CS 524 Introduction to Cloud Computing

Dharmit Viradia

Homework 4

Prof. Igor Faynberg

1) Find out the exact number of all top domain names. Make sure you put a date and time of your finding.

Sol: This list of Internet top-level domains (TLD) contains top-level domains, which are those domains in the DNS root zone of the Domain Name System of the Internet. The official list of all top-level domains is maintained by the Internet Assigned Numbers Authority (IANA) at the Root Zone Database. IANA also oversees the approval process for new proposed top-level domains. As of June 2019, the root domain contains 1530 top-level domains, while a few have been retired and are no longer functional.

(Reference: https://en.wikipedia.org/wiki/List_of_Internet_top-level_domains)

2) Experiment with http://whois.domaintools.com (and also take a look at www.internic.net) and

a. Find the information about the stevens.edu domain as well as the domain of some other school (for instance, the school you had studied at before you came to Stevens). Who are the administrative contacts for the domains listed there?

b. Now, what happens when you try to find the administrative contact for the .xxx domain? Explain what you have found.

Sol: a. http://www.stevens.edu/
Administrative Contact:
Domain Name Administration
Stevens Institute of Technology
Information Technology
Castle Point on the Hudson
Hoboken, NJ 07030
USA
+12012165457

webmaster@stevens.edu

http://www.atharvaeducation.com/
Admin Name: Nainesh Desai
Admin Organization: Secure Net
Admin Street: c/3 Vijay Park society Mathiradas ext road, kandivli (w)
Admin City: Mumbai
Admin State/Province: Maharashtra
Admin Postal Code: 400067
Admin Country: IN
Admin Phone: +9102228652564
Admin Email:
naineshddesai@gmail.com

b. Administrative contact details on mentioned websites for .xxx domain. xxx is a new Sponsored Top-Level Domain (sTLD), specifically designed for the benefit of the global online adult entertainment industry. It will work alongside the other existing ICANN accredited TLDs such as .com, .net, .org, etc. and offers significant benefits and advantages to both providers and consumers.

Google.xxx

Domain Name: GOOGLE.XXX
Registry Domain ID: D29317-AGRS
Registrar WHOIS Server:
Registrar URL: http://www.markmonitor.com
Updated Date: 2018-10-30T09:36:37Z
Creation Date: 2011-12-01T21:25:32Z
Registry Expiry Date: 2019-12-01T21:25:32Z
Registrar Registration Expiration Date:
Registrar: MarkMonitor Inc.
Registrar IANA ID: 292
Registrant Organization: Google Inc.
Registrant State/Province: CA
Registrant Country: US
Name Server: NS3.GOOGLEDOMAINS.COM
Name Server: NS1.GOOGLEDOMAINS.COM
Name Server: NS2.GOOGLEDOMAINS.COM
Name Server: NS4.GOOGLEDOMAINS.COM
DNSSEC: unsigned

(Reference: www.internic.net, http://whois.domaintools.com)

3) a. Look up www.cs.stevens.edu https://network-tools.com/nslookup/ with different options and explain all the entries in the responses.

b. Then use the returned CNAME entry to find the exact IP address. (Now, just for fun, do the reverse DNS lookup using the services of the http://dnsquery.org and find the geographic location of the host!)

c. Does Stevens specify IPV6 addresses to any of its hosts? Does Google?

Sol: a. Domain: www.cs.stevens.edu

Name	TTL Until Refresh	Class	Type	Data
www.cs.stevens.edu.	739	IN	CNAME	www.cs.stevens-tech.edu.

Name: It is the name of the domain that we searched for.
Class: IN mean Internet
CNAME: A Canonical Name (abbreviated as CNAME) record is a type of resource in the Domain Name System (DNS) used to specify that a domain name is an alias for another domain, which is the “canonical” domain. All information, including sub-domains, IP addresses, etc. are defined by the canonical domain. When a DNS resolver encounters a CNAME record while looking for a

regular resource record, it will restart the query using the canonical name instead of the original name. The CNAME record for the website we look up i.e. www.cs.stevens.edu is www.cs.stevens-tech.edu. This means that www.cs.stevens-tech.edu is the CNAME or Canonical Name for www.stevens.edu.

Time To Live (TTL): It has TTL (Time to Live), which is a setting for each DNS record that specifies how long a resolver is supposed to cache (or remember) the DSN query before the query expires and a new one need to be done. In www.cs.stevens.edu case, TTL is 3600s, this means that the data packets from www.cs.stevens.edu can remain in the network for 3600s.

