CloudBank Protocol Zero

A CloudBank memo on success in cloud research computing

CloudBank Onboarding Purpose and Overview

Introduction

This document is intended for NSF-funded research teams working with CloudBank to build and use research computing environments on the public cloud (AWS, GCP, Azure, IBM). Its purpose is to spell out what CloudBank can (and can't) do in support of your research computing on the cloud, and to indicate what you can do to ensure your use of the cloud is efficient, secure and productive. The CloudBank team requests that research team members review this document as a starting point in the cloud adoption process within the CloudBank program. It will help "you the research team members" understand what you are responsible for knowing and doing.

The CloudBank team's mission is to support research computing on the cloud. This is (often) a non-trivial transition for research teams; so CloudBank is working to make the process a success above and beyond the basic task of paying for cloud resources. To this end, CloudBank does not track a research team's activities on the cloud. Rather we help map out tasks, connect with learning resources, anticipate challenges, and respond when a team hits obstacles. This process begins with an 'onboarding call' where we hope to learn about the research program and how cloud computing fits in.

Cloud account management in this context can be seen in light of several subtopics:

- ☐ Cloud access
- ☐ Security
- ☐ Cost management
- ☐ Compute infrastructure: building and use
- ☐ Data: Upload, access, download
- ☐ Cloud services above and beyond compute, storage and networking
- ☐ Paying the bills

CloudBank's model of operation:

- ☐ CloudBank pays the bills and provides cloud access
- ☐ CloudBank also provides support for learning to work on the cloud
- ☐ The research team is ultimately responsible for everything else

CloudBank support begins with online resources:

- ☐ CloudBank helpdesk help@cloudbank.org
- ☐ CloudBank portal cloudbank.org

- [ ] CloudBank community forum community.cloudbank.org

CloudBank also maintains strong positive relationships with the cloud vendors; who in turn are invested in research success.

PoP and Post-PoP

In this document we refer to two time frames: The funded Period of Performance (PoP) and the subsequent time period immediately afterward (post-PoP). We call out the latter as there are usually project-related obligations to maintain for some period of time, as for example can be laid out in a Data Management Plan (DMP).

The balance of this document touches further on the following topics.

- [ ] Cloud management subtopics in further detail
- [ ] Research team approach to cloud computing
- [ ] Period of Performance spend tracking
- [ ] Examples of challenges in cloud computing practice

Some of the terminology used herein may be unfamiliar. The important point is that research teams are encouraged to contact the CloudBank team (help@cloudbank.org) with questions. Responses from CloudBank can include direct answers, coordinated consultation with technical cloud experts, and mapping out courses of learning.

Cloud Management Subtopics: Expanded

For each subtopic we include a checklist: What a research team using CloudBank is responsible for understanding.

Cloud Access

- [ ] Distinction: Infrastructure building versus infrastructure use
- [ ] Five methods of accessing cloud resources
- [ ] CloudBank portal, community forum, and help desk

Research team members can connect with supported cloud consoles through the CloudBank portal. Browser-based access is one of several modes of cloud interactivity. Others include command line interfaces, API libraries for programmatic access, key pair credentials for ssh access to virtual machines, and remote desktop applications. In all cases there is some form of authentication; which touches on the topic of security.

Security

- [ ] Research teams are responsible for cloud account security
- [ ] Obtain and follow security guidelines for cloud platforms
- [ ] ‘Insider threat’ awareness
- [ ] Risk from authentication credentials made public
- [ ] GitHub versioning and .rhistory public credential narratives
- [ ] Failsafe data backup and recovery (ransomware etcetera)

This Google Cloud link to security guidelines is an example of how cloud providers help cloud Users avoid common mistakes. The “first 30 days guardrails” described therein suggest that the first source of risk while operating a cloud environment is from the research team members themselves. That is, it is easy to accidentally delete resources or ‘leave the gate open’ to people who are not involved in the project. While unintentional this is what is meant by ‘Insider threat’.

Cloud security begins with learning good practices like these. As a simple example, cloud resources can and should be tagged with labels that spell out purpose, timeline, responsible individuals, and so on. These tags can be checked prior to deleting the resources.

Security threats such as ransomware attacks are also part of the cloud security landscape. While it may not be necessary to implement, a research team should be aware of a Failsafe approach to data backup. Below is a condensed view of vendor-provided Failsafe guidelines. It presupposes a research team working on a primary cloud account containing valuable data. The Failsafe is a backup of this data that can be used to restore it, should that working data become compromised.

