

Cornell Red Cloud: Campus-based Hybrid Cloud

Steven Lee

Cornell University Center for Advanced Computing

sh11@cornell.edu



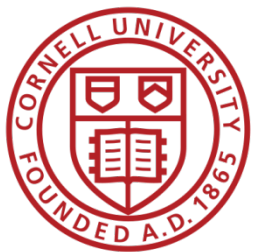


Cornell Center for Advanced Computing (CAC) Profile

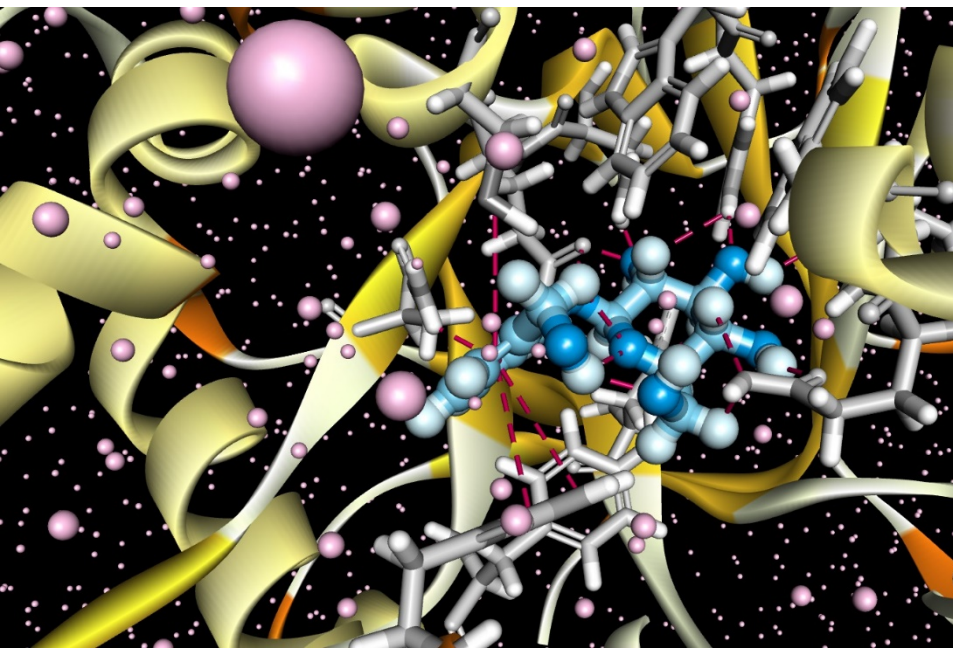


CAC mission, impact on research

- Research computing and consulting services
 - Mission: accelerate discovery and broaden impact
- Wide range of services
 - HPC cluster maintenance and storage to data management, programming, and visualization
- Impact on research
 - Supporting Cornell faculty with over \$100 million in research funding from NSF, NIH, USDA, DOE, NASA ...
 - Management roles in national cyberinfrastructure program
 - NSF Computer, Information Science & Engineering Advisory Committee



Red Cloud & Aristotle Cloud Federation Overview



Red Cloud overview

- Launched 2011
- Motivation: for workloads not suitable for HPC batch queues
- Features
 - AWS-compatible API
 - Compute: EC2
 - Storage: object storage (S3); block storage (EBS)
 - Networking: elastic IP/security groups
 - Accounts and access management: IAM
 - 10 Gigabit Network Interconnect
 - No CPU/RAM over-subscription
 - Subscription model
 - Limits accidental research budget overruns



Red Cloud infrastructure

- Cloud stack: Eucalyptus
- 2 locations
 - Cornell's main Ithaca campus
 - Weill Cornell Medicine NYC
- Compute: 472 CPU cores
 - Users can choose from instance types up to 28 cores/192GB RAM
- Storage
 - Storage in Ithaca cloud is hosted in a Ceph cluster with ~1 PB capacity
 - Storage in NYC cloud is hosted in a Dell SAN
- Networking
 - 10 Gbit Ethernet interconnect between cloud components
 - Each cloud has a 10 Gbit uplink to Cornell campus network

On-demand scalable infrastructure, deployed in minutes

- Red Cloud gives users a fully customizable computing resource
 - Many instance sizes to choose from
 - Get root access to instances
 - Allocate block storage in increments of GB
 - Define network access policies via security groups
 - Managing cloud resources via web console, command line client, API
- Application examples
 - Develop/test code and burst production workload to AWS if necessary
 - Web portals
 - Interactive workloads
 - Software as a service: MATLAB MDCS Cluster
 - Virtualize center's internal infrastructure: web portals, file servers, Nagios monitor, etc.

