

# Table of Contents

Introduction and Overview	1.1
Server Setup and Installation	1.2
Architecture	1.2.1
Server Setup	1.2.2
Pre-requisites	1.2.2.1
Aurora	1.2.2.1.1
Amazon Elastic Container Repository	1.2.2.1.2
Athena	1.2.2.1.3
SQS Queue	1.2.2.1.4
SNS Topics	1.2.2.1.5
Secret for Configuration Information	1.2.2.1.6
Application Load Balancer with HTTPS	1.2.2.1.7
Docker on EC2 Instance	1.2.2.2
Docker on ECS with EC2 Capability	1.2.2.3
Docker on ECS with Fargate Capability	1.2.2.4
Deployment via CloudFormation	1.2.2.5
Manual Installation on EC2	1.2.2.6
Application Installation	1.2.3
Post Install Configuration	1.2.4
Features and Workflows	1.3
Deployment Options	1.3.1
Directory Structure	1.3.2
Dashboard	1.3.3
Notifications	1.3.4
Setup	1.4
Accounts	1.4.1
Users	1.4.2
Reset Data	1.4.3
User and Account Management	1.5
User Roles	1.5.1
Hierarchy of User Roles	1.5.2
User Access Restrictions	1.5.3
Login and Access Control	1.5.4
Account Management	1.5.5
User Management	1.5.6
Change User Password	1.5.7
Update User Profile	1.5.8

Command Logging	1.6
Hadoop	1.7
Create a new Hadoop Configuration	1.7.1
Start a Hadoop Configuration	1.7.2
Stop a Hadoop Configuration	1.7.3
Database	1.8
File Management	1.9
Template	1.9.1
Create User Text Template	1.9.1.1
Create Core Text Template	1.9.1.2
Create User Parquet Template	1.9.1.3
Create Core Parquet Template	1.9.1.4
Sync Template	1.9.1.5
Upload	1.9.2
Upload File to Text Template	1.9.2.1
Upload File to Parquet Template	1.9.2.2
Copy File to Shared Folder	1.9.2.3
Delete File	1.9.2.4
Recover Deleted File	1.9.2.5
Watcher	1.9.3
Create Watcher	1.9.3.1
Run Watcher Interactively	1.9.3.2
Run Watcher in Background	1.9.3.3
Delete Watcher	1.9.3.4
Model	1.9.4
Create Model	1.9.4.1
Run Model Interactively	1.9.4.2
Run Model in Background	1.9.4.3
Delete Model	1.9.4.4
Timesheet	1.10
Projects	1.10.1
Rates	1.10.2
Timesheet Tasks	1.10.3
Assignment	1.10.4
Submission	1.10.5
Tasks	1.11
Categories	1.11.1
Task Queue	1.11.2
Task Template	1.11.3
Activity	1.12

---

<a href="#">Current</a>	1.12.1
<a href="#">History</a>	1.12.2
<a href="#">Reports</a>	1.13
<a href="#">Queries</a>	1.13.1

---

## 1.1 OnCloudTime User Manual

This is a tool that provides a simplified interface for carrying out big data operations on data sets with a simplified interface

## 1.2 Application Setup and Installation

### 1.2.1 Architecture

OnCloudTime requires PHP 7.1 and MySQL 5.6 so the recommendation is to run it on the LAMP (Linux Apache MySQL PHP) stack.

### 1.2.2 Server Setup

There are three different ways to setup the application which are covered below:

1. Docker Container Installation on EC2 Instance
2. Container Installation via ECS
3. Manual installation of all software on an EC2 Instance

#### 1.2.2.1 Pre-requisites

The pre-requisites for the setup are to create the following:

- An ECR repository for the docker images
- Athena database
- Aurora database with MySQL 5.6 compatibility
- SQS Queue
- SNS Application Topic
- A secret to store the credentials for ECS and access permissions for the ECS roles to read the secret
- Application Load Balancer with HTTPS

**NOTE** The recommended region to setup all the services is `us-east-1`

##### 1.2.2.1.1 Aurora database

Aurora is a pre-provisioned database with MySQL compatibility which removes the need to additional containers running it. The steps to setup are shown in the images below:

## Select engine

### Engine options

Amazon Aurora

**Amazon  
Aurora**

MySQL



MariaDB



PostgreSQL



Oracle

**ORACLE®**

Microsoft SQL Server



### Amazon Aurora

Amazon Aurora is a MySQL- and PostgreSQL-compatible enterprise-class database, starting at <\$1/day.

- Up to 5 times the throughput of MySQL and 3 times the throughput of PostgreSQL
- Up to 64TiB of auto-scaling SSD storage
- 6-way replication across three Availability Zones
- Up to 15 Read Replicas with sub-10ms replica lag
- Automatic monitoring and failover in less than 30 seconds

#### Edition

MySQL 5.6-compatible

Aurora Serverless and Parallel Query capacities are only available with this edition.

MySQL 5.7-compatible

PostgreSQL-compatible

## Specify DB details

### Configuration

Estimate your monthly costs for the DB Instance using the [AWS Simple Monthly Calculator](#)

DB engine

Aurora - compatible with MySQL 5.6

Capacity type [Info](#)

**Provisioned**

You provision and manage the server instance sizes.

**Provisioned with Aurora parallel query enabled** [Info](#)

You provision and manage the server instance sizes, and Aurora improves the performance of analytic queries by pushing processing down to the Aurora storage layer (currently available for Aurora MySQL 5.6)

**Serverless** [Info](#)

You specify the minimum and maximum of resources for a DB cluster. Aurora scales the capacity based on database load.

DB engine version [Info](#)

Aurora (MySQL)-5.6.10a

DB instance class [Info](#)

db.t2.small — 1 vCPU, 2 GiB RAM

Multi-AZ deployment [Info](#)

Create Replica in Different Zone

No

### Settings

DB instance identifier [Info](#)

Specify a name that is unique for all DB instances owned by your AWS account in the current region.

oncloudtime

DB instance identifier is case insensitive, but stored as all lower-case, as in "mydbinstance". Must contain from 1 to 63 alphanumeric characters or hyphens (1 to 15 for SQL Server). First character must be a letter. Cannot end with a hyphen or contain two consecutive hyphens.

Master username [Info](#)

Specify an alphanumeric string that defines the login ID for the master user.

oncloudtime

Master Username must start with a letter. Must contain 1 to 16 alphanumeric characters.

Master password [Info](#)

\*\*\*\*\*

Master Password must be at least eight characters long, as in "mypassword". Can be any printable ASCII character except "/", "", or "@".

Confirm password [Info](#)

\*\*\*\*\*

Cancel

Previous

Next

RDS > Create database

## Configure advanced settings

### Network & Security

**Virtual Private Cloud (VPC) [Info](#)**  
VPC defines the virtual networking environment for this DB instance.

C

Only VPCs with a corresponding DB subnet group are listed.

**Subnet group [Info](#)**  
DB subnet group that defines which subnets and IP ranges the DB instance can use in the VPC you selected.

▼

**Public accessibility [Info](#)**

Yes  
EC2 instances and devices outside of the VPC hosting the DB instance will connect to the DB instances. You must also select one or more VPC security groups that specify which EC2 instances and devices can connect to the DB instance.

No  
DB instance will not have a public IP address assigned. No EC2 instance or devices outside of the VPC will be able to connect.

**Availability zone [Info](#)**

▼

**VPC security groups**  
Security groups have rules authorizing connections from all the EC2 instances and devices that need to access the DB instance.

Create new VPC security group

Choose existing VPC security groups

▼

X

### Database options

DB cluster identifier [Info](#)

If you do not provide one, a default identifier based on the instance identifier will be used.

Database name [Info](#)

If you do not specify a database name, Amazon RDS does not create a database.

Port [Info](#)

TCP/IP port the DB instance will use for application connections.

DB parameter group [Info](#)

▼

DB cluster parameter group [Info](#)

▼

Option group [Info](#)

▼

## Encryption

### Encryption

- Enable encryption [Learn more](#) 

Select to encrypt the given instance. Master key ids and aliases appear in the list after they have been created using the Key Management Service(KMS) console.

- Disable encryption

## Failover

### Priority [Info](#)

No preference 

## Backup

### Backup retention period [Info](#)

Select the number of days that Amazon RDS should retain automatic backups of this DB instance.

1 day 

- Copy tags to snapshots

## Backtrack

Backtrack lets you quickly move an Aurora database to a prior point in time without needing to restore data from a backup. [Info](#)

- Enable Backtrack

- Disable Backtrack

## Monitoring

Enhanced monitoring

**Enable enhanced monitoring**  
Enhanced monitoring metrics are useful when you want to see how different processes or threads use the CPU.

**Disable enhanced monitoring**

Monitoring Role	Granularity
Default	60 seconds

I authorize RDS to create the IAM role rds-monitoring-role.

## Log exports

Select the log types to publish to Amazon CloudWatch Logs

Audit log  
 Error log  
 General log  
 Slow query log

**IAM role**  
The following service-linked role is used for publishing logs to CloudWatch Logs.

RDS Service Linked Role

**ⓘ Ensure that General, Slow Query, and Audit Logs are turned on. Error logs are enabled by default.**

[Learn more](#)

### Maintenance

Auto minor version upgrade [Info](#)

Enable auto minor version upgrade  
Enables automatic upgrades to new minor versions as they are released. The automatic upgrades occur during the maintenance window for the DB instance.

Disable auto minor version upgrade

Maintenance window [Info](#)  
Select the period in which you want pending modifications or patches applied to the DB instance by Amazon RDS.

Select window  
 No preference

### Deletion protection

Enable deletion protection  
Protects the database from being deleted accidentally. While this option is enabled, you can't delete the database.

Cancel Previous Create database

Your DB instance is being created.  
Note: Your instance may take a few minutes to launch.

### Connecting to your DB instance

Once Amazon RDS finishes provisioning your DB instance, you can use a SQL client application or utility to connect to the instance.  
[Learn about connecting to your DB instance](#)

### Usage charges

The following selections disqualify the instance from being eligible for the free tier:

- DB instance class
- Engine

You will be charged normal RDS Prices. [Learn More](#)

Estimate your monthly costs for the DB Instance using the [AWS Simple Monthly Calculator](#)

All DB instances View DB instance details

#### 1.2.2.1.1 Amazon Elastic Container Repository

1. Create a new repository

ECR > Repositories > Create repository

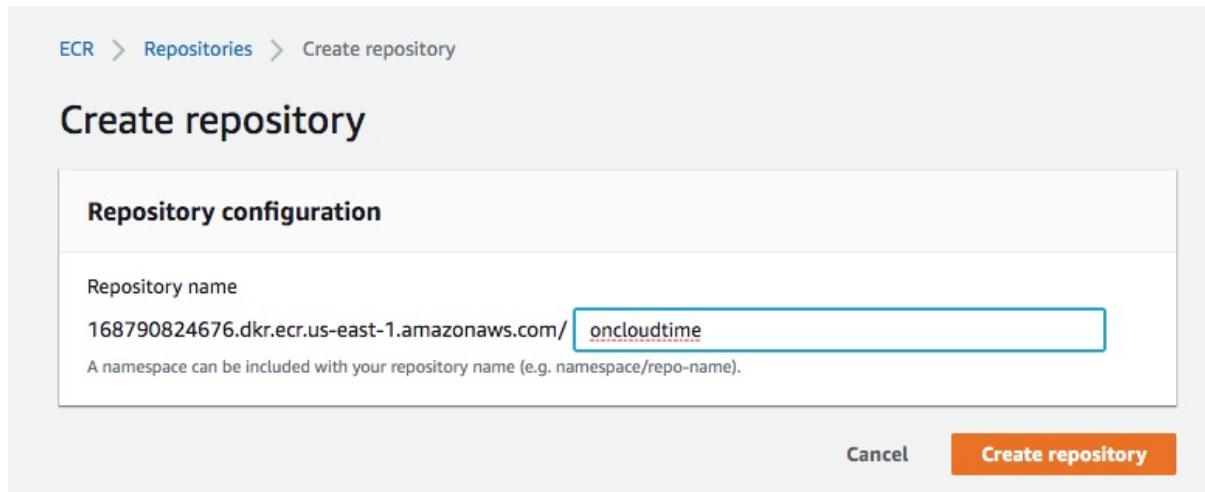
## Create repository

**Repository configuration**

Repository name  
168790824676.dkr.ecr.us-east-1.amazonaws.com/ **oncloudtime**

A namespace can be included with your repository name (e.g. namespace/repo-name).

**Create repository**



- When successfully created it is shown in a list

Successfully created repository		<a href="#">View push commands</a>	X
Repositories (5)		<a href="#">Create repository</a>	
<a href="#">Find Repositories</a>		<a href="#">View push commands</a>	Delete
Repository name	URI	Created at	
oncloudtime	168790824676.dkr.ecr.us-east-1.amazonaws.com/oncloudtime	03/16/19, 1:16:17 PM	



- Clicking the [View push commands](#) button shows a list of commands that can be used to interact with the repository via the cli

## Push commands for oncloudtime

[macOS / Linux](#)[Windows](#)

Ensure you have installed the latest version of the AWS CLI and Docker. For more information, see the [ECR documentation](#).

1. Retrieve the login command to use to authenticate your Docker client to your registry.

Use the AWS CLI:

```
$ aws ecr get-login --no-include-email --region us-east-1
```



Note: If you receive an "Unknown options: --no-include-email" error when using the AWS CLI, ensure that you have the latest version installed. [Learn more](#)

2. Build your Docker image using the following command. For information on building a Docker file from scratch see the instructions [here](#). You can skip this step if your image is already built:

```
docker build -t oncloudtime .
```



3. After the build completes, tag your image so you can push the image to this repository:

```
docker tag oncloudtime:latest 168790824676.dkr.ecr.us-east-1.amazonaws.com/oncloudtime:latest
```



4. Run the following command to push this image to your newly created AWS repository:

```
docker push 168790824676.dkr.ecr.us-east-1.amazonaws.com/oncloudtime:latest
```



### 1.2.2.1.1 Athena Database

### 1.2.2.1.1 SQS Queue

### 1.2.2.1.1 SNS Application Topic

### 1.2.2.1.1 Secret for Configuration Information

1. Create the secret in which the application configuration will be stored

**Step 1 Secret type**

**Step 2 Name and description**

**Step 3 Configure rotation**

**Step 4 Review**

**Store a new secret**

**Select secret type** Info

Credentials for RDS database  Credentials for Redshift cluster  Credentials for DocumentDB database  Credentials for other database

Other type of secrets (e.g. API key)

**Specify the key/value pairs to be stored in this secret** Info

**Secret key/value**  Plaintext

app_config	
------------	--

+ Add row

**Select the encryption key info**

Select the AWS KMS key to use to encrypt your secret information. You can encrypt using the default service encryption key that AWS Secrets Manager creates on your behalf or a customer master key (CMK) that you have stored in AWS KMS.

DefaultEncryptionKey

Add new key

**Cancel** **Next**

### 1. Set the name of the secret

**Step 1 Secret type**

**Step 2 Name and description**

**Step 3 Configure rotation**

**Step 4 Review**

**Store a new secret**

**Secret name and description** Info

**Secret name**  
Give the secret a name that enables you to find and manage it easily.  
**oncloudtime\_config**

Secret name must contain only alphanumeric characters and the characters /\_=.:@-

**Description - optional**  
Configuration for Oncloudtime

Maximum 250 characters

**Tags - optional**

Key	Value - optional
Enter key	Enter value

**Add** **Remove**

**Cancel** **Previous** **Next**

### 1. Disable the rotation policy

### Add permissions to ecsInstanceRole

#### Attach Permissions

The screenshot shows the 'Attach Policy' step in the AWS IAM console. A single policy, 'SecretsManagerReadWrite', is selected from a list. The policy description indicates it provides read/write access to AWS Secrets Manager via the AWS Management Console. Below the list is a detailed table of the policy's permissions, showing actions like 'List', 'Read', and 'Write' across various AWS services.

