CloudBank Onboarding Protocol

Appendix 2: Security and Cost Management

CloudBank Onboarding Protocol (parent doc)
CloudBank Onboarding Protocol Appendix 1: Cloud Operations For Researchers

Introduction

Failing to take up security and cost management on the cloud can lead to outcomes that are very unpleasant. These include

- Trying to recover a code base or data store deleted by a well-meaning colleague
- Asking a cloud vendor for a $15,000 refund after bots have hijacked one's cloud account
- Running out of cloud funds by unintentionally spending $20 for every $1 of computation

This document is intended for research teams preparing to use the public cloud (AWS, Azure, GCP, IBM, etcetera) for research. It covers security and cost management on the cloud in the context of research computing. These terms 'security' and 'cost management' are really two sides of the same ***efficient use of the cloud for research*** coin. We don't leave our front door ajar and we don't turn on the heater full blast with the windows open, and so it is with the cloud. We take security measures on the cloud to reduce risk. Here we describe some basic tactical practices together with a conceptual framework for being efficient (i.e. secure and cost-effective) and reducing risk while computing on the cloud.

Three Starting Principles

To start framing 'this is efficient research computing on the cloud' let's identify some principles.

The first principle of efficient cloud use is to recognize that the biggest issue at hand is the *insider threat*. By this we mean research team members new to the cloud who make mistakes that lead to unhappy outcomes. To deal with the insider threat we acknowledge there are cloud learning curves and skills to master; and everyone shares the responsibility to be properly trained for their role.

The second principle of efficient cloud use is the principle of *least privilege*: Don't grant access to resources unnecessarily. This is familiar to anyone who has used a shared computer with multiple User accounts and only one super user. Hence on the cloud we have a distinction in roles: A research team includes people responsible for building research infrastructure (a *builder* role); and then there are team members who use that infrastructure (a *user* role). The

users don't build, so the principle of least privilege gives them access to the work environment but not to the building tools.

The third principle of efficient cloud use we characterize in terms of *zombies* in relation to *garbage collection*. The cloud works on a utility model: The team allocates and uses resources (also called 'services') and is charged proportionally by the cloud provider. A resource that is allocated but not used efficiently–or is not used at all–is a *zombie* resource. Clearing away zombies is a periodic necessity, what we are calling *garbage collection*. This is a central aspect of cloud cost management. It consists largely of two activities: optimization and tag-based tidying up. First (optimization) the research team must learn to use resources efficiently. This means, for example, stopping virtual machines when they are not in use and choosing virtual machine instance types to deliver the best performance per cloud dollar. Second (tag-based tidying), the research team should energetically tag new resources. A tag is a key-value associated with the resource, for example { 'project': 'tree frog call analysis' }. By labeling a storage volume or virtual machine or a cloud database (all resources) with metadata as to purpose, owner, contact information, expiration date, and so on, the research team has a means of evaluating resources that are present but not accounted for. *Tags* facilitate *garbage collection*: And that means deleting resources that are of no use so as to not pay for them.

More Tactics: Best Cloud Efficiency Practices

The following information structure represents about a 10–20% depth dive into a number of common security and cost management topics and practices. This section may look intimidatingly complicated. The counterpoint to that observation is that cloud providers have produced a collection of reference and learning materials.

For research teams without IT-inclined persons

- Investigate whether your institution has support staff on hand
    - In some cases they can help guide your learning process
    - In some cases they can take on a supportive role

For research teams that have questions about cloud computing

- Contact help@cloudbank.org for consulting, networking and referral
- Check the Cloudbank community forum, community.cloudbank.org
- Either directly or through CloudBank: Contact cloud vendors for guidance
- For technical issues: Create a support request from within your cloud account

For connecting to cloud Virtual Machines

- For Linux start with ssh
- For Windows use Remote Desktop Services (RDS) and learn about ADUC

For infrastructure builders

- Learn access credential use in terms of *frequent delete/replace*
    - …in contrast to long-lasting, sacrosanct, etcetera
- Learn about automated services on the cloud
    - Example: Enable account a cloud logging service
        - This creates a digital paper trail of cloud activities
        - This anticipates that something may go awry at some point
        - Cloud logs are invaluable for diagnosing what happened
- Learn about cloud access methods
    - These include the vendor's console, command line interface, and APIs
- CLI base concept: Where to use it from is not 'locally'
    - CLI stands for Command Line Interface
    - This is an interface to the cloud that can be installed on a local computer
    - Use the CLI to allocate, configure and de-allocate resources
        - Analogous to the controls of a radio controlled airplane
    - Installing the CLI on a local computer is possible but discouraged
    - Alternative: Set up a low-cost VM on the cloud with the CLI installed
    - Authenticate into this computer via ssh
    - Your base of infrastructure-building operations is now itself on the cloud
        - This reduces your exposure
    - Alternative: Use a vendor-provided IDE such as AWS' CloudNine service