NS: Name Server is a computer hardware or software server that implements a network service for providing responses to queries against a directory service. It translates an often humanly meaningful, text-based identifier to a system-internal, often numeric identification or addressing component. For cs.stevens.edu, it has 3 name servers i.e. nrac.stevens-tech.edu, drdns2.stevens.edu, and sitult.stevens-tech.edu. These all three of them for a purpose of redundancy, so that in case of failure of one of them, the website does not shutdown and is still reachable through other active name servers. Each of them has a TTL or Time to Live of 3600s or 1 hour.

A: A refers to Address. An A record or Address record maps a domain name to the IP address (IPv4) of the computer hosting the domain. A record is used to find the IP address of a computer connected to the internet from the name. It returns 32-bit IPv4 address, most commonly used to map hostnames to IP address of the host, but it is also used for DNSBLs, storing subnet masks. Here the hostname drdns2.stevens.edu, is mapped to an IP address 155.246.248.20, with a TTL of 600s or 10mins.

b. The returned CNAME entry to find the exact IP address is depicted below: (http://network-tools.com/nslook/Default.asp?domain=www.cs.stevens-tech.edu)

Name	**TTL Until Refresh**	**Class**	**Type**	**Data**
www.cs.stevens-tech.edu.	594322	IN	A	155.246.56.11

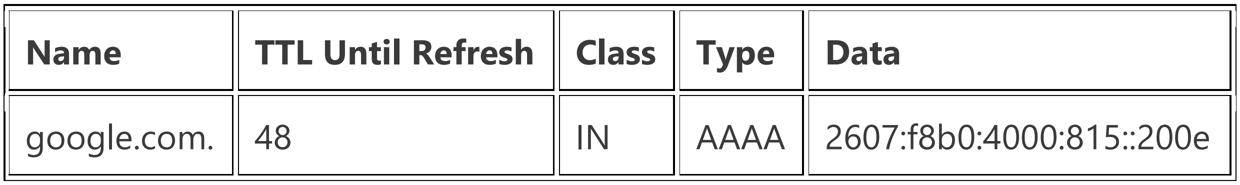
After Reverse DNS lookup (https://dnsquery.org/ip2location/155.246.56.11), the geographic locations and the IP address for the host I found is

Country : United States (US)
Region : New Jersey
City : Hoboken
Latitude : 40.745800018311
Longitude : -74.032096862793

c. No, Stevens does not specify IPv6 address to any of its hosts and Yes, Google specifies IPv6 addresses to its hosts.

DNS Records for: 'cs.stevens.edu'
Returned Data
None

DNS Records for: 'google.com'
Returned Data



Name	TTL Until Refresh	Class	Type	Data
google.com.	48	IN	AAAA	2607:f8b0:4000:815::200e

(Reference: http://network-tools.com/nslook/, https://en.wikipedia.org/wiki/CNAME_record, http://dyn.com/blog/dyntech-everything-you-ever-wanted-to-know-about-ttls/, https://support.dnsimple.com/articles/a-record/, https://dnsquery.org/reversedns/www.cs.stevens-tech.edu)

4) Find your PC's IP address (preferably at home, if you have an Internet connection there.) Can you find your domain with the reverse look up? If you can, what is the domain name? If you cannot, explain why.

Sol: The private IP of my PC's is 192.168.1.159
Hence, it is my private IP so the domain with reverse lookup is showing is

```
Server:  G3100.myfiosgateway.com
Address:  192.168.1.1

Name:    Dharmit-HPEnvy
Address:  192.168.1.159
```

I found my public IP on google by typing "whats my ip" i.e. 71.255.89.254 and the domain it is showing with reverse lookup is

```
Server:  G3100.myfiosgateway.com
Address:  192.168.1.1

Name:    pool-71-255-89-254.nwrknj.east.verizon.net
Address:  71.255.89.254
```

5) Research the responsibilities and structure of IANA (www.iana.com) and ICANN (www.icann.com). What are the differences in responsibilities between these two organizations? Search the web for the information and then describe the controversy in ICANN concerning Whois?