Service	Access level	Resource	Request condition
CloudFormation	Limited: List, Read, Write	All resources	None
EC2	Limited: List	All resources	None
KMS	Limited: List, Read	All resources	None
Lambda	Limited: List	All resources	None
RDS	Limited: List	All resources	None
Resource Group Tagging	Limited: Read	All resources	None
S3	Limited: Read	BucketName   string like   aws:serverlessrepo-changesets*	None
Secrets Manager	Full access	All resources	None
Serverless Application Re...	Limited: Write	arn:aws:serverlessrepo:":applications/SecretsManager"	None

[Cancel](#) [Attach policy](#)

1. Add privileges to the **ecsInstanceRole** (for EC2 compatibility) and **ecsTaskExecutionRole** (for Fargate compatibility) roles so that it can read the secrets on application startup

Welcome to Identity and Access Management

IAM users sign-in link:  
<https://168790824676.signin.aws.amazon.com/console>

**Customize**

**IAM Resources**

Users: 1 Roles: 15  
Groups: 0 Identity Providers: 0  
Customer Managed Policies: 0

**Security Status**

2 out of 5 complete.

- Delete your root access keys** (Green checkmark)
- Activate MFA on your root account** (Orange warning icon)
- Create individual IAM users** (Green checkmark)
- Use groups to assign permissions** (Orange warning icon)
- Apply an IAM password policy** (Orange warning icon)

IAM roles are a secure way to grant permissions to entities that you trust. Examples of entities include the following:

- IAM user in another account
- Application code running on an EC2 instance that needs to perform actions on AWS resources
- An AWS service that needs to act on resources in your account to provide its features
- Users from a corporate directory who use identity federation with SAML

IAM roles issue keys that are valid for short durations, making them a more secure way to grant access.

**Additional resources:**

- IAM Roles FAQ
- IAM Roles Documentation
- Tutorial: Setting Up Cross Account Access
- Common Scenarios for Roles

**Create role Delete role**

Role name	Description	Trusted entities
ecsInstanceRole	Allows EC2 instances to call AWS services on your behalf.	AWS service: ec2

Showing 1 result

**Add permissions to ecsInstanceRole****Attach Permissions**

**Create policy**

**Filter policies** Q: Secre Showing 1 result

Policy name	Type	Used as	Description
SecretsManagerReadWrite	AWS managed	None	Provides read/write access to AWS Secrets Manager via the AWS Management Console. Note: this excludes IAM actions, so combine with IAMFullAccess if rotation configuration is required.

SecretsManagerReadWrite  
Provides read/write access to AWS Secrets Manager via the AWS Management Console. Note: this excludes IAM actions, so combine with IAMFullAccess if rotation configuration is required.

This policy defines some actions, resources, or conditions that do not provide permissions. To grant access, policies must have an action that has an applicable resource or condition. For details, choose Show remaining. [Learn more](#)

**Policy summary** { JSON

Q: Filter

Service	Access level	Resource	Request condition
Allow (9 of 174 services) Show remaining 165			
CloudFormation	Limited: List, Read, Write	All resources	None
EC2	Limited: List	All resources	None
KMS	Limited: List, Read	All resources	None
Lambda	Limited: List	All resources	None
RDS	Limited: List	All resources	None
Resource Group Tagging	Limited: Read	All resources	None
S3	Limited: Read	BucketName   string like   awsserverlessrepo-changesets*	None
Secrets Manager	Full access	All resources	None
Serverless Application Re...	Limited: Write	arn:aws:serverlessrepo:":":applications/SecretsManager*	None

**Roles > ecsInstanceRole Summary**

**Role ARN** arn:aws:iam::168790824676:role/ecsInstanceRole

**Role description** Allows EC2 instances to call AWS services on your behalf. | Edit

**Instance Profile ARNs** arn:aws:iam::168790824676:instance-profile/ecsInstanceRole

**Path** /

**Creation time** 2018-12-16 20:36 UTC+0300

**Maximum CLI/API session duration** 1 hour [Edit](#)

**Permissions** **Trust relationships** **Tags** **Access Advisor** **Revoke sessions**

**Permissions policies (2 policies applied)**

**Attach policies** **Add inline policy**

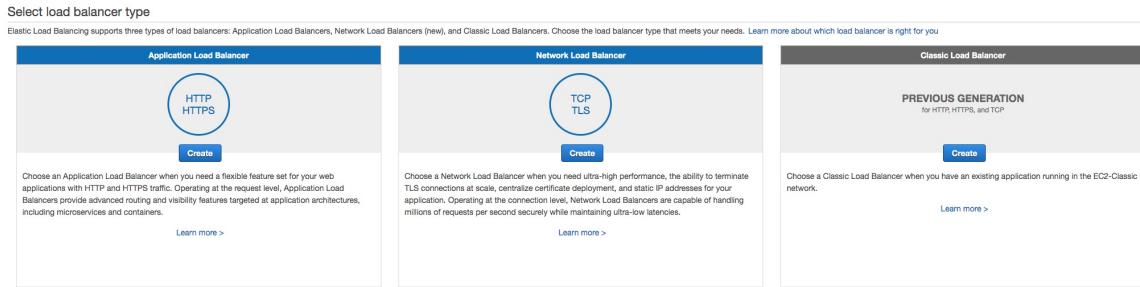
Policy name	Policy type
AmazonS3ReadOnlyAccess	AWS managed policy
AmazonEC2ContainerServiceforEC2Role	AWS managed policy

**Permissions boundary (not set)**

### 1.2.2.1.1 Application Load Balancer with HTTPS

The load balancer provides a single point of access for your application via a predefined sub-domain and routes, also providing security negating the need to setup a reverse proxy.

1. Select the type of load balancer - always select `Application Load Balancer`



2. Configure ELB:

- o Name `oncloudtime`
- o Scheme `internet-facing` which is accessible from the Internet
- o IP Address type `ipv4`
- o Listeners - add HTTPS on 443
- o Availability Zones - select a VPC and at least two subnets

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 already selected for you. This configuration receives HTTP traffic on port 80.

Name	<code>elb-oncloudtime</code>
Scheme	<input checked="" type="radio"/> internet-facing <input type="radio"/> internal
IP address type	<code>ipv4</code>

**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
<code>HTTPS (Secure HTTP)</code>	443

**Add listener**

3. Add an existing HTTPS certificate, upload or create a new one via ACM

Step 2: Configure Security Settings

Select default certificate

AWS Certificate Manager (ACM) is the preferred tool to provision and store server certificates. If you previously stored a server certificate using IAM, you can deploy it to your load balancer. Learn more about HTTPS listeners and certificate management.

Certificate type	<input checked="" type="radio"/> Choose a certificate from ACM (recommended) <input type="radio"/> Upload a certificate to ACM (recommended) <input type="radio"/> Choose a certificate from IAM <input type="radio"/> Upload a certificate to IAM
Certificate name	<code>oncloudtime.stytechgroup.com (arn:aws:acm:us-east-1:168790824676)</code>

Select Security Policy

Security policy	<code>ELBSecurityPolicy-2016-08</code>
-----------------	--

4. Add security groups - these must provide access HTTPS ports as specified in Step 2

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

**Step 3: Configure Security Groups**

A security group is a set of firewall rules that control the traffic to your load balancer. On this page, you can add rules to allow specific traffic to reach your load balancer. First, decide whether to create a new security group or select an existing one.

Assign a security group:  Create a new security group  Select an existing security group

Security Group ID	Name	Description	Actions
sg-035da31703ee9e59	default	default VPC security group	Copy to new
sg-065f1a4031611261	EC2ContainerService-oncloudtime-demo-2-EcsSecurityGroup-10J642K5ZGM77	ECS Allowed Ports	Copy to new
sg-08a39b7073974fb65	onclou-5329	2019-04-15T18:09:51.467Z	Copy to new

## 5. Select Security Policy - leave the default ELBSecurityPolicy-2016-08

### 6. Configure Routing

- o Target group
  - Select create a new target group with name tg-elb-oncloudtime-https
  - Target type Instance
  - Protocol HTTPS
  - Port 443
- o Health Checks
  - Protocol HTTP
  - Path /monitor/health/

(this will later be pointed to ECS instances)

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

**Step 5: Register Targets**

Register targets with your target group. If you register a target in an enabled Availability Zone, the load balancer starts routing requests to the targets as soon as the registration process completes and the target passes the initial health checks.

**Registered targets**

To deregister instances, select one or more registered instances and then click Remove.

Remove	Instance	Name	Port	State	Security groups	Zone
No instances available.						

**Instances**

To register additional instances, select one or more running instances, specify a port, and then click Add. The default port is the port specified for the target group. If the instance is already registered on the specified port, you must specify a different port.

Add to registered	on port 443	Search Instances	Instance	Name	State	Security groups	Zone	Subnet ID	Subnet CIDR
No instances available.									

Cancel   Previous   Next: Review

## 7. Do not register any targets as these will be added later

## 8. Confirmation of successful creation or any errors that may have occurred

### Load Balancer Creation Status

✔ Successfully created load balancer  
Load balancer elb-oncloudtime was successfully created.  
Note: It might take a few minutes for your load balancer to be fully set up and ready to route traffic, and for the targets to complete the registration process and pass the initial health checks.

Close

## 9. Set the target group to have sticky sessions keeping each set of requests to a single instance

- o On the created ELB select the target group

The screenshot shows the AWS CloudFront console with the 'Create New Distribution' wizard. Step 1 of 3 is completed, with the domain name 'www.example.com' entered. Step 2, 'Select Origin', is in progress, showing a list of origins including 'Amazon S3' and 'Custom'. Step 3, 'Configure Price Class', is shown with the 'Optimize for CloudFront' option selected. The 'Next Step' button is visible at the bottom.

### 1.2.2.2 Docker on EC2 Instance

1. Create a new EC2 server from Amazon Linux AMI 2018.03.0 (HVM), SSD Volume Type - ami-01e24be29428c15b2
2. SSH into the server using `ssh -i /path/my-key-pair.pem ec2-user@public-dns-hostname`
3. Install docker on the instance:

```
[ec2-user]$ sudo yum update -y
[ec2-user]$ sudo yum install -y docker
[ec2-user]$ sudo service docker start
```

4. Next, add the `ec2-user` to the `docker` group so you can execute Docker commands without using sudo. Note that you'll have to log out and log back in for the settings to take effect:

```
[ec2-user]$ sudo usermod -a -G docker ec2-user
[ec2-user]$ exit
```

5. To test run `docker info`
6. Download the docker publishing project package by running `wget` on the provided url for example [https://s3-us-west-2.amazonaws.com/techsoft/oncloudtime/releases/oncloudtime-docker-26\\_Jan\\_2019.zip](https://s3-us-west-2.amazonaws.com/techsoft/oncloudtime/releases/oncloudtime-docker-26_Jan_2019.zip)
7. Unzip the package by running `unzip oncloudtime-docker.zip` which opens in the current folder
8. Update the following configurations:
  - `docker_config_oncloudtime.cnf`
    - `release_package_path` with the url of the code package

- `repository_url` the url of the ECR repository to which the application image is deployed
  - `config_url` the ARN of the secret to which the configuration is stored, which is passed as an environment variable to the docker container
  - `host_port` the host port on which to run the container
  - `.aws_oncloudtime` - add the AWS access key and secret credentials to enable deployment of the image to ECR
9. Build the docker images for the application by running `sh docker_build_oncloudtime.sh`
10. Start the application by running `sh docker_run_oncloudtime.sh`
11. Open the application by accessing the public dns name of your EC2 instance `https://public-dns-hostname:host_port`

#### NOTES:

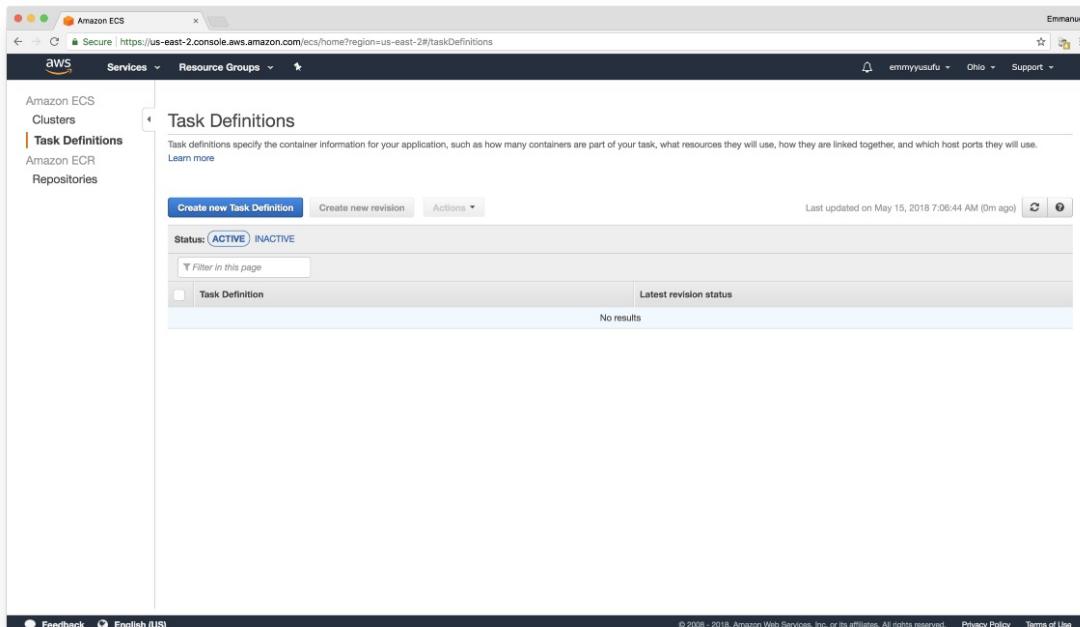
The following scripts are available in the package directory:

- `docker_bashintocontainer_oncloudtime.sh` - open a bash script into a running container
- `docker_build_oncloudtime.sh` - builds a docker image using the Dockerfile in the web directory and application code in a url specified by `release_package_url` in the `docker_config_oncloudtime.cnf`
- `docker_config_oncloudtime.cnf` - configuration settings file used by all docker containers
- `docker_pushtoecr_oncloudtime.sh` - build the containers and push to the ECR repository specified by the `repository_url` variable in `docker_config_oncloudtime.cnf` file
- `docker_remove_oncloudtime.sh` - delete all containers, images and volumes on the server
- `docker_run_oncloudtime.sh` - run the docker container mapping the image to the port specified by the `host_port` variable in the `docker_config_oncloudtime.cnf` file. The image is deleted once the container is stopped
- `docker_stop_oncloudtime` - stop a running container

### 1.2.2.3 ECS Cluster with EC2 Launch Compatibility

This requires the task definition to be with EC2 launch compatibility which allows more control over the instance creation and configuration

1. Create a new task definition - From Task Definitions in the ECS dashboard, press on the Create new Task Definition (ECS) button



- o Select the launch compatibility type to EC2 - this will not work for ECS Fargate clusters

### Create new Task Definition

**Step 1: Select launch type compatibility**

Step 2: Configure task and container definitions

Select launch type compatibility

Select which launch type you want your task definition to be compatible with based on where you want to launch your task.

**FARGATE**



Price based on task size

Requires network mode awsvpc

AWS-managed infrastructure, no Amazon EC2 instances to manage

**EC2**



Price based on resource usage

Multiple network modes available

Self-managed infrastructure using Amazon EC2 instances

\*Required

Cancel Next step

- o Set a task name and use the following steps:
    - Add Container: `oncloudtime` (the one we pushed).
    - Image: the URL to your container pushed to ECR e.g., `168790824676.dkr.ecr.us-east-1.amazonaws.com/oncloudtime` used in the example
    - Soft limit: 768
    - Map 80 (host) to 80 (container) for `oncloudtime`
  - o Environment Variables - this one is very important:
    - CONFIG\_URL: e.g., `arn:aws:secretsmanager:us-east-1:168790824676:secret:oncloudtime_config-Wa3muL` - the ARN of the secret in which the application configuration is stored for the cluster. Without this each instance of the application will default to start at the installation screen
2. Create Cluster to run the application containers run using configurations similar to EC2 instances.
    - o Select a cluster template - EC2 Linux + Networking

## Select cluster template

The following cluster templates are available to simplify cluster creation. Additional configuration and integrations can be added later.

