AWS

- Adding Administrator Access Existing AWS Accounts for Onboarding — When we onboard existing AWS accounts into the main Cornell AWS Organization, we need administrator-level access to the account in order to implement our Standard AWS Account Configurations. These instructions shown how to accomplish that.
- After Onboarding to AWS
- Approaches for CLI, SSH, and RDP Access to AWS and AWS Resources — The pages contains tools and approaches to accessing AWS resources deployed in Cornell AWS accounts and to executing AWS APIs. This information was compiled as CIT teams realized the need for "air gapped" workstations or platforms to work with controlled data and systems. Some of these options may be appropriate replacements to the on-premise, controlled access systems currently being used for tools workstations or utility servers.
- AWS: Convert a server running from Instance Store to EBS (possibility)
- AWS Certificate Manager
- AWS Chatbot
- AWS Client VPN
- AWS Cross-Account Secret Access — AWS Secrets Manager http://is a great way to safely store secrets needed by applications. Sometimes you need to access those secrets from an AWS account other than the account where the secret is stored. Here are some notes about that. is a great way to safely store secrets needed by applications. Sometimes you need to access those secrets from an AWS account other than the account where the secret is stored. Here are some notes about that.
- AWS EC2 Reserved Instances — Reserved Instances at AWS provide a significant discount (up to 75%) compared to On-Demand pricing and can provide a capacity reservation when used in a specific Availability Zone.
- AWS Elastic File Systems (EFS) Troubleshooting
- AWS FAQs — Answers to questions about AWS that we often see at Cornell.
- AWS Networking
    - AWS Network Reach Tool — This Python tool was created to help analyze AWS network connectivity.
    - AWS Public IPv4 Use
    - Baseline AWS Network ACL — This network ACL http://docs.aws.amazon.com/AmazonVPC/latest/UserGuide/VPC_ACLs.html is the recommended baseline for VPC subnets in Cornell AWS accounts. It should be configured and used on all AWS VPC subnets. You are welcome to make your NACL more stringent, but we recommend careful consideration before making it less stringent.
    - Cornell AWS Direct Connect
        - 2023 Cornell AWS Direct Connect Architecture Migration — This document provides details about the Direct Connect architecture migration Cornell will be executing in early 2023.
            - Terraform Configuration Guidance for 2023 Direct Connect Architecture Migration
        - Cornell AWS Direct Connect Architecture
        - Cornell AWS Direct Connect Costs — The management and routing simplification offered by the v2 (2023) architecture comes with a shift in costs seen by Cornell AWS accounts using Direct Connect, but the overall impact to Cornell AWS account costs are negligible.
        - Cornell AWS Direct Connect FAQs
        - Cornell AWS Direct Connect Routing Diagrams — This page shows and discusses the different routing options over the Cornell Direct Connect to AWS VPCs.
        - Direct Connect Resources in Cornell AWS Accounts — This document provides details about the resources in Cornell AWS accounts that support the 2023 (v2) Direct Connect architecture.
        - Peering AWS VPCs that Use Direct Connect — Since the 2023 Direct Connect architecture fully interconnects all VPCs using Direct Connect (i.e., attached to the Transit Gateway), individual peering between VPCs is no longer technically necessary.
    - Route 53 Subdomain Delegation — Delegating specific sub-domains of cucloud.net to Route 53 allows your group or department the ability to create dynamic environments with the tools provided by Amazon Web Services. While anyone can create a Hosted Zone for a sub-domain in Route 53, DNS delegation requires the owner/administrator of the parent domain ("cucloud.net") to create nameserver (NS) and start-of-authority (SOA) records that direct incoming requests for your specific sub-domain to the nameservers AWS assigned to the Host
    - Shared AWS VPC for Cornell AWS Accounts
        - Shared AWS VPC FAQs — FAQs about the Multitenant Subnets and the Exclusive Use Subnets option within the Shared VPC offering.
        - Using the AWS Shared VPC Offerings — This document provides practical information about using either the Multitenant Subnets or Exclusive Use Subnets options of the Shared VPC offering once its has been provisioned to your Cornell AWS account.
- AWS News
    - AWS Official Blog
    - AWS Recent Announcements RSS
- AWS S3-Glacier migration and backup strategies and 3rd party tools
- AWS Stand Alone account : tools/thoughts on closing/decommissioning
- AWS Standard Tagging
- AWS Storage Pricing Models
- AWS Storage Services Cost Comparison Tool — This handy new (as of May 2020) tool provides a quick comparison of storage costs across a huge variety of AWS services, including S3, Glacier, EBS, EFS, DynamoDB, and RDS.
    - AWS Region Comparison Tool
- Benefits of AWS Accounts under the Cornell-AWS Umbrella — We encourage all Cornell teams, staff, and faculty to contact the Cloud Team (cloud-support@cornell.edu) if they have an AWS account that they use for Cornell business or research purposes and that is not part of the contract that Cornell has with AWS. Here are some of the benefits of getting a new AWS account, or bringing an existing AWS account to the master Cornell-AWS contract.
- Configure Oracle RDS to use AWS SES for SMTP
- Configure Oracle RDS to Use AWS SES SMTP — Configuring Oracle RDS instances to use AWS SES SMTP is not a trivial process. This page pulls together some documentation that might be helpful if you are trying to do that.
- Cornell User VPN and AWS Interactions
- Enabling MFA Delete for AWS S3 Buckets — When working with S3 Versioning in Amazon S3 buckets, you can optionally add another layer of security by configuring a bucket to enable MFA (multi-factor authentication) delete. When you do this, the bucket owner must include two forms of authentication in any request to delete a version or change the versioning state of the bucket.
- Examples of Email Sent to AWS Root Account Addresses and AWS Security Contacts
- How To Recover Management Access to AWS KMS Keys — Each key managed by the AWS Key Management Service (KMS) must have a resource policy that describes what AWS security principals can use and manage the key. If you create a policy that does not include

