Global Resource Naming

Certain resources in Cloud services require their names to be universally unique. In order to provide predictability the following naming standard is used.

Institution[1]-Organizational Name[2]-Environment[3]-Purpose[4]

As an example, the globally unique name of an AWS S3 Bucket that is used to store Terraform state information for Admins of the `Dev` AWS Account would be named :

```
VUMC-CloudServices-DevAdmin-TerraformState
```

To aid readability, CamelCasing should be used in the names. If the environment does not support CamelCasing, underscores should be used. The following is how the above example would appear if CamelCasing could not be used :

```
vumc-cloud_services-dev_admin-terraform_state
```

If neither underscores or CamelCasing is supported, as in the case of names that must be DNS compliant, then the terms should be run together. This happens to be a constraint of AWS S3 Bucket naming, so the S3 Bucket name in these examples would have to be :

```
vumc-cloudservices-devadmin-terraformstate
```

Localized Shortening

In the case where the Institution, Organizational Name, or Environment is already a part of the resource path, those elements can optionally be removed. This provision is for preserving readability through removal of name component duplication. For example, all GitHub repositories in the `VUMC Cloud` GitHub Organization implicitly have **VUMC** as an institution component, so it does not have to be a component of a repository name. As such, GitHub repository names in the `VUMC Cloud` GitHub Organization can begin with the `Organizational Name` component.

When it is ambiguous whether a name component is implied or not, implication should not be assumed. Continuing with the `VUMC Cloud` GitHub Organization example, it is unclear whether it is only for Cloud Services repositories or whether other departments and groups will use it. Due to that ambiguity, the `Organizational Name` component should always be used. In the event that it becomes dedicated to one organization, such as Cloud Services, then the GitHub Organization name should change to `VUMC Cloud Services` and then the `Organizational Name` component can be dropped from repository names.

Exceptions

The following exceptions cover situations that don't fall strictly within the naming definition above, and require some adjustments or rationalizations to reasonably fit within the structure.

Azure Storage Accounts

Azure Storage Accounts have a number of unusual limitations. The name length is capped at `24` characters, and those characters *must* be lowercase and alphanumeric. This means there can be no special character usage for delineations, such as – or _. These limitations wouldn't be so onerous except Azure Storage Account names must also be universally unique.

To help insure universal uniqueness, the first four characters of a Storage Account is `vumc`.

The next set of characters are determined by the Storage Service that the Storage Account is being used for, following the Azure Storage documentation.

The remaining characters are free form, and left up to the creator of the Storage Account to try and provide something both unique and meaningul in roughly a dozen lowercase characters.

For example, an Azure Storage Account that is using the Files Storage Service for a Kubernetes cluster in a "sandbox" subscription owned by CloudServices might be named:

```
vumcfilescldsvsdev1aks01
```

Due the fact that different areas will consider different elements to be more or less significant it is intentional that the characters after the `vumc` and Storage Service type is considered freeform.

---

1. Helps insure global uniqueness. ↩
2. Billing boundary that can refer to department or project. ↩
3. Logical boundary based on purpose, such as development, staging, administrative, etc. ↩
4. A short description of the resource's intended purpose. ↩