# 3.1 AWS Cloud Reference Architecture Layout

## ****Summary****

* AWS Environment Overview
* Cloud Connectivity
* Tagging Strategy
* Storage
* Backup & Restore
* Disaster Recovery
* Access & IAM
* Security Groups
* Encryption and Key Management
* Patching & Vulnerability management of Cloud servers
* Monitoring & Logging
* Application Architecture
* Container services

**AWS environment Overview**

**VPCs**

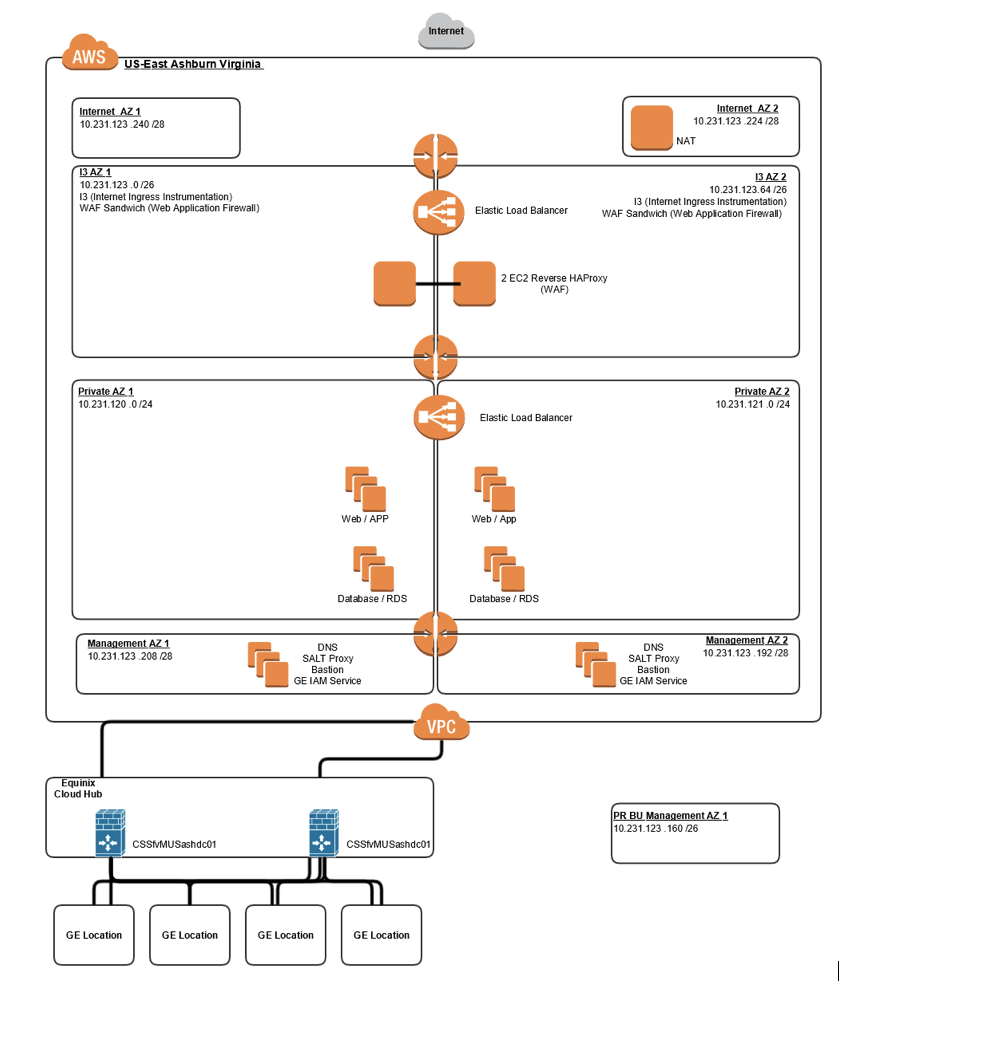
Amazon Virtual Private Cloud (Amazon VPC) enables you to launch AWS resources into a virtual network that you've defined. This virtual network closely resembles a traditional network that you'd operate in your own data center, with the benefits of using the scalable infrastructure of AWS.

