VPC and Subnet Layout

“A Virtual Private Cloud (VPC) is an on demand configurable pool of shared computing resources allocated within a public cloud environment, providing a certain level of isolation between the different organizations using the resources.” (Wikipedia)

VPC organization and subnet layout are important considerations in setting up a cloud infrastructure to deploy services. There are many things to consider:

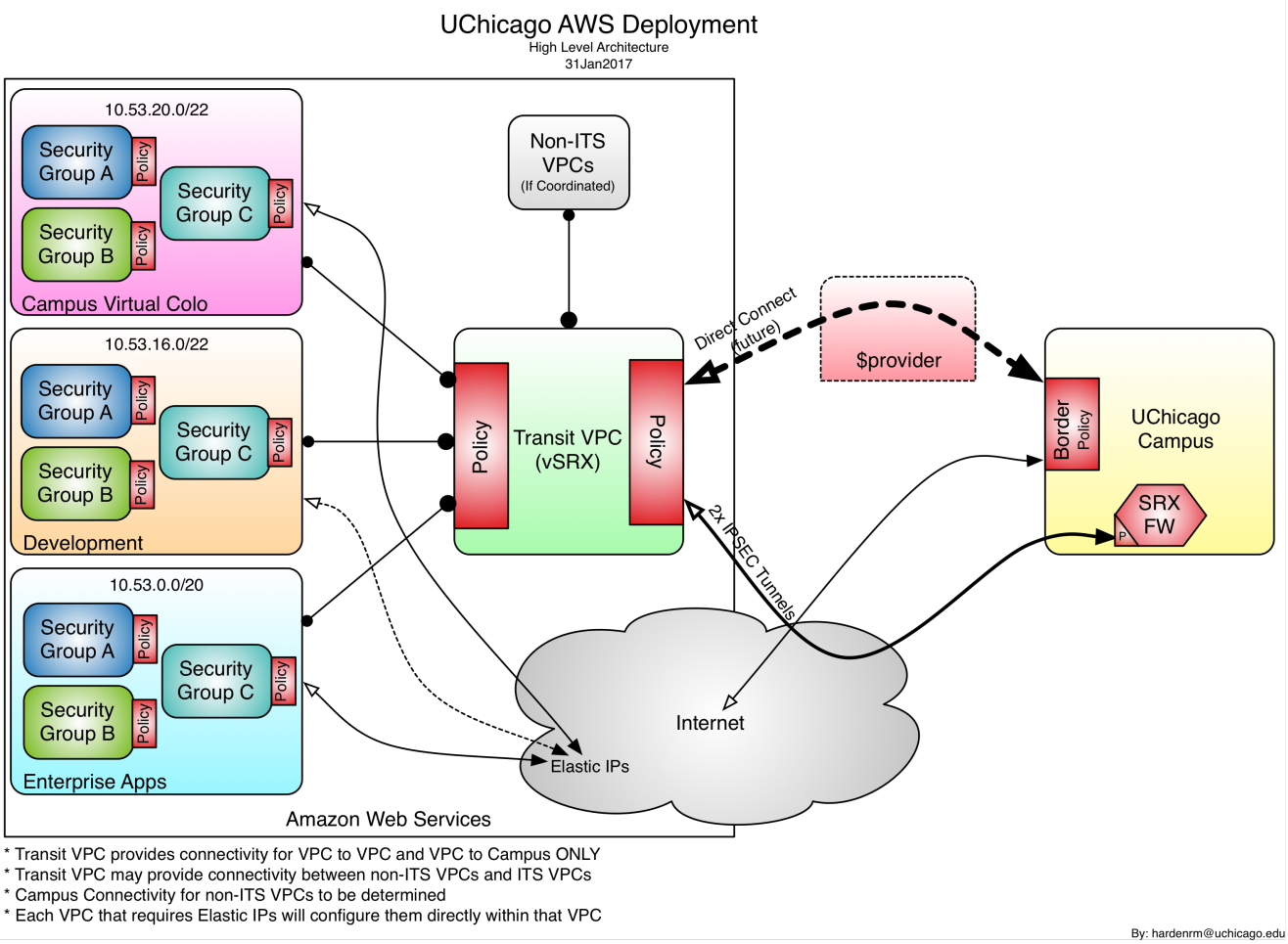
- Number and purpose of VPCs
- Subnetting, supernetting, and IP address allocation within VPCs
- Overall architecture of VPCs (Hub and spoke shared services to isolated applications)
- Use of VPCs as an isolation boundary
- The effect of VPC and subnet design of security groups, routing, and NACLs
- Choke points for data capture/inspection if desired
- Campus/AWS connectivity and its effect on VPCs

Examples

- University of Chicago Example
- Notre Dame Example
- University of Michigan Example
- Carnegie Mellon University Example
- Cornell University Example

University of Chicago Example

This is a preliminary diagram of the structure UChicago IT Services is standing up at AWS, as of February 2017.