Sol: **Structure and Responsibilities of IANA**

The Internet Assigned Numbers Authority (IANA) is a department of ICANN, a nonprofit private American corporation. IANA allocates and maintains unique codes and numbering systems that are used in the technical standards ("protocols") that drive the Internet. Their various activities can be broadly grouped in to three categories:

- Domain Names: Management of the DNS Root, the .int and .arpa domains, and an IDN practices resources.
- Number Resources: Co-ordination of the global pool of IP and AS numbers, primarily providing them to Regional Internet Registries.
- Protocol Assignments: Internet protocols' numbering systems are managed in conjuction with standards bodies.

Structure and Responsibilities of ICANN

ICANN is made up of a number of different groups, each of which represent a different interest on the internet and all of which contribute to any final decisions that ICANN's make. There are three supporting organizations those deals with IP addresses, domains names, and manage of country code top-level domains. There are four advisory committees that provides with advice and recommendations. And finally, there is a Technical Liaison Group which works with organizations that devise the basic protocols for internet technologies. The roles of ICANN are:

- It includes the consideration and implementation of new TLDs and the introduction of IDNS'.
- It coordinate the global Internet's system of unique identifiers, and in particular to ensure the stable and secure operation of the Internet's unique identifier systems.
- It formalize relationships with root name server operators.
- It ensures appropriate contingency planning; maintain clear processes in root zone changes.
- It maintains and improves multi-stakeholder model and the global participation of all stakeholders, and will continue to further the effectiveness of the bottom-up policy development processes.
- It implements appropriate mechanisms that foster participation in ICANN by global Internet stakeholders, such as providing educational services and fostering information sharing for constituents and promoting best practices among industry segments.
- It shall conduct a review of, and shall make necessary changes in, corporate administrative structure to ensure stability, including devoting adequate resources to contract enforcement, taking into account organizational and corporate governance best practices.

Differences in responsibilities between IANA and ICANN

- IANA is the institution which runs TLDS whereas, ICANN based on the Memorandum of Understanding (MoU), is the institution which runs IANA.
- IANA runs Top-Level Domains and manages the task of IP address and ranges, ports, and other related characteristics whereas, ICANN is a non-profit association that coordinate Internet's worldwide space framework.

Controversy in ICANN concerning Whois

Internet regulators are pushing a controversial plan to restrict public access to WHOIS Web site registration records. Proponents of the proposal say it would improve the accuracy of WHOIS data and better protect the privacy of people who register domain names. Critics argue that such a shift would be unworkable and make it more difficult to combat phishers, spammers and scammers.

A working group within The Internet Corporation for Assigned Names and Numbers (ICANN), the organization that oversees the Internet's domain name system, has proposed scrapping the current WHOIS system — which is inconsistently managed by hundreds of domain registrars and allows anyone to query Web site registration records. To replace the current system, the group proposes creating a more centralized WHOIS lookup system that is closed by default. According to an interim report (PDF) by the ICANN working group, the WHOIS data would be accessible only to “authenticated requestors that are held accountable for appropriate use” of the information. (Reference: https://en.wikipedia.org/wiki/Internet_Assigned_Numbers_Authority, https://www.iana.org, https://www.icann.org/resources/pages/what-2012-02-25-en, http://www.securityweek.com/icanns-rolling-controversyverification-whois-registration-data)

6) The Spam Haus attack
a. Read https://www.isc.org/blogs/is-your-open-dns-resolver-part-of-a-criminalconspiracy-2/. Describe (in no more than a couple of paragraphs) the Spam Haus attack and explain the dangers of open recursive resolvers.
b. Find out (you will need to search the Web and sort a lot of information out!) how cloud services were used to mitigate this attack.

Sol: Spam Haus was the target of a distributed denial of service (DDoS) attack exploiting a long-known vulnerability in the Domain Name System (DNS) which permits origination of massive quantities of messages at devices owned by others using IP address spoofing.

1. Spamhaus is coming from a technique that has been known for years — a variety of reflection attack commonly known as a “DNS amplification attack. As an industry leader in the field of DNS software, ISC sees the Spamhaus DDOS as a perfect opportunity to remind DNS operators why it is important to not operate an “open” recursive resolver, a policy recommendation we have been making since 2005. The attacker sends a DNS query a few bytes in size to an open resolver, forging a “spoofed” source address for the query. The open resolver, believing the spoofed source address, sends a response which can be hundreds of bytes in size to the machine it believes originated the request. The end result is that the victim’s network connection is hit with several hundred bytes of information that were not requested. They will be discarded when they reach the target machine, but not before exhausting a portion of the victim’s network bandwidth. And the traffic reaching the victim comes from the open resolver, not from the machine or machines used to initiate the attack. Given a large list of open resolvers to reflect against, an attacker using a DNS amplification attack can hide the origin of their attack and magnify the amount of traffic they can direct at the victim by a factor of 40 or more. DNS operators who operate open resolvers without taking precautions to prevent their abuse generally believe they are harming nobody, but as the Spamhaus DDOS proves, open resolvers can be effortlessly coopted by attackers and used in criminal attacks on third parties.