<p><b>Networking only</b></p> <p><u>Resources to be created:</u></p> <ul style="list-style-type: none"> <li>Cluster</li> <li>VPC (optional)</li> <li>Subnets (optional)</li> </ul> <p><b>Powered by AWS Fargate</b></p>	<p><b>EC2 Linux + Networking</b></p> <p><u>Resources to be created:</u></p> <ul style="list-style-type: none"> <li>Cluster</li> <li>VPC</li> <li>Subnets</li> </ul> <p>Auto Scaling group with Linux AMI</p>
<p><b>EC2 Windows + Networking</b></p> <p><u>Resources to be created:</u></p> <ul style="list-style-type: none"> <li>Cluster</li> <li>VPC</li> <li>Subnets</li> </ul> <p>Auto Scaling group with Windows AMI</p>	

\*Required

[Cancel](#)

[Next step](#)

- o Define the following:
    - Cluster name: `oncloudtime-cluster`
    - Provisioning Model: On-Demand Instances
    - EC2 instance type: `t2.micro`
    - Number of instances: 1
    - EBS storage: 22
    - Key pair: (Select an existing key-pair)
    - VPC: Create a new VPC (you also have an option of selecting an existing VPC but ensure that it has a public subnet). Selecting a new VPC automatically selects to create a new security group
    - Security group inbound rules - leave the default CDR block with port 80 (for http). This will be updated later to include HTTPS on 443
    - Container Instance IAM role select `ecsInstanceRole` if available otherwise select to create a new role.  
**NOTE:** This role will have to be given permission to access the secret for the configuration variable passed to the Task definition
3. Create a service to run the task definition on the cluster by defining the following:
    - o Launch type: `EC2`
    - o Task definition - select the appropriate revision, the latest is usually the recommended
    - o Service Name - use the cluster name then add service for example `oncloudtime-cluster-service`
    - o Service type - select Replica
    - o Number of tasks - 1
    - o Deployment - leave default `Rolling Update`
    - o Task Placement - leave the default `AZ Balanced Spread`

- o Auto Scaling - select `Do not adjust the service's desired count`

## Configure service

A service lets you specify how many copies of your task definition to run and maintain in a cluster. You can optionally use an Elastic Load Balancing load balancer to distribute incoming traffic to containers in your service. Amazon ECS maintains that number of tasks and coordinates task scheduling with the load balancer. You can also optionally use Service Auto Scaling to adjust the number of tasks in your service.

Launch type	<input type="radio"/> FARGATE <input checked="" type="radio"/> EC2	<a href="#">i</a>
Task Definition	Family <input type="text" value="oncloudtime-ec2"/>	<a href="#">i</a>
Revision	<input type="text" value="4 (latest)"/>	<a href="#">i</a>
Cluster	<input type="text" value="oc-demo"/>	<a href="#">i</a>
Service name	<input type="text" value="oncloudtime"/>	<a href="#">i</a>
Service type*	<input checked="" type="radio"/> REPLICA <input type="radio"/> DAEMON	<a href="#">i</a>
Number of tasks	<input type="text" value="1"/>	<a href="#">i</a>
Minimum healthy percent	<input type="text" value="100"/>	<a href="#">i</a>
Maximum percent	<input type="text" value="200"/>	<a href="#">i</a>

## Deployments

Choose a deployment option for the service.

- Deployment type\*
- Rolling update [i](#)
  - Blue/green deployment (powered by AWS CodeDeploy) [i](#)

This sets AWS CodeDeploy as the deployment controller for the service. A CodeDeploy application and deployment group are created automatically with **default settings** for the service. To change to the rolling update deployment type after the service has been created, you must re-create the service and select the "rolling update" deployment type.

## Task Placement

Lets you customize how tasks are placed on instances within your cluster. Different placement strategies are available to optimize for availability and efficiency.

Placement Templates	<input type="text" value="AZ Balanced Spread"/>	<a href="#">Edit</a>
This template will spread tasks across availability zones and within the availability zone spread tasks across instances. <a href="#">Learn more</a> .		
<b>Strategy:</b> spread(attribute:ecs.availability-zone), spread(instanceId)		

## Create Service

- Step 1: Configure service
- Step 2: Configure network
- Step 3: Set Auto Scaling (optional)**
- Step 4: Review

### Set Auto Scaling (optional)

Automatically adjust your service's desired count up and down within a specified range in response to CloudWatch alarms. You can modify your Service Auto Scaling configuration at any time to meet the needs of your application.

- Service Auto Scaling
- Do not adjust the service's desired count
  - Configure Service Auto Scaling to adjust your service's desired count

\*Required

[Cancel](#)

[Previous](#)

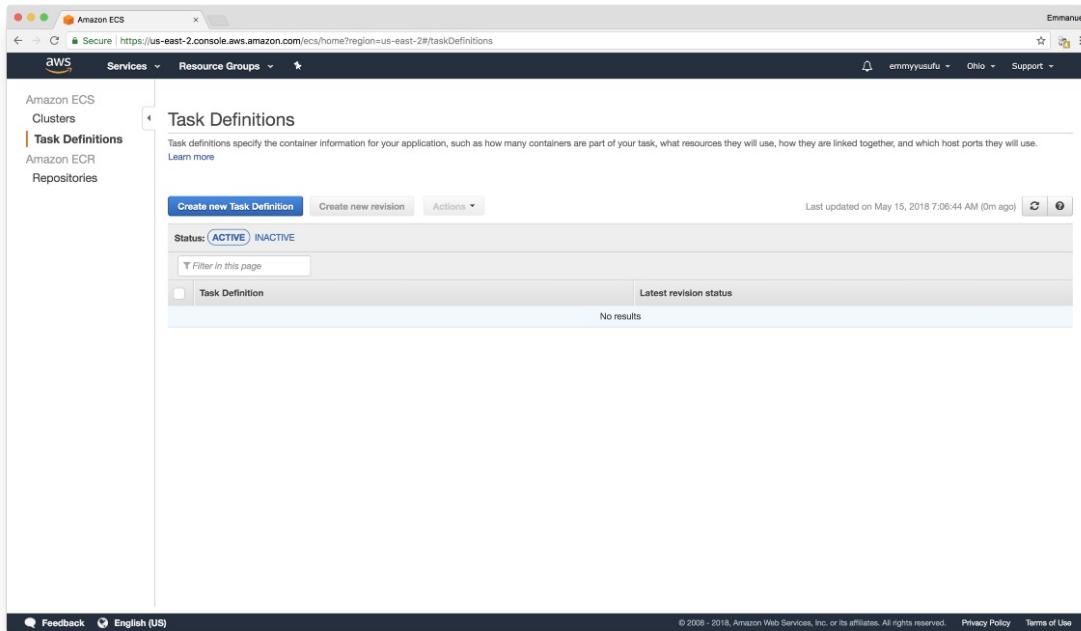
[Next step](#)

**NOTE:** Once the service is created you may need to wait about 3-4 minutes for the tasks to be created and started.

### 1.2.2.4 ECS Cluster with Fargate Launch Compatibility

This requires the task definition to be with Fargate launch compatibility which provides more automation for tasks and service management

1. Create a new task definition - From Task Definitions in the ECS dashboard, press on the Create new Task Definition (ECS) button



- o Select the launch compatibility type to Fargate - this will not work for clusters of launch type EC2

#### Create new Task Definition

Step 1: Select launch type compatibility  
Step 2: Configure task and container definitions

#### Select launch type compatibility

Select which launch type you want your task definition to be compatible with based on where you want to launch your task.

**FARGATE**



Price based on task size

Requires network mode awsvpc

AWS-managed infrastructure, no Amazon EC2 instances to manage

**EC2**



Price based on resource usage

Multiple network modes available

Self-managed infrastructure using Amazon EC2 instances

\*Required

Cancel

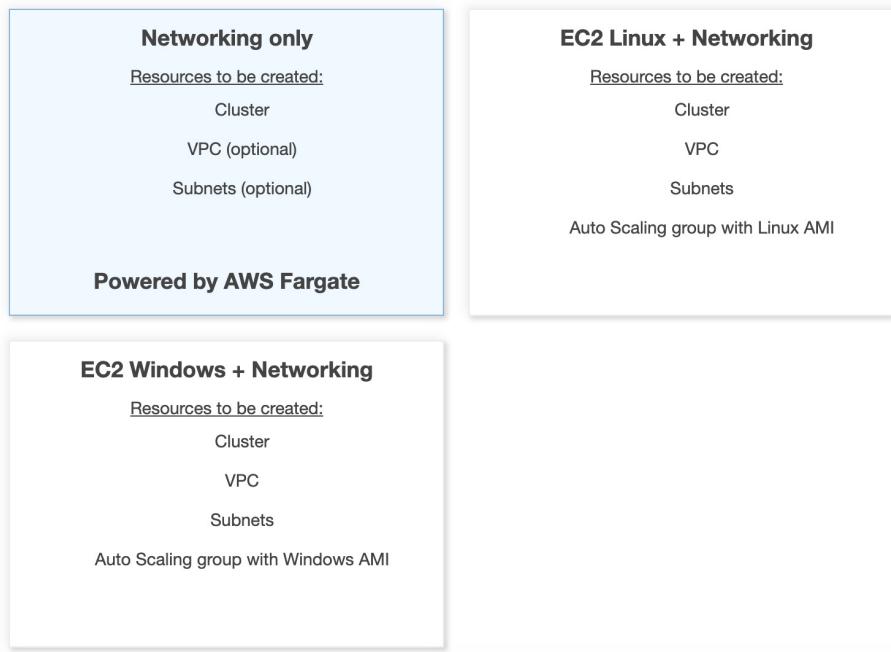
Next step

- o Configure Task and Container Definitions
  - Task Definition Name `oncloudtime-fargate`
  - Task Role, select `ecstaskExecutionRole`
  - Task execution IAM Role select `ecsTaskExecutionRole`
  - Task size:
    - Task memory (GB) select `1GB`

- Task CPU (vCPU) select 0.5 vCPU
  - Configure the container
    - Name: `oncloudtime` (the one we pushed).
    - Image: the URL to your container pushed to ECR e.g., `168790824676.dkr.ecr.us-east-1.amazonaws.com/oncloudtime` used in the example
    - Soft limit: 768
    - Select container port `80` and protocol `tcp`
  - Environment Variables - this one is very important as this contains the configurations to connect to the different services:
    - Key: `CONFIG_URL`
    - Value: e.g., `arn:aws:secretsmanager:us-east-1:168790824676:secret:oncloudtime_config-Wa3muL` - the ARN of the secret in which the application configuration is stored for the cluster. Without this each instance of the application will default to the installation screen
2. Create Cluster to run the application containers run using configurations similar to EC2 instances.
- Select a cluster template - Networking only

### Select cluster template

The following cluster templates are available to simplify cluster creation. Additional configuration and integrations can be added later.



- Define the following:
  - Cluster name: `oncloudtime-fargate-cluster`
  - Networking - do not create a VPC
- 3. Select the cluster to display its details

[Clusters](#) > oncloudtime-fargate-cluster

Cluster : oncloudtime-fargate-cluster

Get a detailed view of the resources on your cluster.

Status: ACTIVE

Registered container instances: 0

Pending tasks count: 0 Fargate, 0 EC2

Running tasks count: 0 Fargate, 0 EC2

Active service count: 0 Fargate, 0 EC2

Draining service count: 0 Fargate, 0 EC2

[Delete Cluster](#)

Services	Tasks	ECS Instances	Metrics	Scheduled Tasks	Tags	
<a href="#">Create</a>	<a href="#">Update</a>	<a href="#">Delete</a>	<a href="#">Actions</a>	Last updated on May 21, 2019 10:20:29 PM (0m ago) <a href="#">Filter</a> <a href="#">Help</a>		
<a href="#">Filter in this page</a>		Launch type: ALL	Service type: ALL			
<input type="checkbox"/> Service Name	Status	Service type	Task Definition	Desired tasks	Running tasks	Launch type
Platform version						
No results						

4. Create a service to run the task definition on the cluster by defining the following:

- o Launch type: Fargate
- o Task definition - select the appropriate revision, the latest is usually the recommended
- o Service Name: oncloudtime-fargate-service
- o Number of tasks - 2
- o Minimum healthy percent - 100
- o Maximum health percent - 200
- o Deployment - leave default Rolling Update

## Configure service

A service lets you specify how many copies of your task definition to run and maintain in a cluster. You can optionally use an Elastic Load Balancing load balancer to distribute incoming traffic to containers in your service. Amazon ECS maintains that number of tasks and coordinates task scheduling with the load balancer. You can also optionally use Service Auto Scaling to adjust the number of tasks in your service.

<input checked="" type="radio"/> FARGATE <input type="radio"/> EC2 <span style="float: right;">i</span>	
<b>Task Definition</b> Family: <input type="text" value="oncloudtime-fargate"/> <span style="float: right;">▼</span> <span style="margin-left: 10px;"><b>Enter a value</b></span> Revision: <input type="text" value="5 (latest)"/> <span style="float: right;">▼</span>	
<b>Platform version</b> <input type="text" value="LATEST"/> <span style="float: right;">▼</span> <span style="margin-left: 10px;">i</span>	
<b>Cluster</b> <input type="text" value="oncloudtime-fargate-cluster"/> <span style="float: right;">▼</span> <span style="margin-left: 10px;">i</span>	
<b>Service name</b> <input type="text" value="oncloudtime-fargate-service"/> <span style="float: right;">i</span>	
<b>Service type*</b> <input checked="" type="radio"/> REPLICA <span style="float: right;">i</span>	
<b>Number of tasks</b> <input type="text" value="2"/> <span style="float: right;">i</span>	
<b>Minimum healthy percent</b> <input type="text" value="100"/> <span style="float: right;">i</span>	
<b>Maximum percent</b> <input type="text" value="200"/> <span style="float: right;">i</span>	

## Deployments

Choose a deployment option for the service.

- Deployment type\***  Rolling update i  
 Blue/green deployment (powered by AWS CodeDeploy) i

This sets AWS CodeDeploy as the deployment controller for the service. A CodeDeploy application and deployment group are created automatically with **default settings** for the service. To change to the rolling update deployment type after the service has been created, you must re-create the service and select the "rolling update" deployment type.