Aristotle Cloud Federation

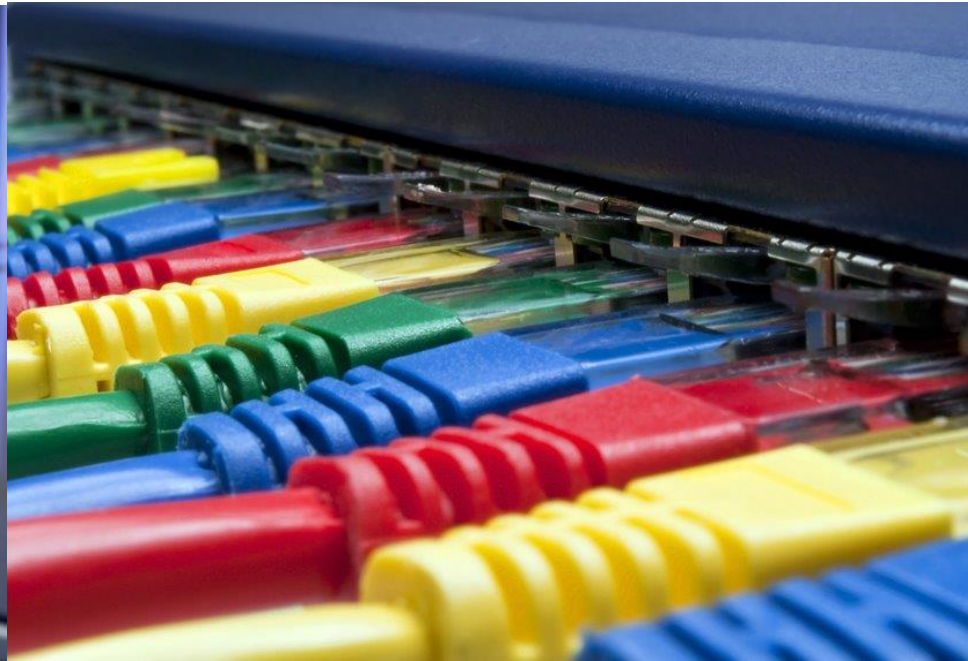
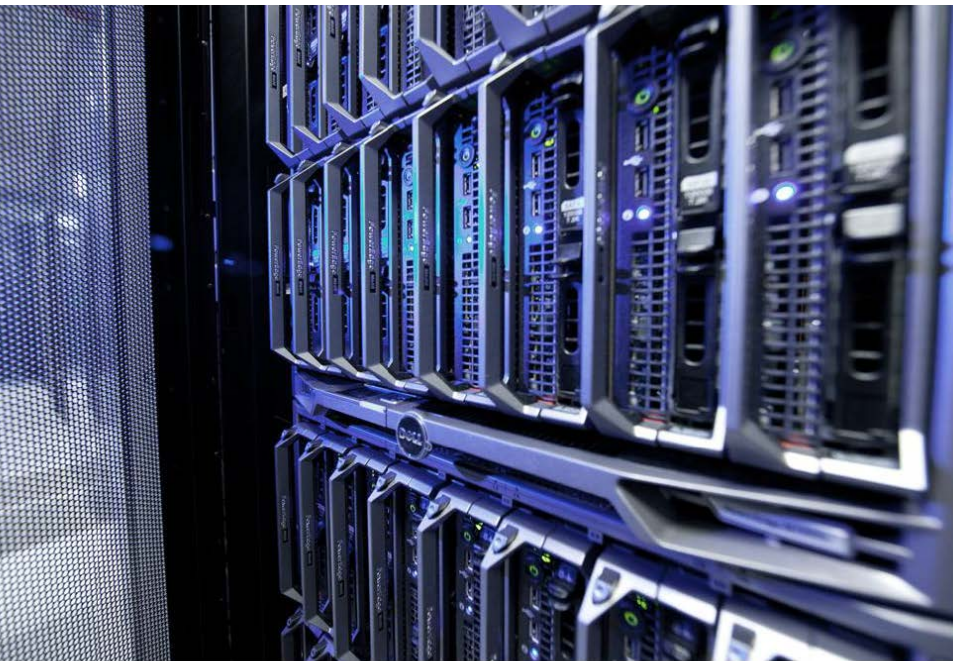
- NSF CC*DNI DIBBs project (2015-2020)
 - Cornell (PI); University at Buffalo, UC Santa Barbara (co-PIs)
- Federated cloud model goals
 - Optimize time to science
 - Share resources “fairly” among institutions
 - Cross-institution allocations
 - Burst to remote federated cloud sites or public cloud during peak usage
 - Open XDMoD cloud monitoring and metrics





