INTERNET2

2022 TECHNOLOGY exchange

MOVING FROM CLOUD CHAOS TO STANDARDS

University of California, Office of the President
- Khalid Ahmadzai, Sr. Cloud Engineer
- Matt Stout, Cloud Architect
- Kari Robertson, Executive Director of Infrastructure Services

INTERNET2

ABOUT US

University of California:

- 10 Campuses - undergraduate/graduate
- 6 Academic Health Centers
- 3 National Laboratories
- >230,000 employees
- >280,000 students

University of California, Office of the President (UCOP):

- Systemwide infrastructure services
- Local infrastructure services
- >2000 employees
- >$2M annual cloud provider utility
- > 50 cloud accounts

Khalid Ahmadzai
Sr. Cloud Engineer

Matt Stout
Cloud Architect

Kari Robertson
Executive Director of
Infrastructure Services

CLOUD JOURNEY TO DATE

< 2019	2020	2021	2022
Cloud Exploration	**Assessment**	**Foundations**	**Scaling**
• Training for all • Freeform builds	• Established FinOps • Analyzed Existing Workloads • Easy Optimizations Migrations: • UC Systemwide Payroll System	• Cloud Standards • Automated Compliance • Security Framework Migrations: • Data Center (secondary)	• Central IT Alignment • Retrofits • Intermediate Optimizations Migrations: • UC Systemwide Student Application System • Mainframe Workloads • ECS Cluster Containers

CLOUD CHAOS

CLOUD CHAOS | WHERE WE STARTED

- Strategy to exit data centers and move to cloud
- Formal training for a majority of IT staff
- Exploration is encouraged
- Grow cloud skills through hands-on experience within teams
- Organic growth, minimal boundaries

CLOUD CHAOS | CHALLENGES

- Undefined security framework
- Roles and responsibilities varied by team
- Expertise expected on all cloud services
- Coke vs. Pepsi moments
- Self-managed FinOps
- Keeping up with new cloud offerings
- No coding/deployment standards

How do you scale without standards?

CLOUD CHAOS | IMPROVED STRATEGY

- Establish cloud standards
- Define roles and responsibilities
- Develop deep expertise
- Automate everything
- Ask for help
- Align existing cloud workloads
- Create documentation and share widely
- Don’t reinvent the wheel
- Look at best practices

Scale! Scale! Scale!

CLOUD STANDARDS

CLOUD STANDARDS | IMPLEMENTATION

- Identify current challenges
- Monthly meeting to review
  - Security/Risk Analysis
  - FinOps
  - Future Plans (Infrastructure and Business)
- Alignment matrix – vetted and approved
- Tools
  - Terraform: Infrastructure as code
  - Cloud Custodian: Compliance
  - Many AWS native services

CLOUD STANDARDS | EXAMPLES

- Account Structure
- Account Owner Responsibilities
- Account Creation Process delivered with base offerings
- Backups into dedicated separate account
- Disaster Recovery into designated regions
- Email Management
- FinOps - cost management/analysis
- IAM Policies
- Logging to S3 -> SIEM
- Networking
- Security framework monitored by SecOps
- Tagging for both functional and technical reasons

…….……the list keeps growing

TERRAFORM

HashiCorp
Terraform

TERRAFORM | OVERVIEW

- What is it?
  - https://www.terraform.io/
  - Infrastructure as Code
  - Terraform codifies cloud APIs into declarative configuration files

- What problem did it fix for us?
  - Repeatable deployment of resources consistently across many accounts
  - Standards made easy
    - Once coded, infrastructure is the easier to deploy
    - Differences and missed standards are reduced

TERRAFORM | OVERVIEW

Why did we choose Terraform over other tools?

- To leverage existing knowledge and public examples, modules, and code
- Existing expertise in house
- Multi-cloud
- Cloud providers do have many of their own tools for this
  - Vendors have solutions worth considering; AWS CDK, AWS Quick Start Cloudformation code
  - So there are choices, for us it was Terraform due to our existing code and expertise

The key is to have something like Terraform and use it, standardize on it, and grow it

TERRAFORM | OVERVIEW

How do we use it?