They are 2 types of VPC setup in AWS (from CoreTech), Classic and Guard rails. Current Capital and Global Ops VPC are listed below.

| **VPC** | **Line of business** | **CIDR** | **LOCATION** |
| --- | --- | --- | --- |
| TREASURY Medium RIsk (aka Sandbox) | Capital - Classic | 10.230.225.0/24 | US East 1 |
| Capital Stage | Capital - Classic | 10.230.250.0/23 | US East 1 |
| Capital Prod | Capital - Classic | 10.231.120.0/22 | US East 1 |
| EMEA Prod | Capital - Classic | 10.225.172.0/22 | EU West 1 |
| gr-ggo-financeit-prod | Global Ops – Guard Rails | 10.232.48.0/21 | US East 1 |
| Arcoe | Guard Rails | 10.222.40.0/21 | EU West 1 |

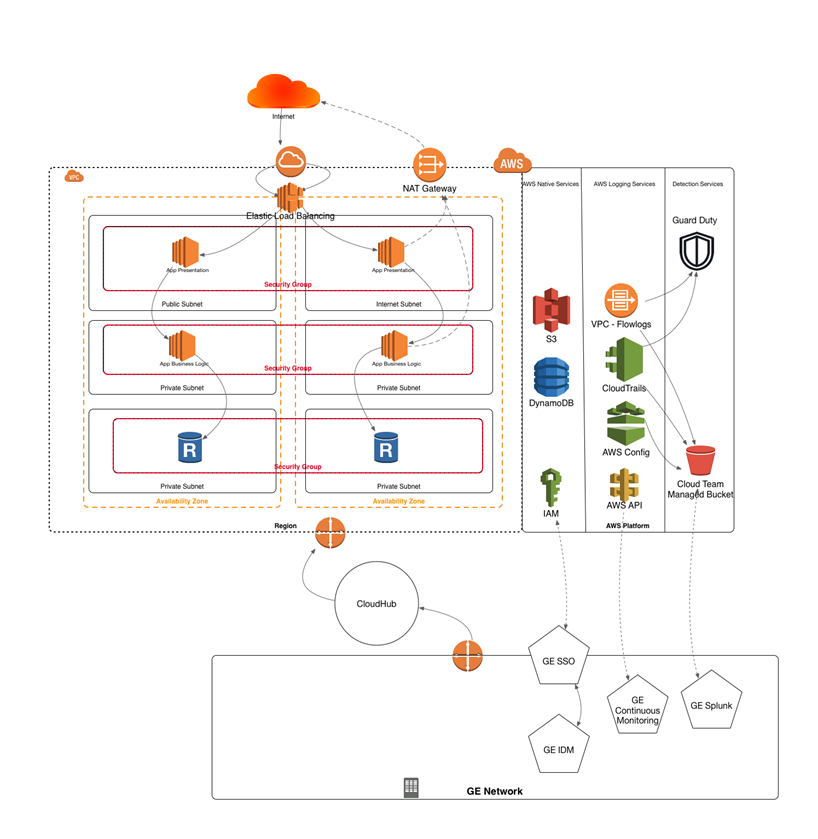
**GE Capital Prod VPC design**

 The below diagram shows typical Capital VPC architecture with different subnets and availability zones which can offer HA for application. By architecting the application across multiple availability zones, we can build a highly resilient application.



**Connectivity**

* Equinox Hub is used to connect with several AWS Cloud regions.
* On premise firewalls control ingress access into the cloud (Direct Connect).
* VPC peering is not done. Any communication with different VPCs has to come through Cloud hub
* Egress traffic are managed by NAT gateway and NAT instances. The NAT will only support standard ports like port 80 and 443.
* For non standard ports egress communication, need to work with Network team to evaluate whether Plug Proxy is the right tool.
* Ingress traffic to Cloud are managed by I3 proxies and WAF devices
* Security Groups allow/deny the network traffic to EC2 and other AWS resources.
* For Onprem to Cloud communication, need to open up firewall rules on top of setting up right Security Groups.



