Estimate cost

When you change your subscriptions and databases, you also change the cost of your deployment. With a dry-run request, you can evaluate the impact and obtain a new cost estimate before you deploy these changes:

- Create subscription
- Create a database
- Update a database

Defining a dry-run request

API operations that support dry-run requests accept the dryRun boolean parameter in the JSON request body.

For example, the JSON body of a create subscription request body can include the dryRun=true parameter:

```json
{
    "name": "DryRun Cost evaluation subscription",
    "dryRun": true,
    "paymentMethodId": 8240,
    "cloudProviders": [
        {
            "cloudAccountId": 16424,
            "regions": [
                {
                    "region": "us-east-1",
                    "networking": {
                        "deploymentCIDR": "10.0.0.0/24"
                    }
                }
            ]
        }
    ],
    "databases": [
        {
            "name": "database-a",
            "memoryLimitInGb": 0.1
        }
    ]
}
```

Executing a dry-run request

Dry-run requests behave like regular requests except that no changes are made to existing resources. A dry-run request produces a cost evaluation report for the subscription.

<table>
<thead>
<tr>
<th>API Operation</th>
<th>dryRun=false (default)</th>
<th>dryRun=true</th>
</tr>
</thead>
<tbody>
<tr>
<td>API Operation</td>
<td>dryRun=false (default)</td>
<td>dryRun=true</td>
</tr>
<tr>
<td>--------------------</td>
<td>-----------------------------------------------------------------------------------------</td>
<td>------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Create subscription</td>
<td>Create a subscription</td>
<td>Returns a cost evaluation report of the planned subscription</td>
</tr>
<tr>
<td>Create database</td>
<td>Creates a new database in the subscription</td>
<td>Returns a cost evaluation report for the relevant subscription</td>
</tr>
<tr>
<td>Update database</td>
<td>Changes the specified database</td>
<td>Returns a cost evaluation report and evaluates whether the relevant subscription requires additional resources based on the database modification</td>
</tr>
</tbody>
</table>

**Example of a dry-run request & response**

This section demonstrates a complete example of a create subscription dry-run request and response.

**dry-run request body**

Here is an example of the pricing request body:

```json
{
    "name": "DryRun Cost evaluation subscription",
    "dryRun": true,
    "paymentMethodId": 8240,
    "cloudProviders": [
        {
            "cloudAccountId": 16424,
            "regions": [
                {
                    "region": "us-east-1",
                    "networking": {
                        "deploymentCIDR": "10.0.0.0/24"
                    }
                }
            ]
        }
    ],
    "databases": [
        {
            "name": "database-a",
            "throughputMeasurement": {
                "by": "operations-per-second",
                "value": 1000
            },
            "memoryLimitInGb": 0.1,
            "quantity": 3
        },
        {
            "name": "database-b",
            "throughputMeasurement": {
                "by": "operations-per-second",
                "value": 1000
            },
            "replication": false
        },
        {
            "name": "database-c",
            "memoryLimitInGb": 0.1
        }
    ]
}
```
The create subscription request contains the "dryRun": true parameter

2. The databases array contains the definitions of multiple database templates, named database-a, database-b, database-c, etc.

3. Note that the databases differ in structure (size, throughput measurement, replication, quantity, etc.)

dry-run response

Here is an example of the pricing response section for the above create subscription dry-run request:

```
{
  "response": {
    "resource": {
      "pricing": [
      {
        "databaseName": "database-a",
        "type": "Shards",
        "typeDetails": "micro",
        "quantity": 6,
        "quantityMeasurement": "shards",
        "pricePerUnit": 0.027,
        "priceCurrency": "USD",
        "pricePeriod": "hour"
      },
      ]
    }
  }
}
"databaseName": "database-b",
"type": "Shards",
"typeDetails": "micro",
"quantity": 1,
"quantityMeasurement": "shards",
"pricePerUnit": 0.027,
"priceCurrency": "USD",
"pricePeriod": "hour"
},

"databaseName": "database-c",
"type": "Shards",
"typeDetails": "high-throughput",
"quantity": 2,
"quantityMeasurement": "shards",
"pricePerUnit": 0.124,
"priceCurrency": "USD",
"pricePeriod": "hour"
},

"databaseName": "database-d",
"type": "Shards",
"typeDetails": "small",
"quantity": 1,
"quantityMeasurement": "shards",
"pricePerUnit": 0.156,
"priceCurrency": "USD",
"pricePeriod": "hour"
},

"databaseName": "database-e",
"type": "Shards",
"typeDetails": "large",
"quantity": 2,
"quantityMeasurement": "shards",
"pricePerUnit": 0.293,
"priceCurrency": "USD",
"pricePeriod": "hour"
},

"type": "EBS Volume",
"quantity": 345,
"quantityMeasurement": "GB"
},

"type": "r5.xlarge",
"quantity": 3,
"quantityMeasurement": "instances"
] }
1. The pricing array contains an element for each database, containing the database name and cost evaluation related to that database.

2. The database cost is measured in type and number of shards required for the specific database, as defined by the database template in the request. See Cloud Pricing.

3. The cost evaluation for each database is measured in quantity of the specific shard type required by the requested database. The cost per shard is defined by the fields `pricePerUnit` (where `unit` is a shard of the specific type), `priceCurrency`, and `pricePeriod`.

4. For example, to calculate the total hourly cost of a database, use the following formula: \( \text{pricePerUnit} \times \text{quantity} \).

5. The structure of the pricing response depends on the cloud account used by the request. There are two types of cloud accounts: a cloud account owned, named, and managed by the customer (AWS only) and one maintained by "Redis resources" cloud account (All cloud providers).

6. For a customer-provided cloud account - The cost evaluation response lists the AWS resources required (storage and compute instances) without pricing data (which depends on the specific details of the customer’s AWS account).

7. For a Redis internal cloud account (defined a `cloudAccountId = 1` in the create subscription request) - The cost evaluation response includes a `MinimumPrice` element. This indicates the minimal hourly cost of the entire subscription. This minimum price will be charged if the sum of all shards for all the subscription's databases is less than the specified minimum price.

**Viewing actual subscription cost**

The Get subscriptions and Get subscription by id JSON response contains an element named `subscriptionPricing` that details the latest calculated cost of a subscription, grouped by shard type for all the databases in the subscription.

*Updated: August 10, 2021*