- Deploy all our resources for networking, compute, storage, security, and more
- We use it at the start of all new deployments
- Modules
  - Reduce the amount of code for each implementation through code reuse
  - Standards are easier, less new code each use
- Backend
  - Use a centralized backend such as Scalr, Terraform Cloud, Cloudify, S3
  - These backends control locking, state files, including other features for logging, security and central deployments

TERRAFORM | CARE AND FEEDING

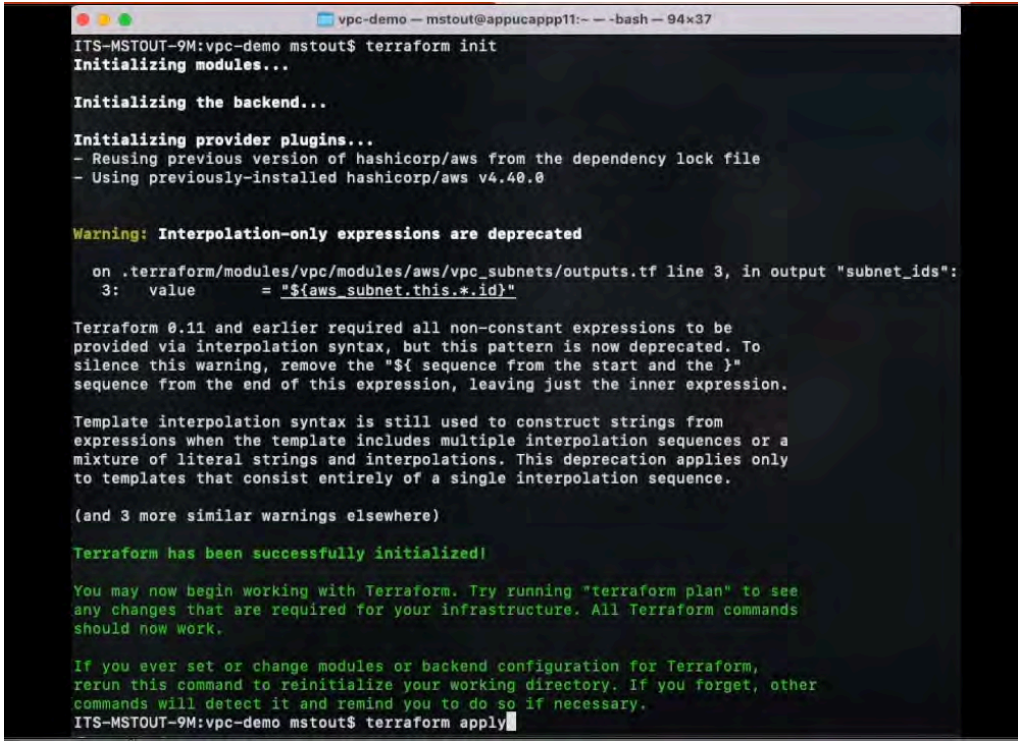
- How did we get people to learn Terraform?
  - Build it and they will come
  - Training resources: https://developer.hashicorp.com/terraform/tutorials (or there are paid trainings and low cost options through udemy, acloudguru)
  - Terraform Registry - a great place to find modules as a starting point

- Is there a review process?
  - Our cloud team schedules code reviews to share knowledge, look for issues, and catch redundancies

TERRAFORM | EXAMPLE CODE

```
locals {
  application = "UCOP Winning Lottery Generator"
  createdBy   = "terraform"
  environment = "prod"
  group       = "cs"
  source      = join("/", ["https://github.com/acme/ucop-terraform-deployments/terraform/UCOPWLG"])
}
module "vpc" {
  source              = "git::https://git@github.com/acme/terraform-modules.git//modules/aws/standard-its-vpc//?ref=v0.0.13"
  application         = "UCOPWLG"
  azs                 = ["us-west-2a", "us-west-2b"]
  cidr_block          = "10.0.0.0/22"
  enabled             = "true"
  environment         = local.environment
  enabled_data_subnets = "true" # change to true to create data_subnet
  enabled_nat_gateway  = "true" # change to true to create nat-gatway
  name                = join("-", [local.application, local.environment])
  tags = {
    "ucop:application" = local.application
    "ucop:createdBy"   = local.createdBy
    "ucop:environment" = local.environment
    "ucop:group"       = local.group
    "ucop:source"      = local.source
  }
}
```

TERRAFORM | DEMONSTRATION



CLOUD CUSTODIAN

INTERNET2

2022 TECHNOLOGY exchange

CLOUD CUSTODIAN | OVERVIEW

- What is it?
  - Cloud Custodian is an open source, stateless rules engine for cloud environments enabling management of resources by filtering, tagging, and then taking action

