Version

**Information for previous CLI tools** - Please update to [the latest version](#).

Symptom: The script prints a generic AWS CLI help message before failing:

```
$ bash datomic-access client -r eu-central-1 good-system
To see help text, you can run:

aws help
aws <command> help
aws <command> <subcommand> help
aws: error: argument command: Invalid choice, valid choices are:
...
```

**Fix:** Update to the latest version of the [AWS CLI](#).

### Wrong AWS Creds

**Information for previous CLI tools** - Please update to [the latest version](#).

```
$ bash datomic-access client -r eu-central-1 good-system
Unable to read bastion key, make sure your AWS creds are correct.
```

**Fix:** Make sure your AWS creds are connected to a policy that is [authorized](#) for your Datomic system.

## Wrong System Name

**Information for previous CLI tools** - Please update to [the latest version](#).

```
$ bash datomic-access client -r eu-central-1 not-a-system
Datomic system not-a-system not found, make sure your system name and AWS creds are correct.
```

## AccessDeniedException

```
An error occurred (AccessDeniedException) when calling the GetResources operation: User: arn:aws:iam::<number>:user/<user-name> is not
Datomic system <system-name> not found, make sure your system name and AWS creds are correct.
```

**Fix:** Ensure that you [configured the correct AWS credentials](#). This error indicates that the configured AWS credentials do not match what is necessary to access the indicated system. You may be using the wrong credentials for the account.

## SOCKS4 tunnel failed, connection closed and connection refused errors in Client

**Information for previous CLI tools** - Please update to [the latest version](#).

This error will throw when the connection is unable to find a Datomic Socks proxy, if you're seeing this error while running remotely it is because the Client is expecting a proxy and you'll need to remove the proxy from your config map:

```
CompilerException clojure.lang.ExceptionInfo: Unable to connect to localhost:8182 {:cognitect.anomalies/category :cognitect.anomalies/fault}
:cause "SOCKS4 tunnel failed, connection closed"
:via
[{:type java.io.IOException
:message "SOCKS4 tunnel failed, connection closed"
:at [org.eclipse.jetty.client.Socks4Proxy$Socks4ProxyConnection onFillable "Socks4Proxy.java" 166]}]
:trace
[[org.eclipse.jetty.client.Socks4Proxy$Socks4ProxyConnection onFillable "Socks4Proxy.java" 166]
[org.eclipse.jetty.io.AbstractConnection$ReadCallback succeeded "AbstractConnection.java" 273]
[org.eclipse.jetty.io.FillInterest fillable "FillInterest.java" 95]
[org.eclipse.jetty.io.SelectChannelEndPoint$2 run "SelectChannelEndPoint.java" 75]
[org.eclipse.jetty.util.thread.strategy.ExecuteProduceConsume produceAndRun "ExecuteProduceConsume.java" 213]
[org.eclipse.jetty.util.thread.strategy.ExecuteProduceConsume run "ExecuteProduceConsume.java" 147]
[org.eclipse.jetty.util.thread.QueuedThreadPool runJob "QueuedThreadPool.java" 654]
[org.eclipse.jetty.util.thread.QueuedThreadPool$3 run "QueuedThreadPool.java" 572]
[java.lang.Thread run "Thread.java" 745]]], :config {:server-type :cloud, :region "us-east-1", :system "wrong", :query-group "wrong", :}
```



This error will throw when the config map is pointing to the wrong proxy port.

```
CompilerException clojure.lang.ExceptionInfo: Unable to connect to localhost:8188 {:cognitect.anomalies/category :cognitect.anomalies/unavailable}
```

You will see this error when using the `datomic-access` if the connection or proxy process has failed.

With both of these errors, you'll want to double check your connection config.

## Error retrieving results

The `datomic` command is unable to access the Datomic system. Specify an [AWS Credentials](#) source using `-p` or `--profile`, and make sure the currently specified credentials are correct for the Datomic system that you are trying to access.