## [Capital AWS Tagging Standards](https://devcloud.swcoe.ge.com/devspace/display/NRINN/Capital+AWS+Tagging+Standards)

This page describes Tagging standards for Capital Cloud Databases and VMs, which are on top of GE standards set by Corporate.

### GE Corporate Requirements

| **Tag Key** | **Tag Value Form** | **Example Tag Values** | **Mandatory** | **Tag Purpose** | **Where Used** |
| --- | --- | --- | --- | --- | --- |
| Name | common-name | my-app-1 | Yes | This is used for the common name for anything tagged.  The default “view” tag used for easy filtering | Used primarily for sorting/finding objects within the AWS console (part of the default view for most services).  Can also be used as the simple name on any bill. |
| role | type-of-role | web-server | Yes | Tag captures the type of role for the tagged aws object | Will be used (in combination with env) to build complex roles to all restricted access to objects (i.e. this IAM user can start/stop all dev web-servers, but not production. |
| app | app-ci-value | gr-my-app-1-prd | Yes | Tag captures a consistent link back to the CI/App name | Will be used for linking to billing.  This is how a BU would divide up their bill. This may also be how we build polices to protect application-to-application communication. |
| env | environment | production | Yes | Tag captures the environmental state for the tagged object, (i.e. dev, stage, prod, stage, etc) | Will be used (in combination with role) to build complex roles to all restricted access to objects (i.e. this IAM user can start/stop all dev web-servers, but not production. |
| UAI | UAI | Unique Identifier | Yes | Tag to be used for admin-functionality.  This will be used for the App Repository purpose. | This will be reserved for admin tags (to be applied by the centralized team).  The UAI will be generated by the App Repository. The UAI will be placed by the BU, manually or automatically, on the approved applications residing at Cloud provider. The UAI will be queried and matched against the App repository, if a delta exists then a corporate action will be executed on against the app - details TBD. To register an app visit <https://applications.ge.com/registration> |
| OPEN | OPEN | Client Defined | No | Tag to be used for the purpose of client actions. | This will be reserved for client usage only and can be variable to clients demand |

## Best Practices:

1. Define a Naming Standard
2. Lower Case (except for Name)
3. No Spaces
4. Dashes vs. Underscores/Spaces
5. Be consistent!

GE Tagging "Rules of the Road" presentation: <http://libraries.ge.com/foldersIndex.do?entity_id=44133338101&sid=101&SF=1>

## AWS Tag Restrictions

The following basic restrictions apply to tags, [as of 04/20/2017](http://docs.aws.amazon.com/AWSEC2/latest/UserGuide/Using_Tags.html#tag-restrictions):

* Maximum number of tags per resource—50
* Maximum key length—127 Unicode characters in UTF-8
* Maximum value length—255 Unicode characters in UTF-8
* Tag keys and values are case sensitive.
* Do not use the aws: prefix in your tag names or values because it is reserved for AWS use. You can't edit or delete tag names or values with this prefix. Tags with this prefix do not count against your tags per resource limit.
* If your tagging schema will be used across multiple services and resources, remember that other services may have restrictions on allowed characters. Generally allowed characters are: letters, spaces, and numbers representable in UTF-8, plus the following special characters: + - = . \_ : / @.
* You [can't tag all resources](https://aws.amazon.com/blogs/security/now-organize-your-aws-resources-by-using-up-to-50-tags-per-resource/), and some you can only tag using API actions or the command line.
* The user: prefix in tag names is reserved by Amazon for [cost allocation tags](http://docs.aws.amazon.com/awsaccountbilling/latest/aboutv2/allocation-tag-restrictions.html) and should not be used.
* Creating a hierarchy of tag keys using ":" is a good way to isolate namespaces, or to allow partial matching of keys

### Additional AWS Restrictions and Suggestions

**Example:**  The keys system:users and webapp:users allow system administrators and web administrators to share a meaningful label name but clearly be distinct from each other

**Example:**  The scheduler:ebs-snapshot key would define a default schedule for automated snapshot creation, unless a more specific label scheduler:ebs-snapshot:morning, scheduler:ebs-snapshot:noon or scheduler:ebs-snapshot:night exists to override it