Layered Security

Cloud perimeter, stack & instances



Cloud perimeter security

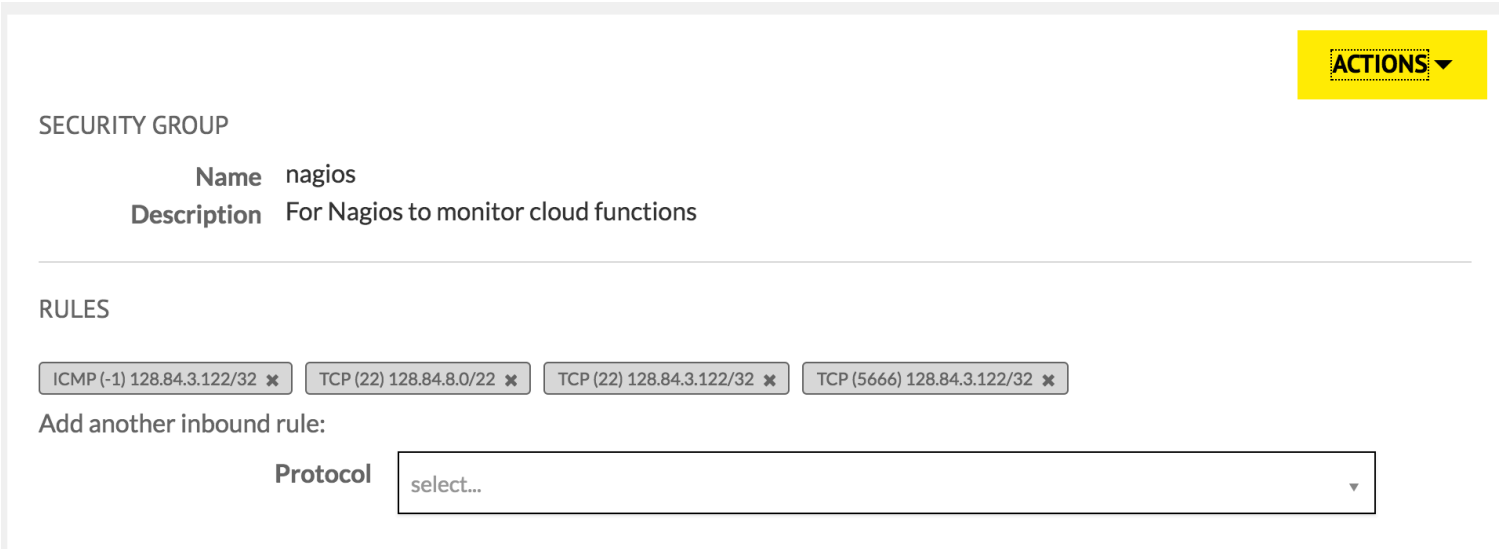
- Red Cloud installations are located in central data centers in Ithaca and NYC on their own subnets
- Leverage central IT security for
 - Network access control by campus firewall
 - Restricts network access to Red Cloud infrastructure
 - Cloud controllers, storage, nodes running cloud instances
 - Allows unrestricted network access to cloud instances
 - Cloud instances are protected by user-defined policies for its security group per AWS architecture
 - Monitoring
 - Red Cloud subnet is monitored like a network in the data center
 - Security incident reporting and handling
 - CAC systems staff serves as liaisons between IT security and users

Cloud stack security features

- Red Cloud accounts are integrated with CAC's Active Directory and fully automated account management system
 - PIs and their proxies can add and remove users from their projects
- Eucalyptus Web Console supports InCommon via Globus Auth
 - Enable SSO by all Aristotle Cloud Federation users
- Access to cloud resources and quotas can be defined on per-user basis by PIs via AWS-styled IAM policies

Cloud stack security features (cont.)

- Users can define network access policies for their instances
 - Each instance is placed in a security group during startup
 - Users defines network access policies for the security group
- Default policy denies all inbound access
 - Users must explicitly grant access by protocol, IP address and range
 - User education and easy-to-use management GUI are critical
 - “Why can’t I ssh into my new instance?”



The screenshot shows a web interface for managing a security group. At the top right, there is a yellow button labeled "ACTIONS" with a downward arrow. Below this, the "SECURITY GROUP" section displays the name "nagios" and the description "For Nagios to monitor cloud functions". A horizontal line separates this from the "RULES" section. Under "RULES", there are four existing inbound rules, each in a grey box with a close button (x): "ICMP (-1) 128.84.3.122/32", "TCP (22) 128.84.8.0/22", "TCP (22) 128.84.3.122/32", and "TCP (5666) 128.84.3.122/32". Below the rules, there is a text prompt "Add another inbound rule:" followed by a "Protocol" label and a dropdown menu currently showing "select..." with a downward arrow.

Securing cloud instances

- Base Linux and Windows images are maintained by CAC staff
- Linux
 - Initial root access granted by user-specified ssh keypairs
 - Password logins are not allowed in base images
- Windows
 - Can configure Windows images to auto-join domains at startup
- Users are encourage to patch their running instances periodically

Lessons learned

User education and communications are critical

- Capacity planning
 - Forgot to shut down instances? Need new cloud capacity? Time to burst to public cloud?
- Liaison between central IT security and user
 - Who's generating this traffic and why?
- Advocating to enable research
 - Analyzing Twitter data during Super Bowl overwhelms Internet link