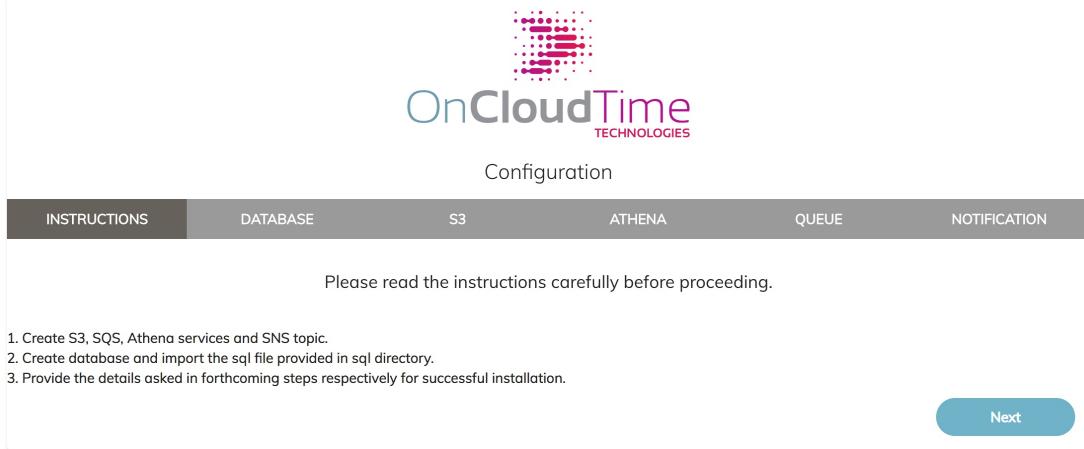
5. VPC and Security Groups
  - o Cluster VPC - select an existing VPC with public facing subnets
  - o Subnets - select at least 2 subnets from the VPC, the list is automatically provided
  - o Security groups - a default group is selected with access to port 80. Make sure that the security groups have access to the database host (RDS),
  - o Auto-assign public IP - leave default **ENABLED**
6. Health Check Grace Period - leave unassigned will be updated when ELB is selected
7. Load Balancing - select None
8. Leave the defaults for the Service discovery section
9. Auto Scaling (option) - leave default value **Do not adjust the service's desired count**

10. After review select `Create Service`

### 1.2.3 Application Installation

1. Download the application package, unzip it and copy the contents to `/var/www/html/oncloudtime`
2. Run the script `/var/www/html/oncloudtime/deploy.sh` to set the correct permissions for the different directories
3. Create a database `oncloudtime` and run the initial data script from `/var/www/html/oncloudtime/sql/new_install.sql`
4. Update the installation configuration file at `/var/www/html/oncloudtime/config/install.yaml` with the following:
  - o Change the value of `need_update` parameter to `yes` to unlock the installer
  - o Change the value of `rdbms_host` to `localhost` to connect to the local database
5. Create the following services on AWS:
  - o AWS Key and Secret
  - o Athena database
  - o SQS Queue
  - o SNS Topic
6. Open the url [<http://public-dns-hostname/oncloudtime/install>] to start the installer below:

- o Installation Instructions

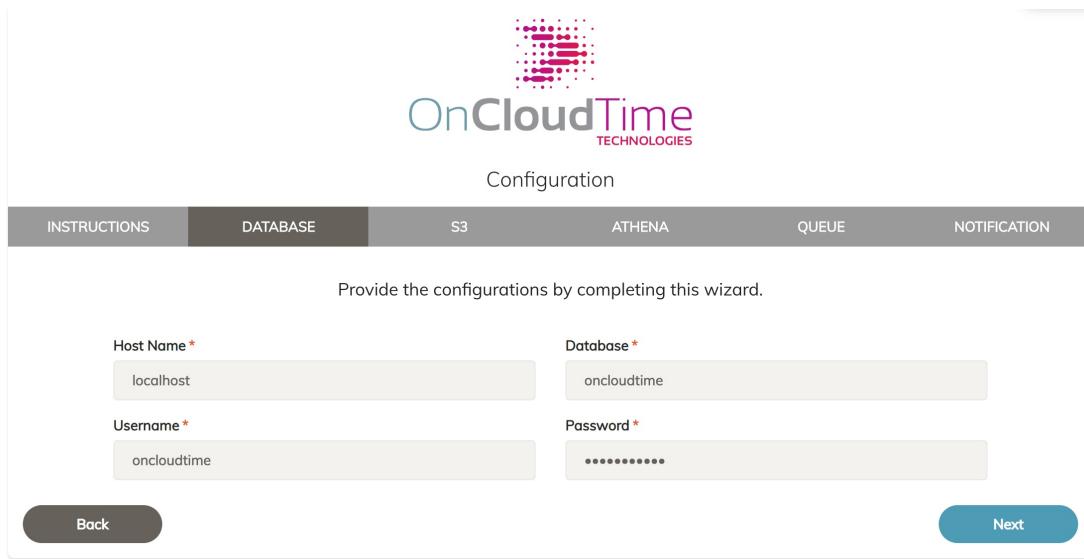


The screenshot shows the first step of the OnCloudTime Configuration Wizard. The title is "Configuration". At the top, there is a navigation bar with tabs: INSTRUCTIONS (which is selected and highlighted in grey), DATABASE, S3, ATHENA, QUEUE, and NOTIFICATION. Below the tabs, a message reads: "Please read the instructions carefully before proceeding." Underneath this message, there is a list of three steps for successful installation:

1. Create S3, SQS, Athena services and SNS topic.
2. Create database and import the sql file provided in sql directory.
3. Provide the details asked in forthcoming steps respectively for successful installation.

At the bottom right of the form is a blue "Next" button.

- o Database Setup for MySQL backend



The screenshot shows the second step of the OnCloudTime Configuration Wizard. The title is "Configuration". The INSTRUCTIONS tab is still selected. Below the tabs, a message reads: "Provide the configurations by completing this wizard." The form contains four input fields arranged in a grid:

<b>Host Name *</b> localhost	<b>Database *</b> oncloudtime
<b>Username *</b> oncloudtime	<b>Password *</b> *****

At the bottom left is a "Back" button, and at the bottom right is a blue "Next" button.

- o S3 configurations



Configuration

INSTRUCTIONS	DATABASE	S3	ATHENA	QUEUE	NOTIFICATION
--------------	----------	----	--------	-------	--------------

Provide the configurations by completing this wizard.

**Key \***

**Secret \***

**Bucket \***

**Back**

**Next**

© 2018 OnCloudTime. All Rights Reserved

- o Athena database configuration



Configuration

INSTRUCTIONS	DATABASE	ATHENA	QUEUE	NOTIFICATION
--------------	----------	--------	-------	--------------

Provide the configurations by completing this wizard.

Please start the athena service before completing this wizard in AWS.

**Athena Directory \***

**Database \***

**Back**

**Next**

© 2018 OnCloudTime. All Rights Reserved

- o SQS Queue configuration (*see placeholder for the format*)



Configuration

INSTRUCTIONS	DATABASE	S3	ATHENA	QUEUE	NOTIFICATION
--------------	----------	----	--------	-------	--------------

Provide the configurations by completing this wizard.

Get the queue url from created sqs queue in AWS

**Queue URL \***

`https://sqs.us-east-1.amazonaws.com/123456789012/oncloudtime`

**Back**

**Next**

© 2018 OnCloudTime. All Rights Reserved

- SNS Notification configuration (see placeholder for the format)

The screenshot shows a web-based configuration wizard for OnCloudTime Technologies. At the top, there's a logo consisting of a grid of dots above the text "OnCloudTime TECHNOLOGIES". Below the logo, the word "Configuration" is centered. A navigation bar at the top has tabs labeled INSTRUCTIONS, DATABASE, S3, ATHENA, QUEUE, and NOTIFICATION, with "NOTIFICATION" being the active tab. A main content area contains the following text: "Provide the configurations by completing this wizard." and "Get the ARN for the default SNS Topic AWS". Below this, there's a form field labeled "Topic ARN \*". The placeholder text in the input field is "arn:aws:sns:us-east-1:123456789012:oncloudtime-global". At the bottom of the page, there are "Back" and "Finish" buttons, and a copyright notice: "© 2018 OnCloudTime. All Rights Reserved".

- Successful installation

The screenshot shows a confirmation message from the OnCloudTime Configuration wizard. It features the company logo at the top. The main message is "Installation Successful". Below the message is a blue button labeled "Continue to login page". At the bottom of the page, there is a copyright notice: "© 2018 OnCloudTime. All Rights Reserved".

- Once the installer runs, the wizard is locked to prevent further changes. When accessed the message below is shown:



## 1.2.4 Post-install Configuration

The post install configurations can be made by ending the config.yaml file in the config folder. The parameter values need to be valid YAML text in double quotes.

The parameters are named in such a way that they are intuitive

## 1.3 User Features and Workflows

This section illustrates the different aspects of the application using the following sample information:

- S3 Bucket: oncloudtime-demo
- Account: Rutrum Industries
- Super User: System Admin
- Account Admin: Jean Black (username: jean-black-6eb865f7-d5d6-4f04-bf6d-4a0ee05a8081)
- User: Frank Morris (username: frank-morris-5df79a88-b73d-47c6-b482-eb681f3c9b4d)

### 1.3.1 Deployment Options

The following deployment options are available:

1. Agency - has access to the Super User role and can create multiple distinct accounts whose data is maintained independently. The Super User can access and manage the data across all accounts
2. Company - only has access to the company's account created by a Super User
3. AWS Marketplace - coming soon

### 1.3.2 Directory Structure

This section describes the directory structure on how data is stored which is illustrated in the image below:



#### NOTES

1. The root folders at the bucket level are `data` and `trash`
2. The `trash` folder contains a mirror of any content deleted from `data`
3. Each account has its own folder i.e., `RutrumIndustries`, `AcmeCorp`, `AccountFolder`
4. Under each account folder is a `home` folder containing user specific data, and a `shared` folder containing resources that are accessible by all users within the account
5. The user folder is of the form `firstname-lastname-uuid` for uniqueness

6. Each module within the application has a folder under the user folder, and in the shared folder if resource sharing is needed

### 1.3.3 Notifications

1. For each action that is carried out, a notification popup is displayed both at the start of the action and at the end of the action in the top right hand corner. The color coding for the notifications are:
  - o Action Start: Blue
  - o Successful Completion: Green
  - o Error occurs during execution: Red
2. The last 10 notifications are always displayed on the bell icon just before the user's name which can be displayed by clicking on it as displayed below



3. The red number is the count of unread notifications by the user
4. If a notification is selected in the drop-down from the bell icon, the counter is reduced by 1 since it only displays the unread notifications and the notification is tagged with an open envelope as displayed below



### 1.3.4 Command Logging

All command execution information is logged into an SQS Queue which is defined at the installation stage before using the application

### 1.3.5 Setup and Application Configuration

These are found under the Setup menu and are visible only to users with the `ROLE_ADMIN` and `ROLE_SUPER_ADMIN`

### 1.3.6 Reset Data

An administrator can reset the data in an account which deletes all defined data while maintaining the user accounts

1. Click the `Reset Data` link under the `Setup Menu`

**Reset Database**

Click the button below to reset all the data. The following will occur:

- Data entered for File Watcher, Hadoop, MPP, Queries, Simulator, Templates will be deleted
- Created tables from the athena database will be deleted
- Contents of user folders will be deleted
- Contents of shared folder will be deleted

**Reset Data**

2. Click **Yes** in the confirmation popup

**Reset Database Output**

Are you sure you want to reset the data in this account?

**Yes**    **No**

Click the button below to reset all the data. The following will occur:

- Data entered for File Watcher, Hadoop, MPP, Queries, Simulator, Templates will be deleted
- Created tables from the athena database will be deleted
- Contents of user folders will be deleted
- Contents of shared folder will be deleted

**Reset Data**

3. A progress indicator runs until the reset process is completed

**Reset Database Output**

Click the button below to reset all the data. The following will occur:

- Data entered for File Watcher, Hadoop, MPP, Queries, Simulator, Templates will be deleted
- Created tables from the athena database will be deleted
- Contents of user folders will be deleted
- Contents of shared folder will be deleted

**Reset Data**

4. The logs from the reset activity are displayed

**Reset Application Data**

- Delete file watcher  
Warning: Using a password on the command line interface can be insecure.
- Delete hadoop  
Warning: Using a password on the command line interface can be insecure.
- Delete MPP  
Warning: Using a password on the command line interface can be insecure.
- Delete queries  
Warning: Using a password on the command line interface can be insecure.
- Delete simulator  
Warning: Using a password on the command line interface can be insecure.
- Delete fileuploads  
Warning: Using a password on the command line interface can be insecure.  
"State": "SUCCEEDED",  
Table coreparquetinvoices\_u deleted from oncloudtime  
"State": "SUCCEEDED",  
Table coreparquetinvoices\_u\_text deleted from oncloudtime  
"State": "SUCCEEDED",  
Table coreparquetinvoices deleted from oncloudtime  
"State": "SUCCEEDED",  
Table coreparquetinvoices\_text deleted from oncloudtime  
"State": "SUCCEEDED",  
Table coretextrivvoices\_u deleted from oncloudtime  
"State": "SUCCEEDED",  
Table coretextrivvoices\_u\_text deleted from oncloudtime  
"State": "SUCCEEDED",  
"String": "SIICFFDFD"

## 1.4 User and Account Management

### 1.4.1 User Roles

Access to the different features within the application depend on the roles that the user has access to. A complete list of roles is below:

1. ROLE\_SUPER\_ADMIN
2. ROLE\_ADMIN
3. ROLE\_CONVERSIONS
4. ROLE\_DATABASE
5. ROLE\_DELETE\_SHARED\_DATA
6. ROLE\_DELETE\_USER\_DATA
7. ROLE\_FILE\_UPLOAD
8. ROLE\_FILE\_WATCHER
9. ROLE\_HADOOP
10. ROLE\_MODEL
11. ROLE\_QUERIES
12. ROLE\_RECOVER\_SHARED\_DATA
13. ROLE\_RECOVER\_USER\_DATA
14. ROLE\_SIMULATION
15. ROLE\_TABLES
16. ROLE\_TEMPLATE
17. ROLE\_USER

### 1.4.2 Hierarchy of User Roles

The hierarchy of user accounts is based on the roles that the accounts have in order to access advanced features:

1. ROLE\_SUPER\_ADMIN - can create new accounts and users. Only accessible in Agency and AWS Marketplace deployments
2. ROLE\_ADMIN - has access to administrative actions on an account plus access to data for all users in the account within the account
3. ROLE\_USER - has access to resources they create in an account and any shared resources by the account admin

### 1.4.3 User Access Restrictions

1. Each user is tied to one account
2. A user can only see their data and any shared data by the account administrator
3. An account administrator can see the data for all the users in the account, copy files into shared folders

### 1.4.4 Login and Access Control

1. Each user requires an email address and password to access the application
2. A user can only access their own data and shared data within the account to which they are associated
3. A user can only access features depending on the roles assigned to them by the account administrator

## 1.4.5 Account Management

This is done by a user with the `ROLE_SUPER_ADMIN`

4. The minimum information needed is:
  - Name
  - Description
  - admin account email address and password
  - Links to information displayed on the dashboard, current and historical activity pages
5. When an account is created the following happens:
  - folder with account name is created in S3 at the path `s3://oncloudtime-demo/data/RutrumIndustries/`
  - The administrator account is created
  - An SNS topic is created for the account to which all account users are subscribed.

## 1.4.6 User Management

This is done by the account administrator

1. The minimum information required for this is:
  - First Name
  - Last Name
  - Email address (cannot be duplicated)
  - Roles granting access to different features within the application
2. When an account is created the following happens:
  - A folder in the account directory, is created at the following path `s3://oncloudtime-demo/data/RutrumIndustries/home/firstname-lastname-uuid`
  - An SNS topic is created for the user to receive notifications
  - The user email address is subscribed to both the application and account SNS topics

## 1.4.7 Change User Password

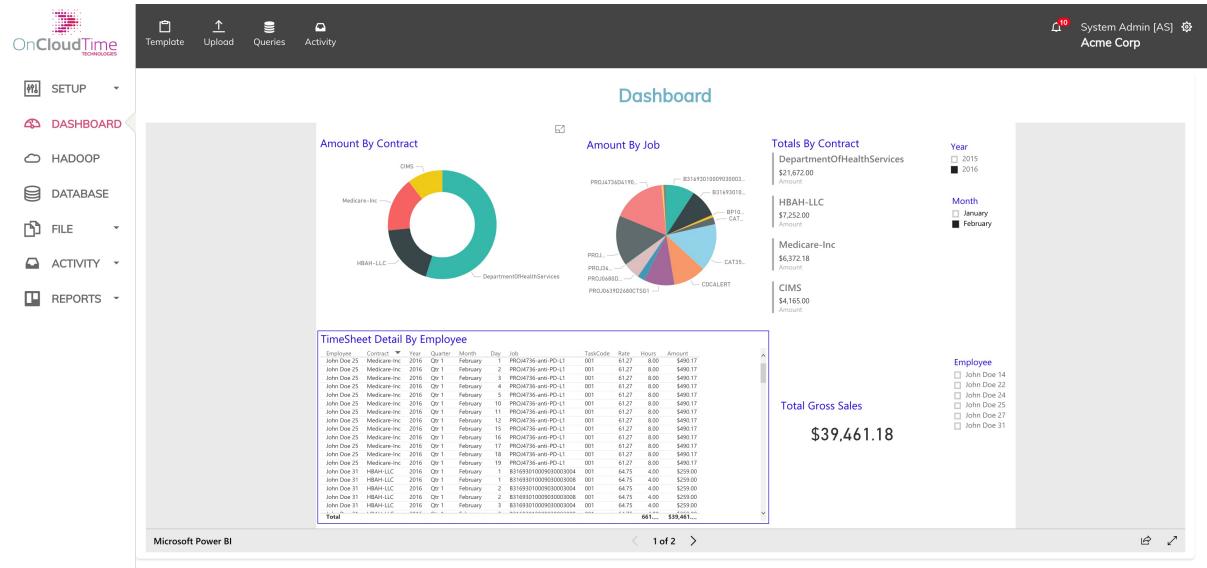
1. Click the Settings (gear icon) just after the user name
2. Select My Profile
3. Click the `Change Password` button
4. Enter the old password, and the new password then click `Submit`

## 1.4.8 Update user profile information

1. Click the Settings (gear icon) just after the user name
2. Select My Profile
3. Click the `Edit Profile` button
4. Update the profile then click `Submit`

## 1.5 Dashboard

Displays data specified by a link in the account configuration. The default display is shown below:



The link can be changed within the account configuration

## 1.6 Hadoop

A user can create and run Hadoop configurations.