2. Beginning on March 18, the Spamhaus site came under attack. The attack was large enough that the Spamhaus team wasn’t sure of its size. It was sufficiently large to fully saturate their connection to the rest of the Internet and knock their site offline. These very large attacks, which are known as Layer 3 attacks, are difficult to stop with any on-premise solution. Spamhaus's blocklists are distributed via DNS and there is a long list of volunteer organizations that mirror their DNS infrastructure in order to ensure it is resilient to attacks. The website, however, was unreachable.

   Very large Layer 3 attacks are nearly always originated from a number of sources. These many sources each send traffic to a single Internet location, effectively creating a tidal wave that overwhelms the target's resources. In this sense, the attack is distributed (the first D in DDoS

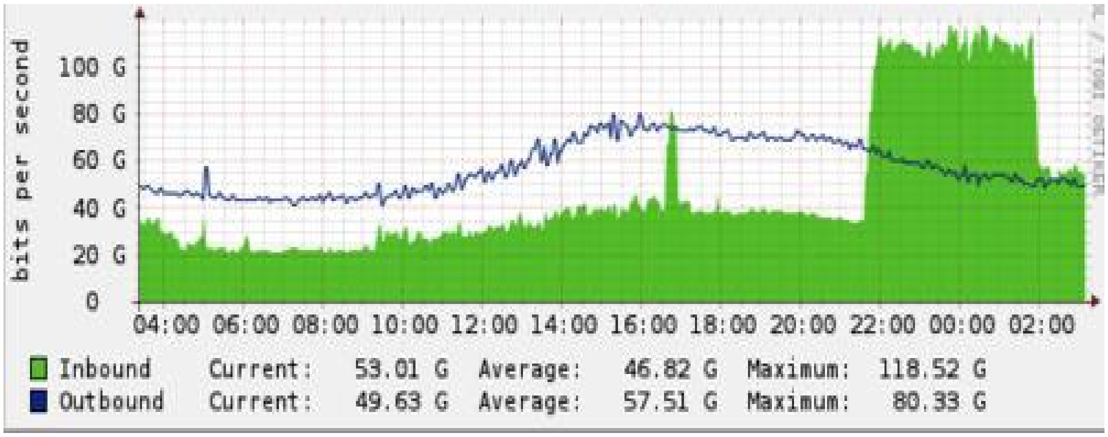
-- Distributed Denial of Service). The sources of attack traffic can be a group of individuals working together (e.g., the Anonymous LOIC model, although this is Layer 7 traffic and even at high volumes usually much smaller in volume than other methods), a botnet of compromised PCs, a botnet of compromised servers, misconfigured DNS resolvers, or even home Internet routers with weak passwords.

Since an attacker attempting to launch a Layer 3 attack doesn't care about receiving a response to the requests they send, the packets that make up the attack do not have to be accurate or correctly formatted. Attackers will regularly spoof all the information in the attack packets, including the source IP, making it look like the attack is coming from a virtually infinite number of sources. Since packets data can be fully randomized, using techniques like IP filtering even upstream becomes virtually useless.

On March 19, 2013 afternoon, CloudFlare was contacted by the non-profit anti-spam organization Spamhaus. They were suffering a large DDoS attack against their website and asked if we could help mitigate the attack.

CloudFlare immediately mitigated the attack, making the site once again reachable. (More on how we did that below.) Once on our network, we also began recording data about the attack. At first, the attack was relatively modest (around 10Gbps). There was a brief spike around 16:30 UTC, likely a test, that lasted approximately 10 minutes. Then, around 21:30 UTC, the attackers let loose a very large wave.

The graph below is generated from bandwidth samples across a number of the routers that sit in front of servers we use for DDoS scrubbing. The green area represents in-bound requests and the blue line represents out-bound responses. While there is always some attack traffic on our network, it's easy to see when the attack against Spamhaus started and then began to taper off around 02:30 UTC on March 20, 2013. As I'm writing this at 16:15 UTC on March 20, 2013, it appears the attack is picking up again.