Notre Dame Example

Notre Dame has a largely centralized IT group. For this reason, ND has adopted an approach of using a traditional “datacenter” like VPC to host most of its applications and a shared services VPC to provide centrally managed services to distributed IT groups. The following diagram represents the VPC structure and IP address allocation:

blocked URL

Notre Dame manages all IP addresses, on campus and AWS, centrally. The network engineering group is authoritative for all IP address space both public and private. It's critical to have non-overlapping IP space. They have also found that in AWS careful allocation of IP space can result in greatly simplified security groups and rules.

Using the consolidated application VPC as an example, a security group for database access can be written as a permit for 1521, 1433, etc. from 172.22.64.0/19 (the whole VPC) in one rule. Shared services can permit access to defined services by permitting access to 172.22.0.0/16 (the entire AWS class B that ND has allocated). Finer grain access can be done with application specific security groups or host access controls. Subnets allow for the use of route tables and NACLs which provide an additional layer of security, however, NACLs are used very sparingly. NACLs are stateless and as such are a coarse control.

ND has firewalls on the on-premises side. They allow free routing of traffic and database protocols between peer DB layers. The Campus exposure layer is directly accessible from campus, subject to security groups that primarily restrict connections to http/https. Access from the campus exposure layer to on-premises datacenter services is pretty broad with peer layers. Access from World accessible tier is controlled with fine grained rules to on-premises resources.

Separate accounts and VPCs are used to isolate/group departmental infrastructure or specialized applications. Using a parent account with linked accounts for departmental infrastructure simplifies billing and IAM. Some specialized applications and tools need the ability to modify IAM or create IAM roles. This may not be appropriate in traditional production accounts.

University of Michigan Example

The networking group reserved a network segment of Private IP addresses that are reserved for use solely in AWS and our VPN. They set up the VPC to use that reserved IP space so that access to the AWS infrastructure would be similar to that of our on-premise infrastructure. Other groups on campus within the M Cloud service have full control over their VPC's IP space.

Firewalling can be done on either side of a VPN tunnel (if established), and via traditional CIDR blocks without a VPN on either side as well.

In M Cloud, each "customer" has their own AWS account and there is no inherent relationship between accounts. Eventually, they expect to either launch new customer accounts into a secured area in our central VPC or explore a multi-account configuration, utilizing something like VPC Peering. Apart from customer account isolation, there has not been a compelling case to use VPC architecture as a security mechanism.

Carnegie Mellon University Example

CMU currently does not have anything deployed in AWS so this is mostly what is being planned for experimentation . The first pass approach is to throw what is currently being done in the cloud and see what parts fit and what may need to be retooled.

IP address management is handled centrally and the plan is to have distinct subnets in AWS. The initial subnet layout is likely to mirror what is being done in the campus data center. CMU does not allow unrestricted traffic between similarly scoped subnets across different locations (e.g. a DMZ network in location A doesn't have any default access to the DMZ network in location B).

A challenge is the management of the firewall rules between on-premise (Cisco ASA) and the firewall-type capability that exists in the cloud. There has been a reasonable amount of talk from various vendors (especially in the SDN micro-segmentation context) about moving to “application” or “service” profiles that are independent of any specific implementation. They’re hoping this gets more traction and may provide a solution for this issue.

They are considering having a ‘gateway VPC’ where traffic will be funnelled for tunneling back to campus and for Internet egress. Packet inspection could be performed in this gateway VPC. There is still a need to investigate the implementation implications of doing this.

Handling networking for other departments is currently still somewhat in the future. They expect that, at least for AWS, there will be a need to maintain separate customer accounts for the sake of billing. Because accounts that don’t have managed IPs may not be able to connect back to campus, they will likely want to eventually handle the IP address management for them. We also think having another, possibly different, ‘gateway VPC’ for campus traffic /tunneling may be the way to go too. This may provide a simple way for departments to connect back to campus by just doing a VPC peer with the gateway.

Unfortunately, as of this writing, the ‘gateway VPC’ is not something available with AWS. While you can peer with another a VPC, you currently cannot transit through a VPC. So if you have VPC A peered with VPC B and VPC B has an internet gateway and a home VPN tunnel, VPC A can use services that exist in VPC B but cannot pass traffic through VPC B to the internet or back to campus. One current workaround is to create an instance in VPC B and run your own proxy software to do the traffic passing, which is a rather ugly solution. Another option is to extend systems that need to access campus systems into a “shared services VPC.” Peering between VPCs will allow all VPCs to access that service.

Cornell University Example

See: https://blogs.cornell.edu/cloudification/2016/04/08/the-cornell-standard-aws-vpc/

blocked URL