### 1.6.1 Create a new Hadoop Configuration

1. Click the Hadoop or + (in the right hand side) to open a popup window

The screenshot shows the OnCloudTime Hadoop configuration page. The top navigation bar includes links for Template, Upload, Queries, and Activity, along with account information for Frank Morris [AS] at Rutrum Industries. The main content area has a search bar and a table titled 'Hadoop'. The table has columns for Actions, Module, Description, Category, Status, Comments, and Timestamp. A note at the top of the table says 'No data available in table'. On the right side of the table, there are '+' and 'x' icons for creating or deleting configurations.

2. Enter the Hadoop configuration information - number of nodes and category

**Create New Hadoop**

Nodes \*

Category \*

\* Required fields

**Close** **Create**

3. As the configuration is being created (which may take some seconds) the `Create` button is disabled

**Create New Hadoop**

Nodes \*

Category \*

\* Required fields

**Close**

4. A new configuration is created with the status `Created`

The screenshot shows the OnCloudTime application interface. The top navigation bar includes links for Template, Upload, Queries, and Activity, along with user information for Frank Morris [AS] from Rutrum Industries. The left sidebar has links for Dashboard, Hadoop (which is currently selected), Database, File, and Activity. The main content area is titled "Hadoop" and contains a table with the following data:

Actions	Module	Description	Category	Status	Comments	Timestamp
>	test_mod	test_desc	invoice_category	Created	test_comments	2018-12-12 12:18:50

## 1.6.2 Start a Hadoop Configuration

1. Click the arrow next to the instance to be started, and a notification will be displayed as below

The screenshot shows a list of configurations under the 'Hadoop' section. One entry is visible:

Actions	Module	Description	Category	Status	Comments	Timestamp
<a href="#">test_mod</a>	test_desc	invoice_category	<span style="color: orange;">● Created</span>		test_comments	2018-12-12 12:18:50

- When the hadoop instance startup is completed, the status changes to **Started** and notification is displayed as below

The screenshot shows the same configuration entry, but the status is now 'Started' (indicated by a green dot) and there is a green notification bar at the top:

Hadoop invoice\_category started at: Wed Dec 12 2018 12:32:34 GMT+0300 (East Africa Time)

### 1.6.3 Stop a Hadoop Configuration

- Click the box next to the instance to be started, and a prompt to stop the Hadoop instance will appear as below. Click Yes to continue

A modal dialog box titled 'Stop Hadoop' is displayed, asking 'Are you sure you want to stop the Hadoop instance?'. There are 'Yes' and 'No' buttons.

- When the hadoop instance instance is stopped is completed, the status changes to **Stopped** as below

The screenshot shows the configuration entry with the status changed to 'Stopped' (indicated by a red dot). The timestamp is updated to 2018-12-12 12:28:19.

## 1.7 Database

A user can create and run MPP database configurations

## 1.8 File Management

The file management features allow the definition of metadata to manage and query files for data

The sample templates that will be used in this section will be the following:

- Core Text Invoices (format Text and type Core)
- User Text Invoices (format Text and type User)

- Core ORC Invoices (format ORC and type Core)
- User ORC Invoices (format ORC and type User)
- Core Parquet Invoices (format Parquet and type Core)
- User Parquet Invoices (format Parquet and type User)

## 1.8.1 Templates

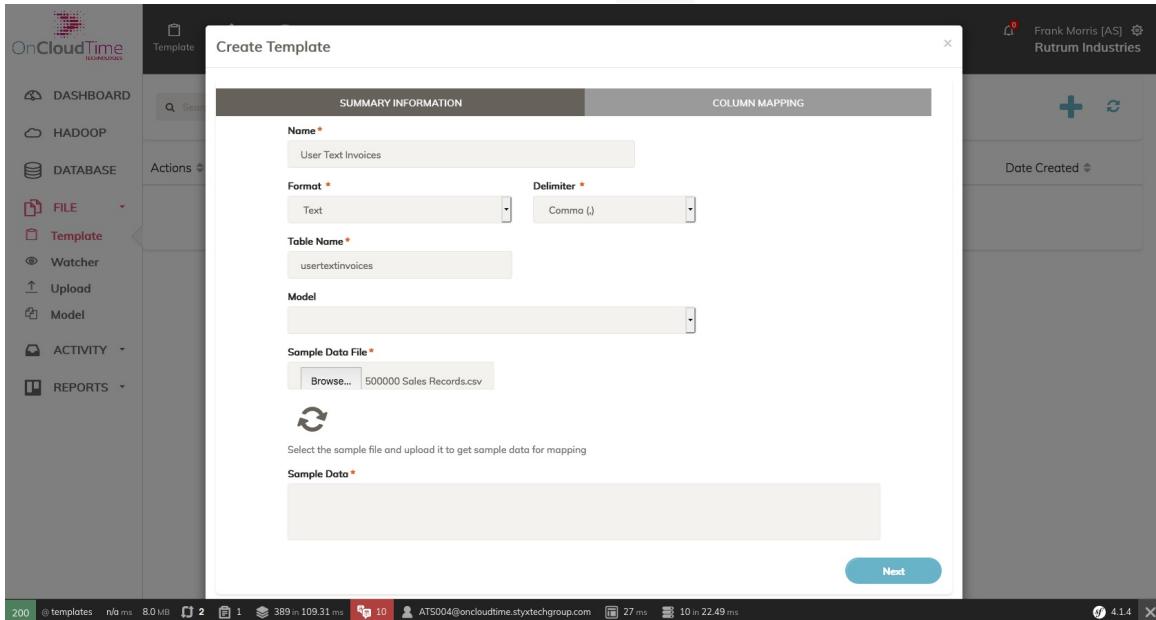
Templates are used to store metadata for processing of files

### 1.8.1.1 Create User Text Template

A user template is created by a user or an administrator and it is not available for other users.

The steps for creating a user template for Text format is as follows:

1. Enter the name of the template `User Text Invoices`
2. Select the Text format, and comma delimiter
3. Enter the table name `usertextinvoices`
4. Select the sample data file from the computer, and click the `Upload` button



5. This loads a sample row as below

## Create Template

**SUMMARY INFORMATION**

Name *	User Text Invoices		
Format *	Text	Delimiter *	Comma (,)
Table Name *	usertextinvoices		
Model			
Sample Data File *	Browse...	500000 Sales Records.csv	Upload
Select the sample file and upload it to get sample data for mapping			
Sample Data *	Region,Country,Item Type,Sales Channel,Order Priority,Order Date,Order ID,Ship Date,Units Sold,Unit Price,Unit Cost,Total Revenue,Total Cost,Total Profit Sub-Saharan Africa,South Africa,Fruits,Offline,M,7/27/2012,443368995,7/28/2012,1593,9.33,6.92,14862.69,11023.56,3839.13		
<b>Next</b>			

- Click **Next** to show the sample data mapping to columns and conversions as below:
  - Source - the name of the column from which the data comes from
  - Sample Data - a sample value for the column
  - Data Type - the type of the data in the column. Defaults to string with other possible values being `Double`, `Date` and `Int`. These match to allowable values in Athena tables that will be created
  - Processing - the conversion that will be carried on the column, defaults to `None`, with additional options such as `Date Processing`, `Zip Lookup`, `Ignore`
  - Parameters - for the conversion function
  - Target - the target column name for the field
  - Ordering - the order of the column in the table

## Create Template

**SUMMARY INFORMATION**

Source	Sample Data	Data Type	Processing	Parameters	Target	Ordering
Region	Sub-Saharan Africa	String	None		region	0
Country	South Africa	String	None		country	1
Item Type	Fruits	String	None		itemtype	2
Sales Channel	Offline	String	None		saleschannel	3

< 1 2 3 4 >

**Back**

**Finish**

- Click the **Finish** button to create the template

- The following are created:

- o External table `usertextinvoices_u` is created in Athena whose input location is `s3://oncloudtime-demo/data/RutrumIndustries/home/frank-morris-5df79a88-b73d-47c6-b482-eb681f3c9b4d/intake/template/User_Text_Invoices/Text/`
- o A query with the code `SELECT * FROM usertextinvoices_u LIMIT 100`

**NOTES:**

1. All the column mapping information is aggregated into a JSON string that is saved to the database

### 1.8.1.2 Create Core Text Template

A core template is created by a user with administrator privileges and its available for all users in the account as it points to a shared folder.

The steps for creating a user template for Text format is as follows:

1. Enter the name of the template `Core Text Invoices`
2. Select the Text format, and comma delimiter and the type `core` (this is only available to administrators)
3. Enter the table name `coretextinvoices`
4. Select the sample data file from the computer, and click the `Upload` button

5. This loads a smaple row as below

## Create Template

**SUMMARY INFORMATION**

**Name \***  
Core Text Invoices

**Format \*** Text      **Delimiter \*** Comma (,)

**Table Name \*** coretextinvoices      **Type \*** Core

**Model**

**Sample Data File \***  
Browse... 500000 Sales Records.csv      **Upload**

Select the sample file and upload it to get sample data for mapping

**Sample Data \***

```
Region,Country,Item Type,Sales Channel,Order Priority,Order Date,Order ID,Ship Date,Units Sold,Unit Price,Unit Cost,Total Revenue,Total Cost,Total Profit
Sub-Saharan Africa,South Africa,Fruits,Offline,M,7/27/2012,443368995,7/28/2012,1593,9.33,6.92,14862.69,11023.56,3839.13
```

**Next**

- Click **Next** to show the sample data mapping to columns and conversions as below:
  - Source - the name of the column from which the data comes from
  - Sample Data - a sample value for the column
  - Data Type - the type of the data in the column. Defaults to string with other possible values being `Double`, `Date` and `Int`. These match to allowable values in Athena tables that will be created
  - Processing - the conversion that will be carried on the column, defaults to `None`, with additional options such as `Date Processing`, `Zip Lookup`, `Ignore`
  - Parameters - for the conversion function
  - Target - the target column name for the field
  - Ordering - the order of the column in the table

SUMMARY INFORMATION			COLUMN MAPPING			
Source	Sample Data	Data Type	Processing	Parameters	Target	Ordering
Region	Sub-Saharan Africa	String	None		region	0
Country	South Africa	String	None		country	1
Item Type	Fruits	String	None		itemtype	2
Sales Channel	Offline	String	None		saleschannel	3

< 1 2 3 4 >

**Back**      **Finish**

- Click the **Finish** button to create the template
- The following are created:

- o User:
  - External Athena table for user `coretextinvoices_u` with input location `s3://oncloudtime-demo/data/RutrumIndustries/home/jean-black-6eb865f7-d5d6-4f04-bf6d-4a0ee05a8081/intake/template/Core Text Invoices/Text`
  - A query with the code `SELECT * FROM coretextinvoices_u LIMIT 100`
- o Shared
  - External shared Athena table `coretextinvoices` with input location `s3://oncloudtime-demo/data/RutrumIndustries/shared/template/Core Text Invoices/Text`
  - A query with the code `SELECT * FROM coretextinvoices LIMIT 100`

### 1.8.1.3 Create User Parquet Template

A user template is created by a user or an administrator and it is not available for other users.

The steps for creating a user template for `Parquet` or nother non-text format is as follows:

1. Enter the name of the template `User Parquet Invoices`
2. Select the Text format, and comma delimiter
3. Enter the table name `userparquetinvoices`
4. Select the sample data file from the computer, and click the `Upload` button

5. This loads a smaple row as below

## Create Template

**SUMMARY INFORMATION**

**Name \***  
User Parquet Invoices

**Format \***  
Parquet

**Delimiter \***  
Comma (,)

**Table Name \***  
userparquetinvoices

**Model**

**Sample Data File \***  
Browse... 500000 Sales Records.csv    Upload

Select the sample file and upload it to get sample data for mapping

**Sample Data \***

```
Region,Country,Item Type,Sales Channel,Order Priority,Order Date,Order ID,Ship Date,Units Sold,Unit Price,Unit Cost,Total Revenue,Total Cost,Total Profit
Sub-Saharan Africa,South Africa,Fruits,Offline,M,7/27/2012,443368995,7/28/2012,1593,9.33,6.92,14862.69,11023.56,3839.13
```

**Next**

6. Click `Next` to show the sample data mapping to columns and conversions as below:
  - o Source - the name of the column from which the data comes from
  - o Sample Data - a sample value for the column
  - o Data Type - the type of the data in the column. Defaults to string with other possible values being `Double`, `Date` and `Int`. These match to allowable values in Athena tables that will be created
  - o Processing - the conversion that will be carried on the column, defaults to `None`, with additional options such as `Date Processing`, `Zip Lookup`, `Ignore`
  - o Parameters - for the conversion function
  - o Target - the target column name for the field
  - o Ordering - the order of the column in the table
7. Click the `Finish` button to create the template
8. The following are created:
  - o External Athena table `userparquetinvoices_u` is created with input location is `s3://oncloudtime-demo/data/RutrumIndustries/home/frank-morris-5df79a88-b73d-47c6-b482-eb681f3c9b4d/intake/template/User Parquet Invoices/Parquet/`
  - o External Text Athena table `userparquetinvoices_u_text` is created with input location is `s3://oncloudtime-demo/data/RutrumIndustries/home/frank-morris-5df79a88-b73d-47c6-b482-eb681f3c9b4d/intake/template/User Parquet Invoices/Text` used to transform the text file uploads into the columnar Parquet format
  - o A query with the code `SELECT * FROM userparquetinvoices_u LIMIT 100`

**NOTES:**

1. All the column mapping information is aggregated into a JSON string that is saved to the database
2. The JSON and ORC format templates of type user behave similarly

**1.8.1.4 Create Core Parquet Template**

A core template is created by a user with administrator privileges and its available for all users in the account as it points to a shared folder.