management privileges for any principal or if principals named in the policy are themselves deleted, you may find yourself unable to manage a KMS key. Fortunately AWS provides a way to regain control of the key in such a situation.

- How to Request a Cornell AWS Account
- IAM Policy to Restrict Scope of Privileges
- Interactions of AWS Root Accounts and Amazon.com Accounts
- Login to AWS Console with Cornell Two-Step Login
- Microsoft Licensing within AWS
- Monitoring in AWS
- Regulated Data in AWS
- S3 Controls - Block Public Access
- Standard AWS Account Configurations
- Standard AWS Technical Solutions
    - Access Keys for AWS CLI Using Cornell Two-Step Login - Shibboleth — This document shows how to setup and use the awscli-login https://github.com/techservicesillinois/awscli-login tool to retrieve temporary AWS access keys using your Cornell netid credentials and Duo (i.e., Cornell Two-Step Login https://it.cornell.edu/twostep). Using temporary access keys associated with an AWS role to authenticate to the AWS Command Line Interface (CLI) https://aws.amazon.com/cli/ is much safer than using fixed AWS access keys tied to an IAM user https://docs.aws.amazon.com/IAM
    - AWS Instance Retirement CloudWatch Event
    - Backup and Archiving to AWS S3
    - Configure Applications to Send Email with SES — On-campus developers are used to having applications send email via appsmtp.mail.cornell.edu. Currently, there is no similar Cornell service in AWS for developers to use. This page shows how to setup the AWS Simple Email Service (SES) to send email from applications.
    - Creating Custom Roles to use With Shibboleth — Cornell AWS account owners can create custom AWS IAM roles and have them linked to Cornell AD so that users with Cornell netids can use Shibboleth to authenticate to AWS and be granted the privileges in the custom role.
        - Cornell Active Directory Groups for use with AWS Shibboleth (Cornell Two-Step Login) — The integration between Cornell CUWebAuth (and Cornell Two-Step Login) and AWS requires Active Directory groups with specific characteristics. When the Cloud Team asks a Cornell AWS account owner to request an Active Directory group from their unit IT support for gating access to roles in the AWS console the following wording can be used in the request.
    - Encrypting AWS RDS Instances
    - How to join an AWS Windows instance to Cornell AD
    - IAM Resources for Limiting AWS Marketplace Access — Some Cornell AWS account owners want to give their teams full access to their AWS account, with the exception of AWS Marketplace. The resources described here can help do that.
    - MyISAM Tables in MySQL RDS Instances
    - Restricting EC2 Actions using Custom IAM Policy — Allow a set of target users to login to the AWS console, and allow them to stop or start only their EC2 instances, based on tag values of the instances.
    - Scheme to Route Traffic to On-Premises and AWS Endpoints — This DNS name and load-balancing architecture shows how to use a Route 53 can be used to send traffic to both an on-premises server and an EC2 deployment. The health checks on Route 53 records (for example-on-prem.xxx.cucloud.net and example-nlb.xxx.cucloud.net) ensure that Route 53 will respond with records that are actually functional.
    - Set Cache Control Header for S3 Object
- Use MFA With AWS Access Keys — Access to some AWS resources require an AWS Security Token Service https://docs.aws.amazon.com/IAM/latest/UserGuide/id_credentials_temp.html session where an MFA key has been used to generate temporary security credentials for an IAM User. This page provides details on how to use MFA and STS together on the command line with AWS CLI (v1) https://docs.aws.amazon.com/cli/latest/userguide/cli-chap-welcome.html.
- Using Certbot and AWS Route53 for TLS Certificates — AWS Certificate Manager https://aws.amazon.com/certificate-manager/ (ACM) provides free TLS certificates for use directly by AWS services (e.g., CloudFront, Application Load Balancer). However, it cannot be used when you need to obtain key and certificate files for use directly on an EC2 instance (e.g., in nginx or apache). Using certbot https://certbot.eff.org/ with Route 53 is an alternative to ACM that can be automated and gives access to the certificate and key files, when the DNS for your s