## Tag Mappings to Capital Cloud Scalr Global Variables:

| **Tag Name** | **Scalr Variable** |
| --- | --- |
| Name | defaults to "{Farm Name} -> {Farm Role} #{instance count}" |
| role | {Farm Role} |
| app | (none defined) |
| env | {CONFIG\_ENV} **NOTE: MAPS TO "Env"** |
| UAI | {UAI} |
| scalr-meta | Used internally by Scalr |

# GE Capital Tags

| **Tag Key** | **Tag Value Form** | **Example Tag Values** | **Mandatory** | **Tag Purpose** | **Where Used** |
| --- | --- | --- | --- | --- | --- |
| scheduler:ebs-snapshot | [See explanation](https://devcloud.swcoe.ge.com/devspace/display/NRINN/Automating+EBS+Snapshots#AutomatingEBSSnapshots-CustomParameters) | default  1030;15;us/pacific;mon,tue,fri | yes | Applicable to EC2 instances. Controls the scheduling and behavior of automated snapshots | This tag should be applied to every EC2 instance, even ones with no backups (in these cases the value will be set to "none") |
| Schedule | [See explanation](https://devcloud.swcoe.ge.com/devspace/pages/viewpage.action?pageId=1141022685#MVP-17:Set/ManageSchedulingwindowforpre-prodservers-ScheduledWindows) | U11D02-7  U06D18-5 | No | Applicable to EC2 instances. Automates the starting/stopping of EC2 instances when not in use | Taken from [GE Power's initiative](https://devcloud.swcoe.ge.com/devspace/pages/viewpage.action?pageId=1141022685). Would need to be implemented/adapted for Capital's use. |

Links:

<https://devcloud.swcoe.ge.com/devspace/display/NRINN/Capital+AWS+Tagging+Standards>

## [Storage for AWS](https://devcloud.swcoe.ge.com/devspace/display/EGKMQ/Storage+for+AWS)

There are multiple types of storage available to the Solution Architect based on the application needs. Application must encrypt their data at rest using AWS KMS keys regardless of what storage type their app uses.

Some of most commonly used storage types are listed below

1. Object Storage (S3 - Simple Storage Service)
   1. S3object store are primarily used to store snapshots and other application artifacts. They cannot be directly attached to any EC2 instance. It is highly scalable, reliable, fast, inexpensive data storage infrastructure from AWS.
   2. <https://aws.amazon.com/s3/>
2. Amazon Elastic Block Store (EBS)
   1. EBS is closer to SAN. It is block store can be attached to the server for persistent storage but once the instance are terminated. The data in the volume would be lost based on the configure option chosen.
   2. <https://aws.amazon.com/ebs/>
3. Amazon Elastic File System (EFS)
   1. EFS is very similar to NAS and performs file locking.  This is not available for windows and only available for Linux. The solution should NOT be use as a primary solution and S3 or EBS is preferred.  EFS has higher cost ($0.30 per Gb vs S3 being $0.03 per GB)
   2. <https://aws.amazon.com/efs/>
4. AWS Storage Gateway
   1. AWS Storage Gateway connects an on-premises software appliance with cloud-based Storage like S3. It can be used as an alternative to NAS storage with few tweaks.
   2. <https://aws.amazon.com/storagegateway/?nc2=h_m1>

### ****Backup & Restore****

* Native EBS based snapshots will be used for backups
* Snapshots backup can be achieved by several modes -  AWS console, AWS CLI, Lifecycle manager and Scalr.
* LifeCycle Manager is under PoC which automate the snapshot and its life cycle management.
* Restores can be done in the need by basis. Snapshot can be restored on the volume which will be attached to the EC2 instances.
* RDS snapshot are used for full DB backup

**Disaster Recovery**

Businesses must have a documented application recovery plan. The application recovery plan must include the following elements:

* Application recovery scope
* Required recovery time objective (RTO)
* Required recovery point objective (RPO)
* Process to invoke the Application Recovery plan, including parties authorized to invoke the process for application recovery
* Steps performed to validate that the application recovery has been completed successfully
* Frequency of testing and validation
* Minimum requirement of annual test
* Annual review and sign off by Business and IT of the Application Recovery Plan
* Annual review and sign off the test results by Business and IT

**DR Pattern**

AWS cloud infrastructure allows multiple DR patterns

* Backup and Restore
* Pilot Light
* Warm Standby
* Multisite Application
* Combination

Application architecture determines the possible DR patterns for that application

* DR Tier (RPO/RTO) may require application architecture change to adopt the required DR pattern.
* i.e. Application logic to have the application consistency across primary and DR instance.