- Create a new ‘Failsafe’ account outside your home organization
- Give only a limited access to the account; with token MFA
- Enable automated logging of account activity
- Create storage space in a “Write Once Read Many” mode
- Establish automatic replication: Main working account storage → Failsafe storage
- On the Failsafe account enforce ‘write only’ policies
- Use archival storage modes on the Failsafe account to reduce storage cost
- Encrypt data in the Failsafe storage
- “Don’t post Access Keys on Github”: Manage credentials as if they are bank passcodes

Cost Management

- [ ] Resources can become un-tracked / abandoned; while still incurring charges
- [ ] Tagging resources helps understand whether they can be terminated/deleted safely
- [ ] On-demand Virtual Machine cost rates are often 3-5 times that of preemptible ones
- [ ] Virtual Machines should be stopped when not in use
- [ ] Virtual environments can be transferred to {lower/higher} {cost/power} virtual machines

- [ ] Both object and block storage costs scale with volume and time duration
- [ ] Object storage rates depend on access needs: Fast access to archival modes

Often in the course of resource allocation (object storage, block storage, virtual machines) a cloud User will go through practice or development stages. This can result in some resources becoming ‘zombies’: They remain allocated to the account (at some cost per unit time) without actually serving any practical purpose. Cost management on the cloud begins with resource management: Knowing how to identify and release zombie resources.

From here there is a second stage of cost management, which is efficient resource use. If a cloud User wishes to take advantage of the reduced price for preemptible virtual machine instances: It is important to understand how to use checkpointing to avoid losing computational results. Learning this methodology can stretch compute dollars on the cloud by a factor of four.

Computing infrastructure

- [ ] Building a research computing infrastructure is the responsibility of the research team
- [ ] Help is available on a consulting basis from CloudBank
- [ ] Help–including technical deep dives–is available from cloud vendors
- [ ] Learning materials are available
- [ ] CloudBank works to identify appropriate and useful learning paths

Computing infrastructure is simply a configuration of resources (virtual machines, storage, access protocols, etcetera) that a research team builds and uses on the public cloud. The ensuing section describes a spectrum of approaches to building computing infrastructure.

Data

- [ ] The form of data storage depends on team data usage patterns
- [ ] ‘What becomes of cloud data after the project Period of Performance?’

Data download from the cloud to the internet (‘data egress’) costs roughly $0.09 per GB. In most typical use cases, however, data egress charges are subject to a cost waiver. In consequence, moderate data egress should result in significant charges. If data egress is a major part of a research team’s planned cloud use: They should consult with CloudBank (help@cloudbank.org) to anticipate what to expect.

Cloud Services Beyond Basics

- [ ] There are many “higher level abstraction” services
- [ ] These services can facilitate data access and sharing
- [ ] Commonly used example: Serverless computing

- [ ] Commonly used example: Hosted database
- [ ] These services can also facilitate data science methodologies like ML

CloudBank curates learning paths, relying primarily on the training resources provided by cloud vendors. This means that a research team interested in making use of advanced cloud services can get training in using those services for a minimal cost.

Paying the Bills

- [ ] The research team is expected to estimate its projected spend (see next section)
- [ ] The research team is expected to work to minimize spend
- [ ] CloudBank takes care of billing and payment
- [ ] CloudBank will periodically check in with the research team on actual spend
- [ ] CloudBank provides the research team with spend notifications, customizable

Research team approach to cloud computing

There is a spectrum of approaches to cloud computing. Research team skills and computational goals typically inform choice of approach. This spectrum includes:

- [ ] Turnkey cloud users plan to make use of pre-built solutions running on the cloud. Setting this turnkey solution up might fall to some other agency.
- [ ] Turnkey-hopeful cloud users: As above with a slight modification: The research team intends to build a turnkey solution themselves, from an existing template or repository. Here we are giving a nod to cases where the solution-building does not go as smoothly as hoped. There may be a need for some technical support.
- [ ] Platform as a Service (PaaS) cloud users are planning to build and operate their research computing environment relying heavily on cloud services, specifically services that go above and beyond the cloud basics of compute power, storage and networking. This link gives a lengthier description of PaaS.
- [ ] Infrastructure as a Service (IaaS) cloud users are working from the basics. They may simply need storage and compute power, “just get me a cursor”. Once having started up and logged into some cloud Virtual Machines they are ready to get to work.

Cloud infrastructure is first built; and then is maintained and/or modified over the course of a project. If a research team member will not be contributing to this infrastructure (i.e. they are primarily a User): CloudBank strongly advises they be provided access to resources rather than more broadly to the full cloud account. This is analogous to the traditional practice of creating User accounts on a computer rather than providing everyone with the root password.

PoP Cloud Spend Management

CloudBank requests that research teams create a table like the one below: Projecting month-by-month cloud spending for reference. The example table shows a 12 month period of performance. The estimated spend is approximate and is in no way binding. The sum of the spend in the example is below budget. This is fine. The purpose of the spend estimate is to allow CloudBank to help research teams maintain awareness of their cloud usage, particularly if spend runs low or high.