The steps for creating a user template for Text format is as follows:

1. Enter the name of the template `Core Parquet Invoices`
2. Select the Text format, and comma delimiter and the type `core` (this is only available to administrators)
3. Enter the table name `coreparquetinvoices`
4. Select the sample data file from the computer, and click the `Upload` button

**Create Template**

**SUMMARY INFORMATION**

**COLUMN MAPPING**

**Name \***  
Core Parquet Invoices

**Format \*** Parquet      **Delimiter \*** Comma (,)

**Table Name \*** coreparquetinvoices      **Type \*** Core

**Model**

**Sample Data File \***  
Browse... 500000 Sales Records.csv      **Upload**

Select the sample file and upload it to get sample data for mapping

**Sample Data \***

```
Region,Country,Item Type,Sales Channel,Order Priority,Order Date,Order ID,Ship Date,Units Sold,Unit Price,Unit Cost,Total Revenue,Total Cost,Total Profit
Sub-Saharan Africa,South Africa,Fruits,Offline,M,7/27/2012,443368995,7/28/2012,1593,9.33,6.92,14862.69,11023.56,3839.13
```

**Next**

5. This loads a sample row as below

## Create Template

**SUMMARY INFORMATION**

**Name \***  
Core Parquet Invoices

**Format \*** Parquet      **Delimiter \*** Comma (,)

**Table Name \*** coreparquetinvoices      **Type \*** Core

**Model**

**Sample Data File \***  
Browse... 500000 Sales Records.csv      **Upload**

Select the sample file and upload it to get sample data for mapping

**Sample Data \***

```
Region,Country,Item Type,Sales Channel,Order Priority,Order Date,Order ID,Ship Date,Units Sold,Unit Price,Unit Cost,Total Revenue,Total Cost,Total Profit
Sub-Saharan Africa,South Africa,Fruits,Offline,M,7/27/2012,443368995,7/28/2012,1593,9.33,6.92,14862.69,11023.56,3839.13
```

**Next**

6. Click `Next` to show the sample data mapping to columns and conversions as below:
  - o Source - the name of the column from which the data comes from
  - o Sample Data - a sample value for the column
  - o Data Type - the type of the data in the column. Defaults to string with other possible values being `Double`, `Date` and `Int`. These match to allowable values in Athena tables that will be created
  - o Processing - the conversion that will be carried on the column, defaults to `None`, with additional options such as `Date Processing`, `Zip Lookup`, `Ignore`
  - o Parameters - for the conversion function
  - o Target - the target column name for the field
  - o Ordering - the order of the column in the table
7. Click the `Finish` button to create the template
8. The following are created:
  - o User:
    - External Athena table for user `coreparquetinvoices_u` with input location `s3://oncloudtime-demo/data/RutrumIndustries/home/jean-black-6eb865f7-d5d6-4f04-bf6d-4a0ee05a8081/intake/template/Core Parquet Invoices/Parquet` for the admin user
    - External Text Athena table `coreparquetinvoices_u_text` is created with input location is `s3://oncloudtime-demo/data/RutrumIndustries/home/jean-black-6eb865f7-d5d6-4f04-bf6d-4a0ee05a8081/intake/template/Core Parquet Invoices/Text` used to transform the text file uploads into the columnar Parquet format
    - A query with the code `SELECT * FROM coreparquetinvoices_u LIMIT 100`
  - o Shared
    - External shared Athena table `coreparquetinvoices` with input location `s3://oncloudtime-demo/data/RutrumIndustries/shared/template/Core Parquet Invoices/Parquet`
    - External shared Text Athena table `coreparquetinvoices_text` with input location `s3://oncloudtime-demo/data/RutrumIndustries/shared/template/Core Parquet Invoices/Text`

- A query with the code `SELECT * FROM coreparquetinvoices LIMIT 100`

### 1.8.1.5 Sync Template

When a template is updated, the sync template updates the mappings for the columns within the external tables

## 1.8.2 Upload

This feature allows a user to upload files for processing. Each file that is uploaded is attached to a template that defines its format and metadata.

### 1.8.2.1 Upload File to Text Template

1. Select the file to upload and the template to which the file is to be assigned

The screenshot shows a modal dialog titled "Upload Files". It contains fields for selecting a file ("Choose File") and choosing a template ("Template"). The file selected is "1000 Sales Records copy 2.csv" and the template chosen is "User Text Invoices". A note at the bottom indicates that the file and template selection are required. At the bottom right are "Close" and "Upload" buttons.

Choose File \*

Choose File 1000 Sales Records copy 2.csv

Template \*

User Text Invoices

\* Required fields

Close      Upload

2. The progress of file upload and processing is shown by a spinner

## Upload Files

X

Choose File \*

Choose File 1000 Sales Records copy 2.csv

Template \*

User Text Invoices

\* Required fields

Close



3. A list of files is shown once the upload is complete, the file has a status Pending

The screenshot shows the OnCloudTime interface. The top navigation bar includes 'Template', 'Upload', 'Queries', and 'Activity'. On the right, there are notifications for 10 messages and a user 'Frank Morris [AS] @Rutrum Industries'. The left sidebar has a 'FILE' section with 'Upload' selected. The main area is titled 'Upload' and shows a table with one row:

Action	Template	Folder	Name	Size	Status	Date Created
	User Text Invoices	User	User Text Invoices.1000salesrecordscopy2.csv	122.1 KB	Pending	2018-12-13 18:13:23

4. The file is uploaded to the following path s3://oncloudtime-demo/data/RutrumIndustries/home/frank-morris-5df79a88-b73d-47c6-b482-eb681f3c9b4d/intake/template/User Text Invoices/Text/User Text Invoices.1000salesrecordscopy2.csv/User Text Invoices.1000salesrecordscopy2.csv

### 1.8.2.2 Upload File to Parquet Template

1. Select the file to upload and the template to which the file is to be assigned

### Upload Files

Choose File \*

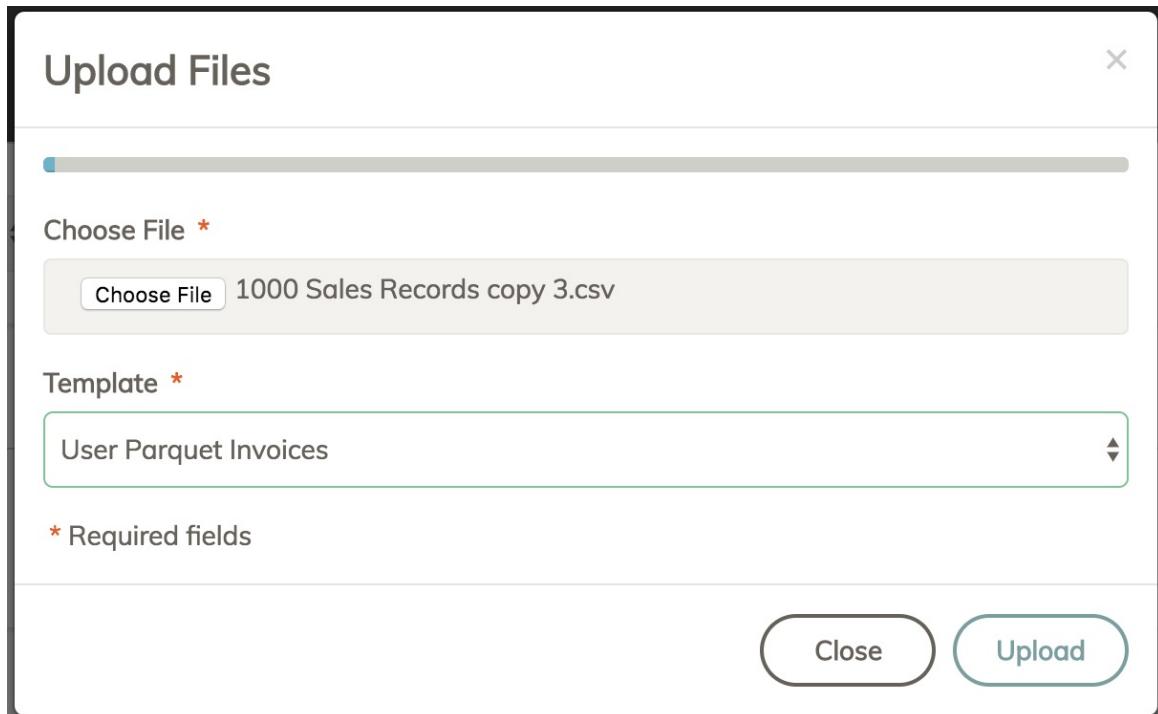
Choose File 1000 Sales Records copy 3.csv

Template \*

User Parquet Invoices

\* Required fields

**Close** **Upload**



2. The progress of file upload and processing is shown by a spinner

### Upload Files

Choose File \*

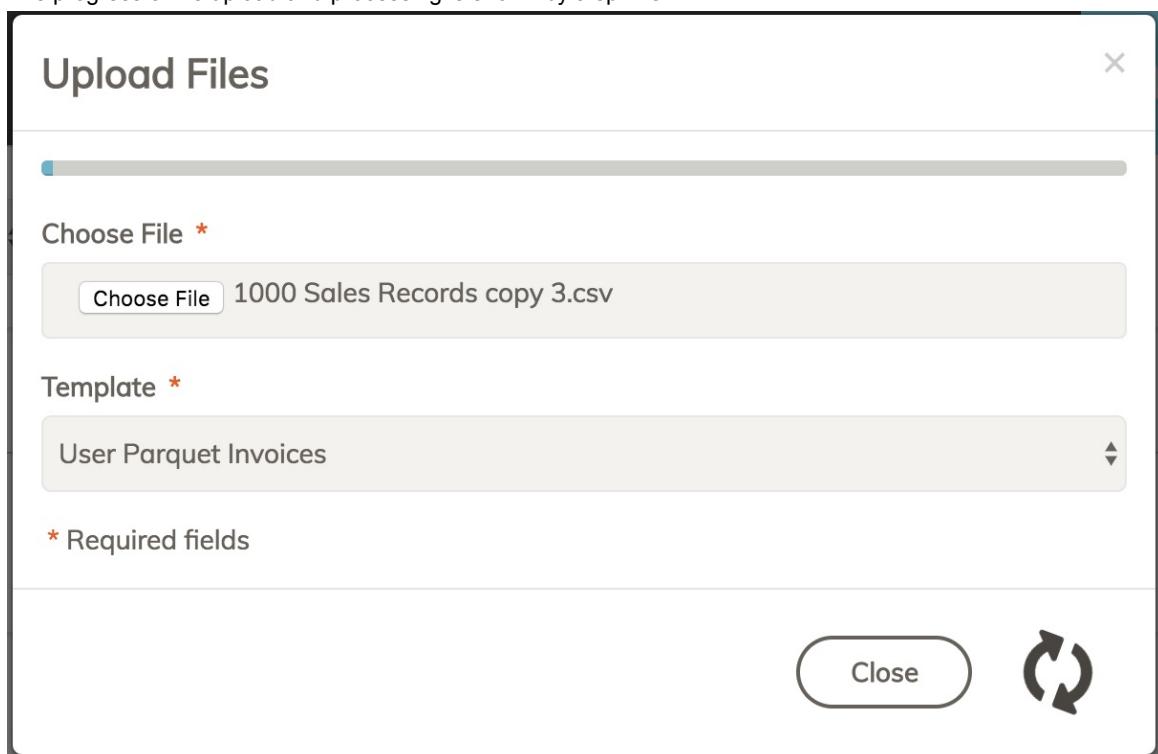
Choose File 1000 Sales Records copy 3.csv

Template \*

User Parquet Invoices

\* Required fields

**Close** 



3. A list of files is shown once the upload is complete, the file has a status Pending

The screenshot shows the OnCloudTime Technologies platform's interface. The main navigation bar at the top includes 'Template', 'Upload', 'Queries', and 'Activity'. The top right corner shows a user profile for 'Frank Morris [AS]' from 'Rutrum Industries' with a notification count of 10. The left sidebar has a 'FILE' section expanded, with 'Upload' selected. The main content area displays an 'Upload' list with two entries:

Action	Template	Folder	Name	Size	Status	Date Created
	User Text Invoices	User	User Text Invoices.1000salesrecordscopy2.csv	122.1 KB	Pending	2018-12-13 18:13:23
	User Parquet Invoices	User	User Parquet Invoices.1000salesrecordscopy3.csv	122.1 KB	Pending	2018-12-13 18:28:14

#### 4. The file is uploaded to the following paths:

- o **Text:** `s3://oncloudtime-demo/data/RutrumIndustries/home/frank-morris-5df79a88-b73d-47c6-b482-eb681f3c9b4d/intake/template/User Parquet Invoices/Text/User Parquet Invoices.1000salesrecordscopy3.csv/User Parquet Invoices.1000salesrecordscopy3.csv`
- o **Parquet Conversion:** `s3://oncloudtime-demo/data/RutrumIndustries/home/frank-morris-5df79a88-b73d-47c6-b482-eb681f3c9b4d/intake/template/User Parquet Invoices/Parquet/User Parquet Invoices.1000salesrecordscopy3.csv/20181213_152800_00005_ny3my_ac5f3650-603b-4c22-bc90-be14a78afade`

### 1.8.2.3 Upload Duplicate file to template

The error below is shown when a user tries to upload the same file to a template

The screenshot shows the 'Upload Files' dialog box. It contains a 'Choose File' input field with the value '1000 Sales Records copy 2.csv' and a 'Template' dropdown set to 'Core Text Invoices'. A red error message at the bottom of the dialog states: '\* Required fields \*'1000 Sales Records copy 2.csv' has already been uploaded to the template 'Core Text Invoices' please select another file'. In the background, the main interface shows a list of uploaded files with status 'Pending'.

### 1.8.2.4 Copy File to Shared Folder

This feature is only available to users with the `ROLE_ADMIN`

The restrictions to files that can be copied are as follows:

- Files already in the shared folder cannot be copied again
- Only files uploaded to templates with the type `core` can be shared, as those templates have a shared external table
- Files that have been deleted and are in the trash folder cannot be copied

The steps to copy a file to a shared folder are:

1. From the list of file uploads select the share ( in the row of the file you want to share

Action	Template	Folder	Name	Size	Status	Date Created	Uploaded By
	Core Text Invoices	User	Core Text Invoices.1000salesrecordscopy2.csv	122.1 KB	Pending	2018-12-14 13:57:10	Jean Black
	Core Parquet Invoices	User	Core Parquet Invoices.1000salesrecordscopy3.csv	122.1 KB	Pending	2018-12-14 13:58:02	Jean Black

2. A popup will appear requesting confirmation of the file copy to a shared folder

Are you sure you want to copy the file 'Core Text Invoices.1000salesrecordscopy2.csv' to the shared folder?

3. The shared file will be displayed in the list of file uploads with folder **Shared**

Action	Template	Folder	Name	Size	Status	Date Created	Uploaded By
	Core Text Invoices	User	Core Text Invoices.1000salesrecordscopy2.csv	122.1 KB	Pending	2018-12-14 13:57:10	Jean Black
	Core Parquet Invoices	User	Core Parquet Invoices.1000salesrecordscopy3.csv	122.1 KB	Pending	2018-12-14 13:58:02	Jean Black
	Core Text Invoices	Shared	Core Text Invoices.1000salesrecordscopy2.csv	122.1 KB	Pending	2018-12-14 14:14:54	Jean Black

### 1.8.2.5 Delete File

The delete action moves a file to the trash folder, and does not permanently remove it.

The restrictions to file deleting are:

- A user can only delete a file they created
- Only users with `ROLE_ADMIN` or `ROLE_DELETE_SHARED_DATA` can delete files in the shared folder
- Files in the trash folder cannot be deleted by any user

The steps to delete a file are as follows:

1. From the list of file uploads select the share () in the row of the file you want to delete

Action	Template	Folder	Name	Size	Status	Date Created	Uploaded By
	Core Text Invoices	User	Core Text Invoices.1000salesrecordscopy2.csv	122.1 KB	Pending	2018-12-14 13:57:10	Jean Black
	Core Parquet Invoices	User	Core Parquet Invoices.1000salesrecordscopy3.csv	122.1 KB	Pending	2018-12-14 13:58:02	Jean Black

2. A popup will appear requesting confirmation to delete the file

The screenshot shows the OnCloudTime interface with a 'Delete Confirmation' dialog box overlaid. The dialog asks, 'Are you sure you want to delete the file 'Core Parquet Invoices.1000salesrecordscopy3.csv'?'. Below the dialog is a table listing files. One file, 'Core Parquet Invoices.1000salesrecordscopy3.csv', is highlighted with a trash icon and has a status of 'Pending'. The table includes columns for Action, Template, Folder, Name, Size, Status, Date Created, and Uploaded By.

3. The deleted file will appear with a folder **Trash** and a status **Trash**

The screenshot shows the OnCloudTime interface after a file has been deleted. The 'Core Parquet Invoices.1000salesrecordscopy3.csv' file is now listed in the 'Trash' folder under the 'FILE' section. It has a red trash icon and a status of 'Trash'. The rest of the table remains the same.