Application DR location - different zone (within Production env.) or geographically redundant region based on application requirements and cost considerations.

## ****AWS Public Cloud Identity Model (IAM & SSO Integration)****

This document strives to explain how to get work done within the GE Public Cloud IAM permissions model that’s included with new and existing Guardrails accounts.

### Contents

* [Prerequisites and Assumptions](https://docs.cloudpod.apps.ge.com/identity/identity-customer#prerequisites-and-assumptions)
* [Introduction](https://docs.cloudpod.apps.ge.com/identity/identity-customer#introduction)
* [Quick Start](https://docs.cloudpod.apps.ge.com/identity/identity-customer#quick-start)
* [Distribution Lists and Roles](https://docs.cloudpod.apps.ge.com/identity/identity-customer#distribution-lists-and-roles)
* [AWSIAM and the Deny Policies](https://docs.cloudpod.apps.ge.com/identity/identity-customer#awsiam-and-the-deny-policies)
* [Situations Outside of bu-iam-admin Control](https://docs.cloudpod.apps.ge.com/identity/identity-customer#situations-outside-of-bu-iam-admin-control)

### Introduction

Once a GE Guardrails account has been provisioned and released it contains several roles and policies already configured to allow the members of certain groups to log into the console and perform basic tasks.

The pre-deployed roles are designed to get up and running at a basic compute level but are by no means intended to the only roles that are ever used in the account.

Since each business is different and has different needs and desires the roles are left mostly generic and it’s assumed that the person(s) designated as the IAM administrator for the account will be responsible for building additional policies and modifying roles as necessary.

This guide goes over the various roles and explains roughly what they are capable of doing “out of the box”. Additionally, the guide will show some examples of how roles might be modified to perform other common tasks.

Where possible the guide strives to explain things that are “just GE things” and which are AWS defaults.

### Quick Start

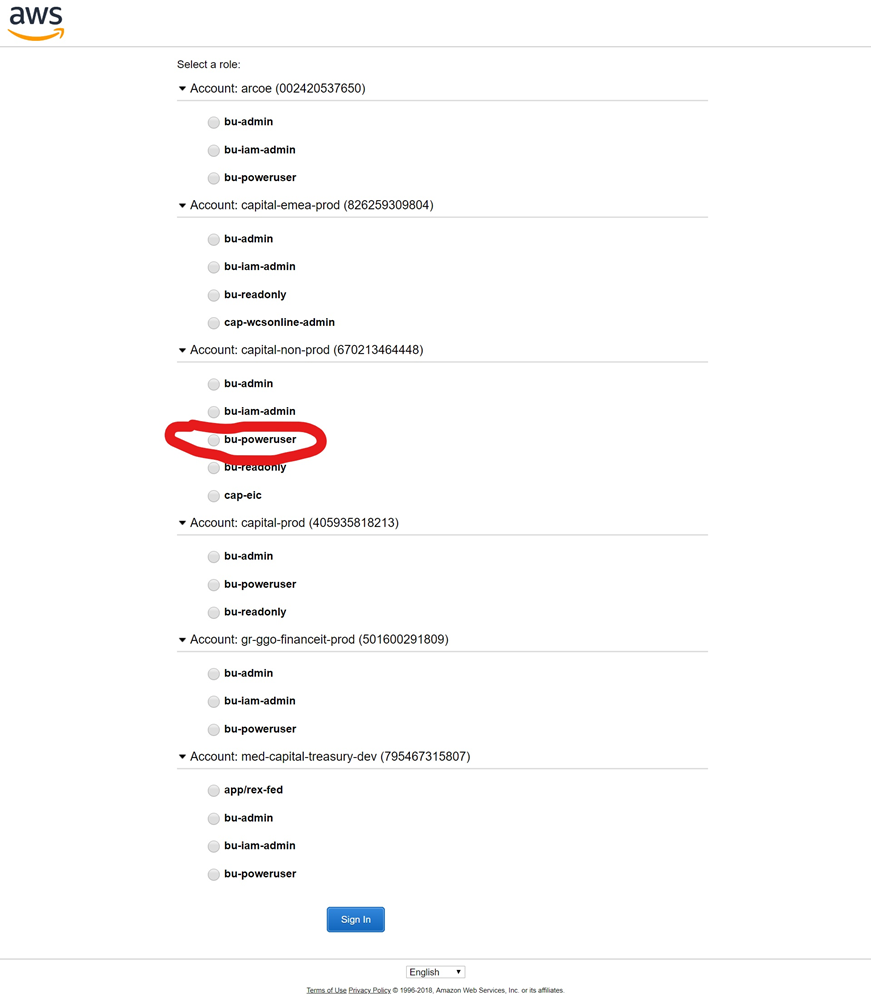
If the desire is to just jump in and start playing around in the console then you can start with this portion of the guide.