For research teams using open repositories like GitHub

- Do not place cloud access credentials in open source repositories
    - Deleting credentials from GitHub is an insufficient response…
        - …because GitHub is about version control
    - IDEs such as R-Studio may preserve credentials invisibly
        - Famously R-Studio creates an .rhistory file that may include cloud credentials; and this file can be unintentionally uploaded to GitHub
- Learn how to protect code from accidental overwrites
    - Not all security precautions are cloud-oriented
    - Learn and use Git branch protection to avoid accidental deletion
- Learn to use rejection utilities like .gitignore

For research teams authenticating (logging in) to the cloud

- Use Multi-Factor Authentication (MFA)
- Periodically rotate keys
- Do not work on infrastructure as the root user (account owner)
    - Rather: Create an administrative identity and use that

For research teams using third party utility applications

- A common example is a storage application
    - This allows researchers to move data back and forth between local computers and the cloud via drag-and-drop
- Utility applications must authenticate to work; via credentials
- This creates an opportunity for accidental mismanagement of credentials

Tagging Revisited

- Tag resources automatically when possible
- Treat tag content as ‘Nobody has any idea what this resource was used for’
    - This can help answer the question ‘What information goes into the tags?’

IAM: Identity and Access Management

- Nobody on the research team should ever share credentials
- Ideally choose one responsible individual to manage team identities
    - Ideally a person predisposed to working in the computer idiom
- Devote time to training beyond the basics
- Review how IAM is working for the research team periodically
- Investigate temporary credentials that expire after a time
    - Example: AWS’ Simple Token Service (STS)

Anticipating a crisis

- Think about / learn about the cloud as a programmable entity
- Alerts can be configured to react to unexpected conditions
    - Example: We average 3 but are suddenly running 40+ Virtual Machines
    - Example response: Everybody out of the pool!
- Learn about serverless functions
    - Functions that run in response to some trigger mechanism
    - Are not encumbered by an underlying virtual machine (abstracted away)
    - Serverless functions are designed to be simple and easy to build
        - They can be configured to respond to crisis conditions
            - Example (from above): Every five minutes a serverless function runs that gets a list of all currently running Virtual Machines for the research team’s account. If this list exceeds pre-set expectations: The serverless function issues stop commands for all virtual machines. It then disables User permission to start new virtual machines.

Using cloud funding efficiently

- Understand the basic costs of the major resources you use
    - Virtual machines, block storage, network-attached storage, object storage, database services and compute clusters are the main expense drivers on the cloud
    - Many cloud services prove to be ‘close to free’
- Virtual machines should be stopped when not in use
    - Learn to auto-start/auto-stop virtual machines: Work days only
    - Learn to manually stop/start virtual machines easily
- Pre-emptible instances can cost 20% of on-demand instances
    - ‘instance’ = Virtual Machine
- Make sure your compute task is using all of your VMs compute power
    - Learn to use utilities such as Linux ‘top’
- Take care not to create high-volume zombie block storage (cloud ‘disk drives’)
- Learn how to use object storage tiers in a cost-effective manner
    - When data are not going to be touched: Use archival storage
- Learn to create images of virtual machines
    - These images are snapshots of the entire virtual machine
    - They can be reconstituted on smaller or larger virtual machines
    - This leads to development on a low-cost machine and deployment to a more powerful machine for extended data analysis
- Create a spend estimate, month by month, and track actual spend against this

Open Questions

These questions will either feed this document or the Community Forum corpus. Nothing is off limits here; all questions welcome.

- Does a cost estimates URL created using Cloud X cost estimator persist forever?
    - Rob thinks it probably does; and that it is entirely anonymous
        - “Security through Obscurity” is not a business plan <humor>
- What is the fastest burn data rate for Cloud X?
    - Example: Cloud X accumulates costs hourly
        - On Cloud X I have turned on a Scram service
        - This becomes an Agent for overspend response
            - It necessarily has permissions to stop all
- How do I MASS STOP ALL THE INSTANCES on Cloud X?
    - This is alluded to above; and we want a recipe for each cloud