Project Name **Phytoplankton Thin Layers**	Principle Investigator **Ellen Smith**	Cloud Budget **$50k**
Month	Estimated Spend	Milestone
1	500	Start
2	1000	
3	2000	
4	5000	
5	6000	functional analysis pipeline completed
6	6000	
7	6000	
8	6000	
9	4000	principal analysis completed
10	4000	
11	4000	exit phase
12	3000	project concludes

CloudBank will periodically check in with each research team via email or call. This informal process is intended to touch base on spending and on possible challenges the team faces in managing their research computing on the cloud. The nominal cadence for check-ins is quarterly.

Cloud Challenge Examples

This section provides some examples of the challenges a research team can face in building and using research computing infrastructure on the cloud.

Example 1: My VM has a wandering ip address

Problem: My cloud VM is automatically stopped every day (intentionally); so I come to work and start it. After it starts up I note that it has a new ip address. I use this ip address to log in. Every day a new address has has become a minor nuisance. A fixed ip address would be much simpler.

Solution: Cloud providers make fixed ip addresses available. They can be associated with a cloud VM; and they will persist through starts and stops of that VM. On AWS this feature is called an Elastic IP. On GCP it is a static external IP. On Azure it is called a public IP address.

Example 2: My VM freezes when running certain commands

Problem: My VM often freezes, requiring me to Stop and Restart it. This happens especially when I am installing software or cloning a source code repository, at which point my terminal stops responding to input and I have to manually close it with the window's 'x' button.

Solution: This is likely because the VM does not have enough RAM to do its tasks. Try moving up to a VM with more memory. If the freezes only happen when installing software, you can try setting the environment up on a mid-weight VM, creating an image from that, and then starting the image on a smaller machine. (The ability to migrate an operating environment from one type of Virtual Machine to another (for example with more power) is an important and commonly useful feature of cloud computing.)

Also: You can exit back to your local terminal when SSH freezes by typing these keys in sequence: [Enter] [~] [.]

Example 3: I can't access web servers running on my VM

Problem: I can use SSH to access my VM just fine, but when I run a web server like flask or jupyter I can't seem to log in through my web browser.

Solution: There are two likely causes of this problem.

1. The incoming port for the server is blocked. Jupyter by default serves its interface through port 8888. Flask serves on port 5000. If you're not sure what port the server is using, it is normally reported as a part of the URL you are told to use when you start the software through the terminal. The port is the number after the initial colon, bolded in the

following example:

```
Running on http://127.0.0.1:5000/ (Press CTRL+C to quit)
```

Using your cloud's portal, add this port to the allowed incoming ports in your VM's firewall.

2. The server is set to only service requests coming from VM itself, rather than external computers. This behavior varies across software, but it usually appears as the server running on "127.0.0.1" or "localhost". See the bolded part of the following example:

```
Running on http://127.0.0.1:5000/ (Press CTRL+C to quit)
```

By changing this value to "0.0.0.0", we instruct the server to accept requests from external sources. The way to do this varies by software and use case, but usually involves changing a setting called "host" or "IP". For Flask, it looks like this:

```
$ flask run --host 0.0.0.0
```

Example 4: My team of four wants to use Jupyter notebooks

Problem "Myself and my three team members want to work in a Jupyter notebook environment; and we have non-trivial processing tasks and a large dataset. It seems like overkill, maintenance maybe? and a lot of learning curve to build out a Jupyter Hub. What are our options here?"

Solution First a Jupyter Hub provides low-cost and typically low-power computing environments to communities, students taking a particular course, or other large groups. The advantage is an automated and uniform experience for the Users. The disadvantage is that the resulting Jupyter environments are not easy to customize. What you are describing often gives rise to customization; so let's set Jupyter Hub aside for the moment. You are right, it is non-trivial to set up. Instead we'll look at some alternative ideas.

First, the skill to install a Jupyter notebook server is pretty minimal. This effort is equivalent for a cloud VM install compared to a personal laptop install. The "extra mile" for the cloud install is to establish an interactive pipeline *from* one's laptop *to* the cloud machine. This involves establishing something called an *ssh tunnel*. But we're not done yet on this path because the real value of a Jupyter Notebook server is in its flexibility and reliability; which will be supported by the high reliability and scalability of the cloud. This is about "Ok my Jupyter Notebook server is up and running; now what?" Both the installation mechanics and the "Now what?" are addressed in detail in this CloudBank Solution.

Each cloud provider hosts their own flavor of Jupyter environment. Colab, Azure Notebooks, SageMaker on AWS, and so on.