First, you’ll need to make sure you’re a member of at least one of the DL’s associated with your account. There were several DL’s created as part of your account build. For detailed explanations of the DL’s and roles see the Distribution Lists and Roles section.

For now we’ll assume that you were the account creator and you’ve been given access to the bu-poweruser DL for your account.

First navigate to [http://sc.ge.com/\*AWSEntLogin](http://sc.ge.com/*AWSEntLogin). After logging in with your SSO you’ll either be logged directly into the console if you only have one DL membership or you’ll be presented with a list or roles if you have multiple DL memberships.

In the below screenshot the user has access to the bu-poweruser role in the capital-non-prod account.



Select the bu-poweruser role and click Sign In

That’s it, you’re logged into the console. You can always tell what role you’re logged in as and the current account/context by clicking the profile dropdown.

Take a moment to play around in the console or continue reading the guide for more information.

### Distribution Lists and Roles

Every AWS Guardrails account provisioned at GE is accompanied by several IDM Distribution Lists that can be used as a jumping off point for managing access levels in the account.

NOTE: As of 4/20/2018 most new Guardrails accounts only ship with the bu-iam-admin role which should have the ability to create all of the other roles you need.

If you head over to the [IDM Distribution List](https://oneidm.ge.com/faces/modules/my_groups/distribution_lists_join.xhtml) search page and type in the name of your account to find all of the Distribution Lists that were created with your account.

The below screenshot shows several of the DL’s associated with the capital-non-prod account.

DL names are structured as follows

 @GE AWS\_<account\_alias>\_<role\_name>\_<account\_number>

The roles are collections of various permissions and can be roughly broken down as follows:

| **Role Name** | **Purpose** | **Job Role** | **Risk Level** |
| --- | --- | --- | --- |
| bu-admin | very limited set of high level VPC permissions specifically for managing route tables, NACL’s, etc. | Sysadmin | Medium |
| bu-brt | for the business response teams such as an IT risk organization within your specific business unit (e.g., Healthcare IT Risk) | BU Security Professional | Medium |
| bu-i3-admin | very limited set of permissions that allow users to create and manage i3’s for internet ingress | Sysadmin or Web Admin Team | High |
| bu-iam-admin | scoped set of permissions that allow users to create/modify IAM permissions within the account | Sysadmin | High |
| bu-poweruser | broad set of permissions for general compute usage and infrastructure management | Application Engineers and Support | Medium |
| bu-readonly | limited set of read-only permissions | App Teams, Support, Auditors | Low |
| bu-user | limited set of compute permissions similar to poweruser but less powerful | App Teams, Support | Medium |
| cirt | scoped set of permissions that let the CIRT respond to global incidents and alerts, not for use by the BU or app owner | CIRT Security Professional | Medium |
| cloud-admin | LEGACY: allows CoreTech cloud operations team access into the account to reboot instance, etc. No longer necessary | CloudPod Support Engineer | Medium |
| cloud-support | LEGACY: allows Coretech cloud operations team read only access into the account | CloudPod Support Engineer | Low |

If you’re the account owner you may be listed as the Primary Manager on all of the bu-\* DL’s. This gives you the responsibility of granting and revoking access to the above roles per the policies and procedures of your own organization. There may be other managers listed on the DL that have delegated permissions to add and remove users from the DL’s. Ownership of the DL’s can be transferred as people leave roles or the organization.