- What problem did it fix for us?
  - Cost management and orphaned resource cleanup
  - Security enforcement, encryption, compliance, tagging
  - Ensure certain standards are used across all AWS accounts
  - Replace ad-hoc cloud-specific scripts with simpler syntax
  - Very easy to write policies that look for issues
  - Automatically generate tickets in our ticketing system (ServiceNow)

CLOUD CUSTODIAN | OVERVIEW

- Why did we choose Cloud Custodian over other tools?
  - Open source tool
  - Easy to write policies specific to your requirements
  - Continuous deployment of polices to multiple AWS accounts

- Policy review and deployment process:
  - Development/testing of new policy
  - Vetting by cloud team
  - Announcement and document
  - Deploy to production (re-evaluate as needed)

CLOUD CUSTODIAN | IN ACTION

How do we use it?

- Inform -> ServiceNow -> Owner Takes Action
  - Example: Missing tags
- Warn -> ServiceNow -> Cloud Custodian Takes Action (14 days)
  - Example: Orphaned volumes
- Cloud Custodian Takes Action (Immediate)
  - Example: Missing encryption

CLOUD CUSTODIAN | POLICY REPOSITORY

- Ensure organization cloud standards are followed
- Enforce security (ex. access, authorization, encryption, logging)
- Detect/notify root login – unauthorized user detection
- Notify and purge orphaned/abandoned resources
- Alerts owner of missing tags used for compliance and cost management
- Scheduled system availability - turn off non-production environments nights/weekends

CLOUD CUSTODIAN | SAMPLE POLICY

```
---
policies:
  - name: s3-auto-encryption
    actions:
      - type: set-bucket-encryption
        enabled: true
      - type: auto-tag-user
        tag: ucop:createdBy
      - type: notify
        template: default-ucop.html
        template_format: "html"
        priority_header: "1"
        subject: "Account: {{ account_id }} - {{account}} - {{ region }} Encrypted the S3 bucket"
        violation_desc: "The following S3 bucket has been encrypted post creation"
        action_desc: "The following S3 bucket has been encrypted post creation"
        slack_template: custodian-slack-s3-encryption-report
        to:
          - EmailAddress
        transport:
          type: sqs
          queue: https://sqs.us-west-2.amazonaws.com/AccountNumber/cloud-custodian-mailer-queue
    comments: |
      This policy is triggered when a new S3 bucket is created and it applies
      the AWS AES256 Default Bucket Encryption.
    description: "Auto encrypt S3 bucket"
    filters:
      - "tag:ucop:encrypton_approval": absent
      - type: bucket-encryption
        state: False

    mode:
      type: cloudtrail
      events:
        - CreateBucket
      role: "arn:aws:iam::{account_id}:role/CloudCustodianLambdaRole"
      tags:
        ucop:application: custodian
        ucop:createdBy: custodian
        ucop:environment: prod
        ucop:group: chs
      timeout: 200
    resource: s3
```

REFLECTIONS

LESSONS LEARNED

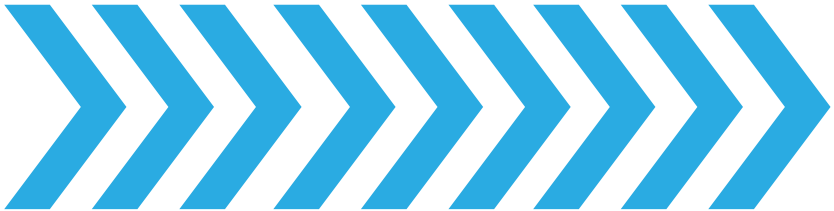
- Acknowledge reality of current state
- Establish cloud standards to scale efficiently
- Security framework is only as strong as your compliance
- Cost management can identify technical mistakes
- Cloud strategy needs an execution plan
- Communication is key to cloud standard adoption
- Notification + Action must be used carefully
- Advertise accomplishments, highlighting the 'what' and 'why'

BIGGEST WINS

- Improved compliance and consistency
- Cost optimization
- Increased capacity enables teams to do more with less (ex. account creation)
- Enabling developers to develop (vs. manage infrastructure)
- Granular FinOps – visibility into actual costs of services
- Terraform modules – reusable code/configuration

WHAT'S NEXT

- Implement more policies
- Increase number of policies that notify then take action
- Improve IAM roles to align access with responsibilities
- Retrofit existing workloads to match cloud standards
- More marketing of our accomplishments!



QUESTIONS

Khalid.Ahmadzai@ucop.edu
Matt.Stout@ucop.edu
Kari.Robertson@ucop.edu