

**Cloud Resource Naming Conventions**

**Version 1.0**

**Author: Cloud Naming Standard Working Group**

**Date: 3/15/17**

**Document Revision History**

|  |  |  |  |
| --- | --- | --- | --- |
| **Date** | **Author** | **Version** | **Change Description** |
| 3/15/17 | Cloud Naming Standard Working Group | 1.0 | Initial version |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

**Table of Contents**

[1 Cloud Resource Naming 4](#_Toc477435578)

[1.1 All cloud resources (except EC2 instances) 4](#_Toc477435579)

[1.2 EC2 Instance Server Names 5](#_Toc477435580)

[1.2.1 Method One: best practice and in compliance with control standards: 5](#_Toc477435581)

[1.2.2 Method Two: 5](#_Toc477435582)

# Cloud Resource Naming

Cloud resource naming standards are important for the following reasons:

* supports the full automation work that is ongoing
* ensures needed information is available for resources that do not support tags
* consistent way of identifying resources which assist Ops and Dev team in trouble shooting and communication
* ensures unique names
* defines standard method to set the facts in four set sections that describe the resources

## All cloud resources (except EC2 instances)

The following will be used for all cloud resources (except EC2 instances, see section 1.2):

Pattern:

The naming pattern is broken into four set fact sections separated by a dash (-)

**<sysid>-<environment >-<bundlename>-<generic \_function>**

* Additional step for **S3** name
  + S3 buckets must be unique across the entire world, to accomplish this:
    - leave the S3 bucket name field blank in the cloudFormation template
    - then AWS will take the name of the cloudFormation template (which is named using the standard listed above) and adds an AWS hash at the end
* Examples:
  + tsm-dev-pipe1-ingestion
  + tsm-qa-pipe2-extractdata
  + S3: tsm-dev-pipe1-ingestion-appbucket-t2y8p5oilw73

Definitions:

* **sysid:** is the application sys id (3 characters) for the application that will be using this resource.
  + Example: TSM
* **environment:**  describes the environment the resource will be running in.
  + Example: Dev; QA Func; PSE; Prod; Lab; QA Perf
* **bundlename:** descriptor for a collection of AWS resources built via automation and allows multiple stacks to be created
  + Examples: pipe1, pipe2
* **generic\_function:** descriptor to give general idea of resource activity
  + Examples: ingestion, extraction

Formatting:

* words are all lowercaseto:
  + differentiates the numbers and characters:
    - examples: I and 1 or o and 0
  + every name can be treated as a url (which some are), prevents different rules for different services
* ‘-‘ for separating the name into the four fact sections
* Permitted characters:
  + Name will only contain:
    - Numbers
    - Letters
  + Separator of the four fact sections
    - – (dash)

## EC2 Instance Server Names

In an effort to move to generic auto –generated names, method one is primary, preferred and compliant method.

But understanding the needs of the business during this transformation period, method two can be used but only with an approved exception from TRM.

### Method One: best practice and in compliance with control standards:

Generic auto-generated name:

[IP-10-10-0-200.aws.com](mailto:IP-10-10-0-200@aws.com), or something similar

Utilize Alias records for friendly names where needed.

### Method Two:

Note: an exception will need to be raised.

EC2 instance server names in our VPC, we will use the following naming conventions. This pattern corresponds in a fashion to the names currently at use in DTCC.

* The first two characters will be “CV” (for cloud virtual).
* Characters 3 – 5 is the function of the server (similar to current DTCC practices), such as “app” for application server, “syb” for Sybase, “ora” for Oracle, etc.
* Character 6 is the environment (matches the DTCC environment), “d” is development, “q” for QA, “u” for pse, “p” for production.
* Characters 7-10 is a unique number for the server.

For example cloud VPC based Sybase development server which is server number one:

* ***Cloud Virtual Sybase Development 0001***
  + ***cvsybd0001***
* ***Cloud Virtual Application Server Production 1234***
  + ***cvappp1234***