Common use cases:

* I’m an application developer and I need to start creating ec2 instances and load balancer and other infrastructure to host my application
  + Use the bu-poweruser role.
  + I’m having trouble finding a subnet I can use to launch my infrastructure
    - Use the bu-admin role to create subnets and build route tables.
    - I’m using the bu-poweruser role but getting a permissions problem when trying to create an ECS cluster
      * Use the bu-iam-admin role to add the necessary policies to the bu-poweruser role.
      * I have an auditor that keeps asking me for an inventory of the account
        + Add the auditor to the bu-readonly DL so they can log in and collect data on their own.

### AWSIAM and the Deny Policies

Within every account there is a service that GE runs called AWSIAM. The purpose of this service is to scan IAM in the account a few times per second for new roles, users, and groups to make sure they have deny policies attached to them. You’ll notice within seconds of creating a new role that policies will show up attached like this:

The purpose of the deny policies are to stop users from doing things within the account that are deemed to be against GE policy. These include but are not limited to:

* Creating public IP addresses.
* Launching images from unapproved source accounts.
* Joining other Organizations.
* Using unapproved services such as:
  + Route53
  + Cloud Directory
  + Workspaces
  + Modifying Direct Connect back to GE Core
  + Deleting protected roles used for corporate audit and security.
  + Deleting Guardrails infrastructure critical to the operation of the account and critical to maintaining security within the account.

The full list of unapproved actions can be understood by studying the awsiam-sauce and awsiam-sauce-the-sequel policies that are automatically attached to all new roles/users/groups that are created in the account.

You should be able to grant the permissions you need for most purposes using the bu-iam-admin role.

With a few exceptions, the role can do everything that’s allowed within GE policies and restrictions. For example, if you created a role and gave it \*.\* the awsiam-sauce\* policies would come and attach themselves and you would still be compliant. It comes down to what you and your business require from an audit perspective.

As bu-iam-admin you have been given the responsibility of protecting the IAM scope for the applications in your account. How you solve the IAM challenges is totally up to you.

When struggling with getting a particular solution set up a possible solution is creating a role that can accomplish and what you need and then temporarily allowing a role such as bu-admin to switch-role to that role and perform the activities needed. You could leave that in place or revoke the assume-role permissions afterwards.

### Situations Outside of bu-iam-admin Control

There are some situations where the CloudPod team absolutely has to step in and assist with such as:

* Creating a role that can create other roles.
  + This will always be blocked by awsiam-sauce.
  + Creating a key for an automated user that can assume-role into multiple accounts. (e.g., for automated audit/alerting purposes)
    - This has to be done in another account.
    - Deleting legacy IAM users that have policies attached directly
      * awsiam-sauce started blocking direct user-policy attach/detach about a year ago and a delete requires a detach.

For these issues you can submit an [exception request form](http://forms.ge.com/create/407018) and we’ll evaluate the request for AWSIAM exceptions. Alternatively you can just cut a ticket to [http://sc.ge.com/\*clopsRITM](http://sc.ge.com/*clopsRITM) and CloudPod can facilitate the request.