### 1.8.2.6 Recover Deleted File

This will restore a file from trash into the user or shared folder, and is only accessible to users with the following roles:

- `ROLE_ADMIN` for user and shared files
- `ROLE_RECOVER_USER_DATA` for user files
- `ROLE_RECOVER_SHARED_DATA` for shared files

The steps to restore a deleted file are as follows:

1. From the list of file uploads select the share ( ) in the row of the file you want to restore

The screenshot shows the OnCloudTime interface with a 'Delete Confirmation' dialog box overlaid. The dialog asks, 'Are you sure you want to delete the file 'Core Parquet Invoices.1000salesrecordscopy3.csv'?'. Below the dialog is a table listing files. One file, 'Core Parquet Invoices.1000salesrecordscopy3.csv', is highlighted with a trash icon and has a status of 'Pending'. The table includes columns for Action, Template, Folder, Name, Size, Status, Date Created, and Uploaded By.

2. A popup will appear requesting confirmation to delete the file

The screenshot shows the OnCloudTime interface with a 'Delete Confirmation' dialog box overlaid. The dialog asks, 'Are you sure you want to delete the file 'Core Parquet Invoices.1000salesrecordscopy3.csv'?'. Below the dialog is a table listing files. One file, 'Core Parquet Invoices.1000salesrecordscopy3.csv', is highlighted with a trash icon and has a status of 'Pending'. The table includes columns for Action, Template, Folder, Name, Size, Status, Date Created, and Uploaded By.

3. The deleted file will appear with a folder **Trash** and a status **Trash**

The screenshot shows the OnCloudTime Technologies platform interface. The left sidebar has sections for SETUP, DASHBOARD, HADOOP, DATABASE, and FILE. Under FILE, there is a 'Template' item. The main area has tabs for Template, Upload, Queries, and Activity. The Upload tab is active, showing a search bar, a filter by folder dropdown, and an 'Upload' button with a cloud icon. Below this is a table with columns: Action, Template, Folder, Name, Size, Status, Date Created, and Uploaded By. Two rows are listed:

Action	Template	Folder	Name	Size	Status	Date Created	Uploaded By
	Core Text Invoices	User	Core Text Invoices.1000salesrecordscopy2.csv	122.1 KB	Pending	2018-12-14 13:57:10	Jean Black
	Core Parquet Invoices	User	Core Parquet Invoices.1000salesrecordscopy3.csv	122.1 KB	Pending	2018-12-14 13:58:02	Jean Black

## 1.8.3 Watcher

Provides the ability to run scripts

### 1.8.3.1 Create Watcher

1. Click the `Watcher` text or the `+` on the list which will popup a window with information to create a watcher

## Create Watcher

**Title \***

**Type \***

----

**Description \***

**Code \***

**S3 folder**

Leave this field blank for the default folder to be generated

**Security Policy**

**Schedule**

**Model**

**Template**

\* Required fields

**Close** **Create**

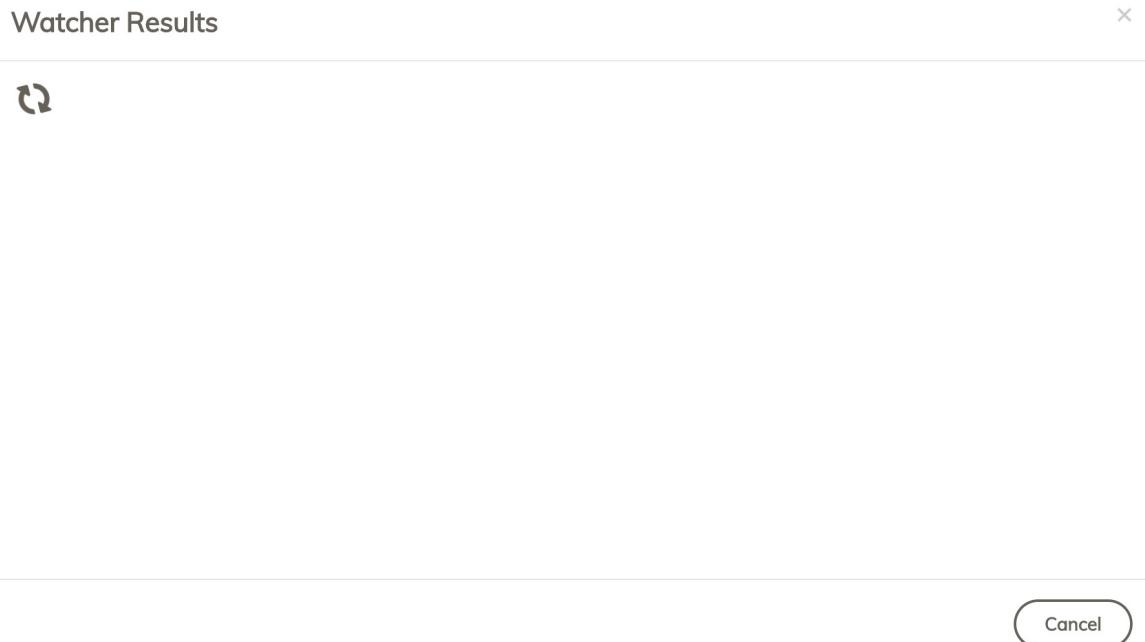
- When you click `create` a new watcher will be created and displayed in the list with a status `Pending`

Action	Title	Type	Folder	Description	Code	Status	Timestamp
	A Watcher	SNS	s3://oncloudtime-dem...	Watcher Description	Execution Code		2018-12-14 16:04:03

### 1.8.3.2 Run Watcher Interactively

A popup window opens and displays the output of the watcher execution. When the window closes the execution is terminated

- Click the run ( ) in the row of the watcher to run interactively that opens up a pop-up window with a progress spinner as below



- When the execution completes the results are displayed in the pop-up window

## Watcher Results

### Run FileWatcher Results

Warning: Using a password on the command line interface can be insecure.

```
uuid afd977d5-9eab-42de-b309-11aed27f05b1
```

```
localhost oncloudtime oncloudtime oncloudtime 1 A Watcher SNS s3://oncloudtime-
demo/data/RutrumIndustries/home/frank-morris-5df79a88-b73d-47c6-b482-eb681f3c9b4d/filewatcher/afd977d5-9eab-
42de-b309-11aed27f05b1/ Watcher Description Execution Code oncloudtime s3://oncloudtime-demo/athena-output/
afd977d5-9eab-42de-b309-11aed27f05b1 us-east-1 oncloudtime-demo
/Users/ssmuso/PhpstormProjects/oncloudtime/var/log/frank-morris-5df79a88-b73d-47c6-b482-eb681f3c9b4d/filewatcher/7890000 frank-
morris-5df79a88-b73d-47c6-b482-eb681f3c9b4d.1544793174.runfilewatcher.afd977d5-9eab-42de-b309-
11aed27f05b1.log oncloudtime-
demo/data/RutrumIndustries/home/frank-morris-5df79a88-b73d-47c6-b482-eb681f3c9b4d/filewatcher/7890000 frank-
morris-5df79a88-b73d-47c6-b482-eb681f3c9b4d.1544793174.runfilewatcher.afd977d5-9eab-42de-b309-
11aed27f05b1.log
```

[Close](#)

- The results of the execution can also be retrieved later by clicking on the value in the status column that opens up a popup window simular to #2 above

Action	Title	Type	Folder	Description	Code	Status	Timestamp
File	A Watcher	SNS	s3://oncloudtime-dem...	Watcher Description	Execution Code	Completed	2018-12-14 16:12:59

### 1.8.3.3 Run Watcher in Background

This is best for long running watchers so that the user can keep working on other tasks

- Click the run (run icon) in the row of the watcher to run in the background that opens up a confirmation prompt as below

Are you sure you want to run the filewatcher 'A Watcher' in the background?

[Yes](#) [No](#)

- The status of the Watcher is set to `Running` on the list

The screenshot shows the OnCloudTime interface with the 'FILE' menu selected. The main area displays a table titled 'Watcher' with columns: Action, Title, Type, Folder, Description, Code, Status, and Timestamp. The first row, titled 'A Watcher', has a status of 'Running'. A small blue circular spinner icon with a white 'i' is located in the 'Status' column next to the 'Running' status.

3. The results of the execution are updated in the background too, so the status can be updated by refreshing the page clicking the spinner( ) in the right hand of the list.

This screenshot shows the same OnCloudTime interface after refreshing. The 'A Watcher' entry now has a status of 'Completed' and the spinner icon is no longer present.

4. Clicking the status opens a popup window that displays the results as below

This screenshot shows a detailed 'Status Result' dialog box overlaid on the OnCloudTime interface. The dialog contains two sections: 'Log File URL' and 'Run FileWatcher Results'. The 'Log File URL' section shows a long URL for an S3 log file. The 'Run FileWatcher Results' section contains a warning about using a password on the command line and displays a large block of command-line output related to the file watcher's execution.

#### 1.8.3.4 Delete Watcher

This removes the watcher from the application

1. Click the delete( ) icon in the row containing the watcher to delete, which opens up a confirmation window

This screenshot shows a 'Delete FileWatcher Confirmation' dialog box with the message 'Are you sure you want to delete the filewatcher 'A Watcher'?'. It has 'Yes' and 'No' buttons.

2. Clicking **Yes** in the confirmation window deletes the watcher and updates the list

## 1.8.4 Model

Provides the ability to run scripts

### 1.8.4.1 Create Model

1. Click the `Model` text or the `+` on the list which will popup a window with information to create a model

**Create Model**

Title \*

Type \*

Description \*

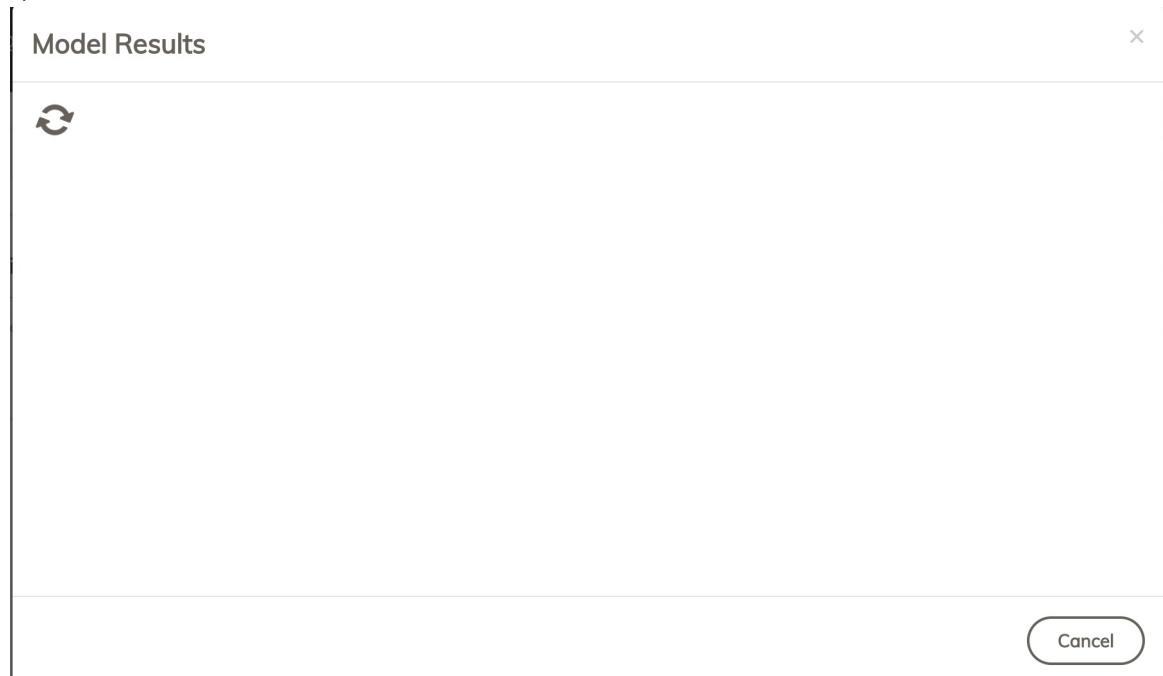
Code \*

2. When you click `Create` a new watcher will be created and displayed in the list with a status `Pending`

### 1.8.4.2 Run Model Interactively

A popup window opens and displays the output of the model execution. When the window closes the execution is terminated

1. Click the run (▶) in the row of the model to run interactively that opens up a pop-up window with a progress spinner as below



2. When the execution completes the results are displayed in the pop-up window

## Model Results

### Run Model Results

Warning: Using a password on the command line interface can be insecure.

```
uuid 4dea6fa0-86ae-43ba-9fc0-623c49f67320
```

```
localhost oncloudtime oncloudtime oncloudtime 1 A New Model SQS s3://oncloudtime-
demo/data/RutrumIndustries/home/frank-morris-5df79a88-b73d-47c6-b482-eb681f3c9b4d/simulation/4dea6fa0-86ae-
43ba-9fc0-623c49f67320/ Test Model Model Code
```

```
11 - oncloudtime s3://oncloudtime-demo/athena-output/ 4dea6fa0-86ae-43ba-9fc0-623c49f67320
```

```
16 - /Users/ssmusoke/PhpstormProjects/oncloudtime/var/log/frank-morris-5df79a88-b73d-47c6-b482-
eb681f3c9b4d.1544795722.runsimulation.4dea6fa0-86ae-43ba-9fc0-623c49f67320.log oncloudtime-
demo/data/RutrumIndustries/home/frank-morris-5df79a88-b73d-47c6-b482-eb681f3c9b4d/simulation/ 7890000 frank-
morris-5df79a88-b73d-47c6-b482-eb681f3c9b4d.1544795722.runsimulation.4dea6fa0-86ae-43ba-9fc0-
```

[Close](#)

- The results of the execution can also be retrieved later by clicking on the value in the status column that opens up a popup window similar to #2 above

Action	Title	Type	Folder	Description	Code	Status	Timestamp
	A New Model	SQS	s3://oncloudtime-dem...	Test Model	Model Code	<span style="color: green;">● Completed</span>	2018-12-14 16:56:11

### 1.8.4.3 Run Model in Background

This is best for long running models so that the user can keep working on other tasks

- Click the run ( in the row of the model to run in the background that opens up a confirmation prompt as below

Are you sure you want to run the model 'A New Model' in the background?

- The status of the Model is set to Running on the list

Action	Title	Type	Folder	Description	Code	Status	Timestamp
	A New Model	SQS	s3://oncloudtime-dem...	Test Model	Model Code	Running	2018-12-14 16:56:06

3. The results of the execution are updated in the background too, so the status can be updated by refreshing the page clicking the spinner( ) in the right hand of the list.

Action	Title	Type	Folder	Description	Code	Status	Timestamp
	A New Model	SQS	s3://oncloudtime-dem...	Test Model	Model Code	Completed	2018-12-14 16:56:11

4. Clicking the status opens a popup window that displays the results as below

The screenshot shows two overlapping modal windows. The top window is titled "Status Result" and contains a "Log File URL" section with a long URL and an AWS signature. The bottom window is titled "Run Model Results" and contains a warning about using a password on the command line interface, followed by several command-line outputs related to model creation and SQS.