## Troubleshooting Ion push

### Ion push requires a clean git commit

If you attempt to [push](#) an ion project with uncommitted local changes, you will see the following exception:

```
java.lang.IllegalArgumentException: You must either specify a uname or deploy from clean git commit
```

You must either

- push from a fully clean git commit (recommended)
- provide an [unreproducible name](#)

### Ion push fails to find a code bucket

The ion push operation pushes your code to a region-specific Datomic code bucket. If push fails to find a code bucket, it will report a message like:

```
{:command-failed "{:op :push}",
:causes
({:message
  "Unable to find a unique code bucket. Make sure that you are running\nDatomic Cloud in the region you are deploying to.",
:class ExceptionInfo,
:data {"datomic:code" nil}})}
```



The most common cause of this is that you are accidentally using the wrong AWS keys, or your AWS credentials configuration is defaulting to the wrong AWS region. Make sure you are [using the right AWS access keys](#).

## Ion push requires ion-dev tools

The ion-dev tools are invoked via a deps.edn alias. If you have not installed the ion-dev tools, the Clojure CLI will report that the alias is undeclared when you try to push:

```
clojure -A:ion-dev '{:op :push}'
WARNING: Specified aliases are undeclared: [:ion-dev]
Execution error (FileNotFoundException) at java.io.FileInputStream/open0 (FileInputStream.java:-2).
{:op :push} (No such file or directory)
```

If you encounter this error, [install the ion-dev tools](#).

## Ion push requires a recent version of Clojure CLI

The ion-dev tools require a recent version of the Clojure CLI. If you see the following error:

```
Error building classpath. Unknown alias key: :deps
```

then you need a [more recent version](#) of the Clojure CLI.

## Troubleshooting Ion deploy

### Ion deploy fails with "Assert failed" message

Failure of [Ion Deployment](#) with the following error message in the [Datomic system CloudWatch Logs](#):

```
"Cause": "Assert failed: cfg",
"Via": [
  {
    "Type": "java.lang.AssertionError",
    "Message": "Assert failed: cfg",
    "At": [
      "datomic.client.impl.local$create_client",
      "invokeStatic",
      "local.clj",
      97
    ]
  }
]
```



Indicates that a namespace in the ion attempted to create a connection to a Datomic system as a side effect of loading a namespace. You should follow the "connect on use" pattern as shown in the [Ion Starter](#) example.

## Troubleshooting Ion deploy-status

If your ion-deploy succeeds, but never reaches SUCCEEDED status, it is likely that your application code cannot load correctly on your compute nodes. Follow the steps for [troubleshooting compute nodes](#).

## Troubleshooting Ions

### Supported Lambda Ion Return Types

Returning an unsupported type from a Lambda Ion will result in the exception:

```
datomic.ion.lambda.handler.exceptions.Incorrect: No implementation of method: :->bbuf of protocol: #'datomic.ion.lambda.dispatcher/ToBb
```

Lambda ions [should return](#) a String, InputStream, ByteBuffer, or File.

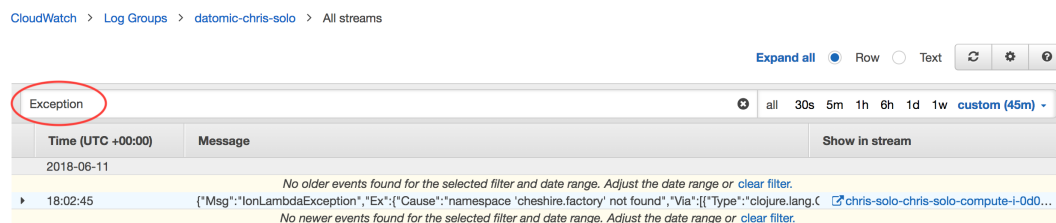
### API Gateway Returns a 500 Internal Server Error

A HTTP 500 Internal server error returned from API Gateway indicates that the ion code threw an exception. Find the full exception text and stack trace in the Datomic system CloudWatch Logs:



- Open the [CloudWatch Logs](#) in the AWS Console
- Click on the Log Group named "datomic-{System}", where System is your system name. Each EC2 instance will create a separate log stream. Usually you will want to search across all log streams.
- Click the "Search Log Group" button
- Enter "Exception" in the search dialog
- Find the event of interest in the search results and click the arrow to expand the details

The screenshot below shows finding an ion exception in the log:



Copyright © Cognitect, Inc  
Datomic® and the Datomic logo are registered trademarks of Cognitect, Inc