**Links**:

<https://docs.cloudpod.apps.ge.com/identity/identity-customer>

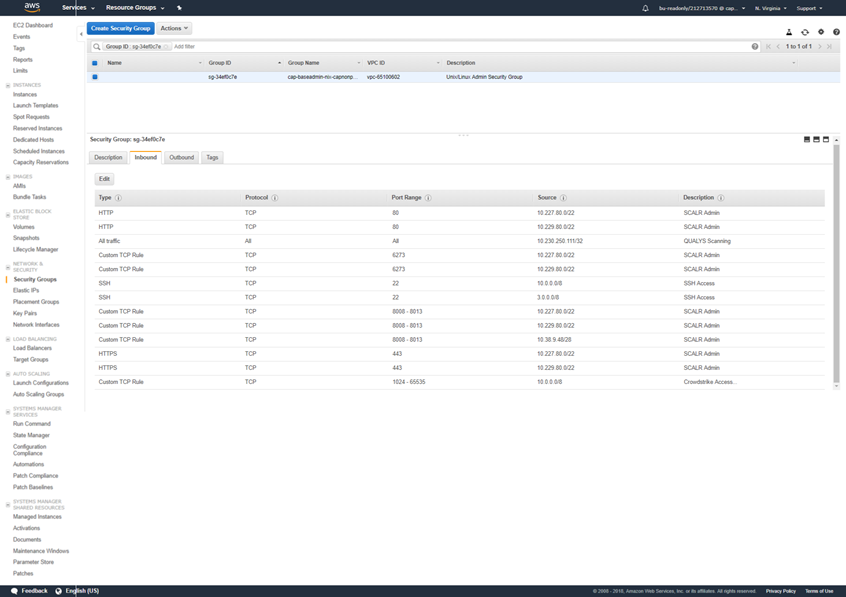
**Security Groups (SG)**

Security group acts as a virtual firewall for your instance to control inbound and outbound traffic. Security groups act at the instance level, not the subnet level. Therefore, each instance in a subnet in your VPC could be assigned to a different set of security groups.

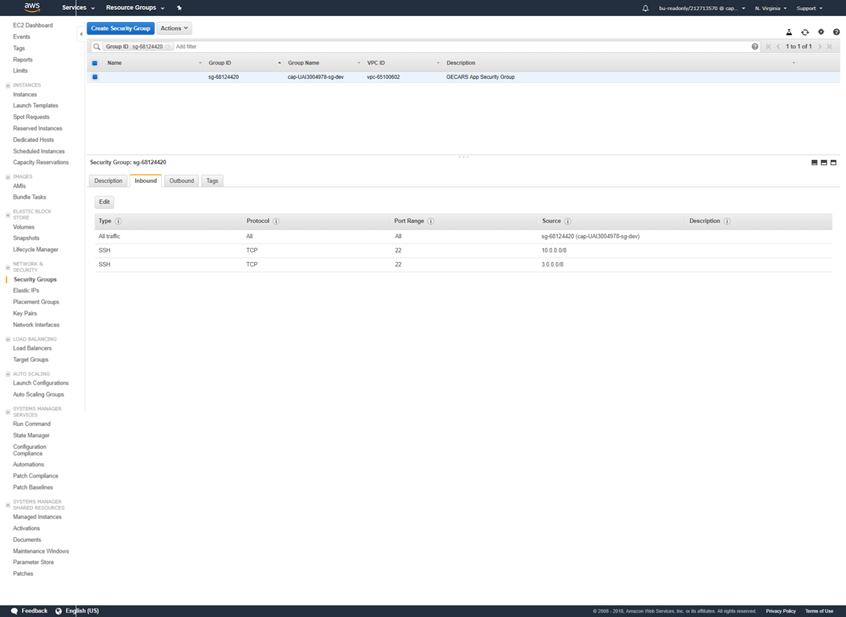
For each security group, you add rules that control the inbound traffic to instances, and a separate set of rules that control the outbound traffic. This section describes the basic things you need to know about security groups for your VPC and their rules.

We have been creating 2 kinds of Security groups. Base Security group and Application Security group. Base group will be assign during the EC2 instance build. App Security group can be added later as the application installation and configuration progress. Typically, App Security Group is required for Application to work properly.

**Example of a base Security group**



**Example of Application Security group**



## ****Encryption and Key Management****

AWS native KMS will be used to encrypt and decrypt data at rest within AWS. Roles will allow access to decryption as needed.

At rest storage encryption – To be added

# Links:

<https://devcloud.swcoe.ge.com/devspace/display/XCCER/User+Information>

AMI –  To be added

Patching - To be added

Monitoring & Logging  - To be added

Application patterns -  To be added