**Status Result**

Log File URL

<https://oncloudtime-demo.s3.amazonaws.com/data/RutrumIndustries/home/frank-morris-5df79a88-b73d-47c6-b482-eb681f3c9b4d/simulation/4dea6fa0-86ae-43ba-9fc0-623c49f67320/frank-morris-5df79a88-b73d-47c6-b482-eb681f3c9b4d.1544795766.runsimulation.4dea6fa0-86ae-43ba-9fc0-623c49f67320.log?AWSAccessKeyId=AKIAJT5BKDAE6FPYCGCA&Signature=Tg0cWrlh%2BVDQTkDZ%2BQsbBxpyj0g%3D&Expires=1552685771>

**Run Model Results**

Warning: Using a password on the command line interface can be insecure.

```
uuid 4dea6fa0-86ae-43ba-9fc0-623c49f67320
localhost oncloudtime oncloudtime oncloudtime 1 A New Model SQS
s3://oncloudtime-demo/data/RutrumIndustries/home/frank-morris-5df79a88-b73d-47c6-b482-eb681f3c9b4d/simulation/4dea6fa0-86ae-43ba-9fc0-623c49f67320/ Test Model Model Code
11 - oncloudtime s3://oncloudtime-demo/athena-output/ 4dea6fa0-86ae-43ba-9fc0-623c49f67320
16 /Users/frank/Downloads/DphantomProjects/oncloudtime/A New Model/Frank Morris
```

#### 1.8.4.4 Delete Model

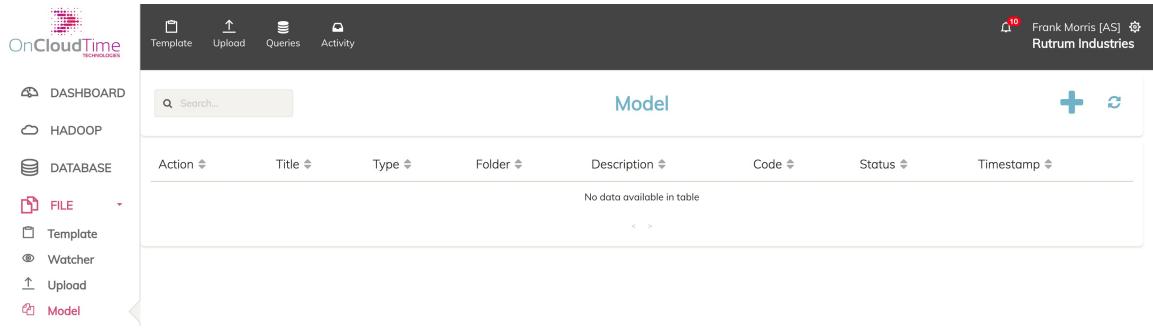
This removes the model from the application

5. Click the delete( ) icon in the row containing the model to delete, which opens up a confirmation window

The screenshot shows the OnCloudTime application interface with a "Delete Model Confirmation" dialog box open. The dialog asks if you're sure you want to delete the model 'A New Model'. Below the dialog is a table listing models, with one row selected for deletion.

Action	Title	Type	Folder	Description	Code	Status	Timestamp
	A New Model	SQS	s3://oncloudtime-dem...	Test Model	Model Code	Running	2018-12-14 16:58:22

6. Clicking **Yes** in the confirmation window deletes the model and updates the list



## 1.9 Timesheet

### 1.9.1 Projects

### 1.9.2 Rates

### 1.9.3 Timesheet Tasks

### 1.9.4 Assignment

When a user is assigned to a project, a default rate is added for the assignment

#### 1.9.4.1 Validation Rules

Project assignments cannot overlap

## 1.9.5 Submission

## 1.10 Tasks

### 1.10.1 Categories

### 1.10.2 Task Queue

### 1.10.3 Task Template

## 1.11 Activity

These are reports whose content is defined at account level by external urls

### 1.11.1 Current

The default view is shown below

## 1.11.2 History

The default view is shown below

## 1.12 Reports

### 1.12.1 Tables

Displays and allows searches for files uploaded to templates

### 1.12.2 Queries

These are queries executed against the data to which the application has access with those having a type Athena are executed against tables in the Athena database defined at installaion.

### 1.12.2.1 Automatically Created Queries

Queries are automatically created along with templates to enable analysis of uploaded files, and these queries are of type `Athena`. A user can also manually create additional queries as needed.

### 1.12.2.2 Create Query

1. Click the `Query` text or the `+` on the list which will popup a window with information to create a query

**Create Query**

Title \*

Type \*

Description \*

Code \*

```
SELECT * FROM test
```

\* Required fields

**Close** **Save** **Cancel**

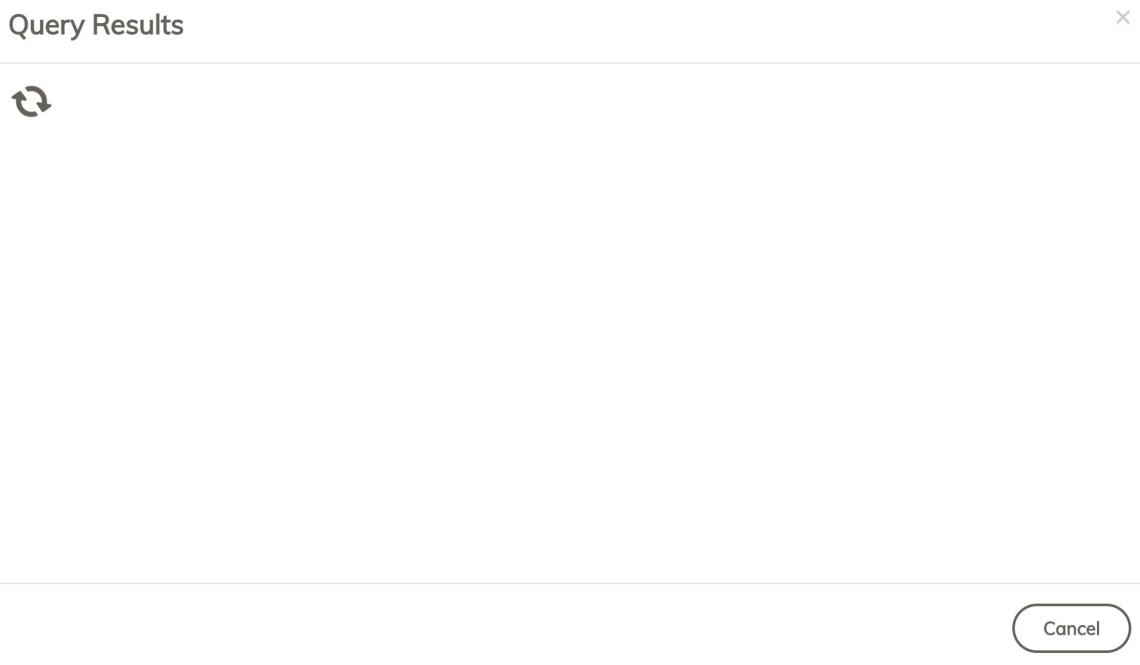
2. When you click `Create` a new query will be created and displayed in the list with a status `Pending`

Action	Title	Type	Folder	Description	Code	Type	Status	Timestamp
	Core Text Invoices	Athena	s3://oncloudtime-dem...	Core Text Invoices	SELECT * FROM corete...	Core	Completed	2018-12-14 19:20:20
	Core Parquet Invoice...	Athena	s3://oncloudtime-dem...	Core Parquet Invoice...	SELECT * FROM corepa...	Core	Pending	2018-12-14 19:13:08
	Sample Query	Hive	s3://oncloudtime-dem...	Query Description	SELECT * FROM test	User	Pending	2018-12-14 19:21:48

### 1.12.2.3 Run Query Interactively

A popup window opens and displays the output of the query execution. When the window closes the execution is terminated

3. Click the run (  ) in the row of the query to run interactively that opens up a pop-up window with a progress spinner as below



4. When the execution completes the results are displayed in the pop-up window



5. The results of the execution can also be retrieved later by clicking on the value in the status column that opens up a popup window simular to #2 above

**NOTE:** The example shown above is for `Athena` type, which runs the query against a created table in Athena. The other types run queries against other data sources, example output shown below

**Query Results**

**Log File URL**

<https://oncloudtime-demo.s3.amazonaws.com/data/RutrumIndustries/home/frank-morris-5df79a88-b73d-47c6-b482-eb681f3c9b4d/query/7308a617-342a-4d90-bae2-20c94452c766/frank-morris-5df79a88-b73d-47c6-b482-eb681f3c9b4d.1544806689.runquery.7308a617-342a-4d90-bae2-20c94452c766.log?AWSAccessKeyId=AKIAJT5BKDAE6FPYCGCA&Signature=Y3nRhIxb09OTzDnxB2MnD3zjQ5o%3D&Expires=1552696693>

**Log File Output**

**Run Query Results**

Warning: Using a password on the command line interface can be insecure.  
uuid 7308a617-342a-4d90-bae2-20c94452c766  
localhost oncloudtime oncloudtime oncloudtime 103 My Test Hive s3://oncloudtime-

**Close**

#### 1.12.2.4 Run Query in Background

This is best for long running queries so that the user can keep working on other tasks

1. Click the run (run icon) in the row of the query to run in the background that opens up a confirmation prompt as below

2. The status of the Query is set to `Running` on the list

Action	Title	Type	Folder	Description	Code	Type	Status	Timestamp
	Core Text Invoices	Athena	s3://oncloudtime-demo...	Core Text Invoices	SELECT * FROM corete...	Core	Running	2018-12-14 19:20:06
	Core Parquet Invoice...	Athena	s3://oncloudtime-demo...	Core Parquet Invoice...	SELECT * FROM corepa...	Core	Pending	2018-12-14 19:13:08

3. The results of the execution are updated in the background too, so the status can be updated by refreshing the page clicking the spinner() in the right hand of the list.

Action	Title	Type	Folder	Description	Code	Type	Status	Timestamp
	Core Text Invoices	Athena	s3://oncloudtime-demo...	Core Text Invoices	SELECT * FROM corete...	Core	Completed	2018-12-14 19:20:20
	Core Parquet Invoice...	Athena	s3://oncloudtime-demo...	Core Parquet Invoice...	SELECT * FROM corepa...	Core	Pending	2018-12-14 19:13:08

4. Clicking the status opens a popup window that displays the results as below

### Query Results

Query Output URL

<https://oncloudtime-demo.s3.amazonaws.com/data/RutrumIndustries/home/frank-morris-5df79a88-b73d-47c6-b482-eb681f3c9b4d/query/0252ed56-69b7-44dc-9d16-c3e64060b9d9/edc56699-7cb6-45b4-8b46-5dc71488e30c.csv?AWSAccessKeyId=AKIAJT5BKDAE6FPYCGCA&Signature=xBiljGQ6Wc3oBjx87R12czzkV0I%3D&Expires=1552696507>

Query Output

region	country	itemtype	saleschannel	orderpriority	orderdate	orderid	shipdate	unitssold
--------	---------	----------	--------------	---------------	-----------	---------	----------	-----------

Log File URL

<https://oncloudtime-demo.s3.amazonaws.com/data/RutrumIndustries/home/frank-morris-5df79a88-b73d-47c6-b482-eb681f3c9b4d/query/0252ed56-69b7-44dc-9d16-c3e64060b9d9/edc56699-7cb6-45b4-8b46-5dc71488e30c.csv?AWSAccessKeyId=AKIAJT5BKDAE6FPYCGCA&Signature=xBiljGQ6Wc3oBjx87R12czzkV0I%3D&Expires=1552696507>

[Close](#)

**NOTE:** The example shown above is for `Athena` type, which runs the query against a created table in Athena. The other types run queries against other data sources, example output shown below

## Status Result

### Log File URL

```
https://oncloudtime-demo.s3.amazonaws.com/data/RutrumIndustries/home/frank-morris-5df79a88-b73d-47c6-b482-eb681f3c9b4d/query/7308a617-342a-4d90-bae2-20c94452c766/frank-morris-5df79a88-b73d-47c6-b482-eb681f3c9b4d.1544806689.runquery.7308a617-342a-4d90-bae2-20c94452c766.log?
AWSAccessKeyId=AKIAJT5BKDAE6FPYCGCA&Signature=Y3nRhIxb09OTzDnxB2MnD3zjQ5o%3D&Expires=1552696693
```

### Log File Output

### Run Query Results

Warning: Using a password on the command line interface can be insecure.

```
uuid 7308a617-342a-4d90-bae2-20c94452c766
```

```
localhost oncloudtime oncloudtime oncloudtime 103 My Test Hive s3://oncloudtime-demo/data/RutrumIndustries/home/frank-morris-5df79a88-b73d-47c6-b482-eb681f3c9b4d/query/7308a617-342a-4d90-bae2-20c94452c766/ Query Code oncloudtime s3://oncloudtime-demo/athena-output/
```

```
Path to log file /Users/ssmusoke/PhpstormProjects/oncloudtime/var/log/frank-morris-5df79a88-b73d-47c6-b482-eb681f3c9b4d.1544806689.runquery.7308a617-342a-4d90-bae2-20c94452c766.log
```

```
S3 upload urls3://oncloudtime-demo/data/RutrumIndustries/home/frank-morris-5df79a88-b73d-47c6-b482-eb681f3c9b4d/query/7308a617-342a-4d90-bae2-20c94452c766/
total 304
```

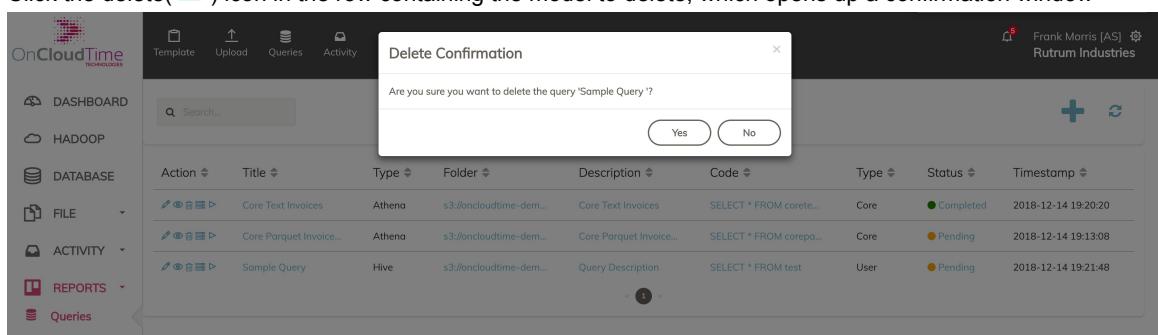
```
-rw-r--r-- 1 ssmusoke staff 727 Oct 24 08:57 addfileuploadrecord.sh
```

```
drwxr-xr-x 17 ssmusoke staff 544 Oct 24 08:57 admin
```

## 1.12.2.5 Delete Query

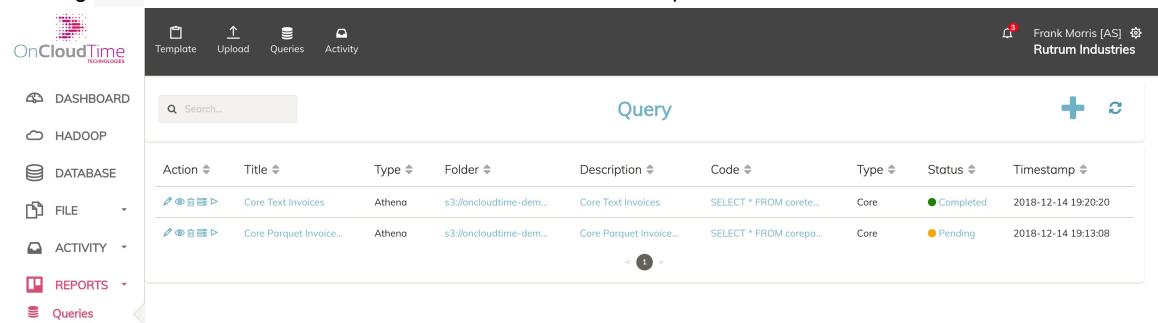
This removes a query from the application

- Click the delete() icon in the row containing the model to delete, which opens up a confirmation window



Action	Title	Type	Folder	Description	Code	Type	Status	Timestamp
	Core Text Invoices	Athena	s3://oncloudtime-dem...	Core Text Invoices	SELECT * FROM corete...	Core	Completed	2018-12-14 19:20:20
	Core Parquet Invoice...	Athena	s3://oncloudtime-dem...	Core Parquet Invoice...	SELECT * FROM corepa...	Core	Pending	2018-12-14 19:13:08
	Sample Query	Hive	s3://oncloudtime-dem...	Query Description	SELECT * FROM test	User	Pending	2018-12-14 19:21:48

- Clicking **Yes** in the confirmation window deletes the model and updates the list



Action	Title	Type	Folder	Description	Code	Type	Status	Timestamp
	Core Text Invoices	Athena	s3://oncloudtime-dem...	Core Text Invoices	SELECT * FROM corete...	Core	Completed	2018-12-14 19:20:20
	Core Parquet Invoice...	Athena	s3://oncloudtime-dem...	Core Parquet Invoice...	SELECT * FROM corepa...	Core	Pending	2018-12-14 19:13:08