In the Spamhaus case, the attacker was sending requests for the DNS zone file for ripe.net to open DNS resolvers. The attacker spoofed the CloudFlare IPs issued for Spamhaus as the source in their DNS requests. The open resolvers responded with DNS zone file, generating collectively approximately 75Gbps of attack traffic. The requests were likely approximately 36 bytes long (e.g. dig ANY ripe.net @X.X.X.X +edns=0 +bufsize=4096, where X.X.X.X is replaced

with the IP address of an open DNS resolver) and the response was approximately 3,000 bytes, translating to a 100x amplification factor.
We recorded over 30,000 unique DNS resolvers involved in the attack. This translates to each open DNS resolver sending an average of 2.5Mbps, which is small enough to fly under the radar of most DNS resolvers. Because the attacker used a DNS amplification, the attacker only needed to control a botnet or cluster of servers to generate 750Mbps -- which is possible with a small sized botnet or a handful of AWS instances. It is worth repeating: open DNS resolvers are the scourge of the Internet and these attacks will become more common and large until service providers take serious efforts to close them.

While large Layer 3 attacks are difficult for an on-premise DDoS solution to mitigate, CloudFlare's network was specifically designed from the beginning to stop these types of attacks. CloudFlare made heavy use of Anycast. That means the same IP address is announced from every one of our 23 worldwide data centers. The network itself load balances requests to the nearest facility. Under normal circumstances, this helps us ensure a visitor is routed to the nearest data center on our network.

When there's an attack, Anycast serves to effectively dilute it by spreading it across our facilities. Since every data center announces the same IP address for any CloudFlare customer, traffic cannot be concentrated in any one location. Instead of the attack being many-to-one, it becomes many-to-many with no single point on the network acting as a bottleneck.

Once diluted, the attack becomes relatively easy to stop at each of our data centers. Because CloudFlare acts as a virtual shield in front of our customers sites, with Layer 3 attacks none of the attack traffic reaches the customer's servers. Traffic to Spamhaus's network dropped to below the levels when the attack started as soon as they signed up for our service. While the majority of the traffic involved in the attack was DNS reflection, the attacker threw in a few other attack methods as well. One was a so-called ACK reflection attack. When a TCP connection is established there is a handshake. The server initiating the TCP session first sends a SYN (for synchronize) request to the receiving server. The receiving server responds with an ACK (for acknowledge). After that handshake, data can be exchanged.

In an ACK reflection, the attacker sends a number of SYN packets to servers with a spoofed source IP address pointing to the intended victim. The servers then respond to the victim's IP with an ACK. Like the DNS reflection attack, this disguises the source of the attack, making it appear to come from legitimate servers. However, unlike the DNS reflection attack, there is no amplification factor: the bandwidth from the ACKs is symmetrical to the bandwidth the attacker has to generate the SYNs. CloudFlare is configured to drop unmatched ACKs, which mitigates these types of attacks.

Whenever CloudFlare see one of these large attacks, network operators will write to us upset that we are attacking their infrastructure with abusive DNS queries or SYN floods. In fact, it is their infrastructure that is being used to reflect an attack at us. By working with and educating network operators, they clean up their network which helps to solve the root cause of these large attacks.

(Reference: https://blog.cloudflare.com/the-ddos-that-knocked-spamhaus-offline-and-ho/)

7) Study the Amazon Route 53 service and answer the following questions

1. What does Route 53 do?
2. Why is it called Route 53?
3. What other Amazon services is it designed to work with (please explain how it happens with one or two examples)?
4. What is the difference between the domain name and hosted zone?
5. Does Route 53 have a default for the Time-to-live (TTL) value?
6. What is the pricing of the service?

Sol: Amazon Route 53 is a highly available and scalable cloud Domain Name System (DNS) web service. It is designed to give developers and businesses an extremely reliable and cost effective way to route end users to Internet. Amazon Route 53 is fully compliant with IPv6 as well.

1. Amazon Route 53 allows you to register new domain name or transfer in existing domain names and supports domain registration for a wide variety if generic-top-level domains (such as .com or .org) and geographic-toplevel domains (such as .be or .us). It helps in creation, updation, and management of public DNS records, let you manage the IP listed for domains names in Internet's DNS phonebook. Route 53 translates specific domain name like www.example.com in to their corresponding IP address like 192.0.2.1. In addition, it sends automated requests over the Internet to your application to verify that it's reachable, avaliable, and functional and offers health check to monitor health and performance of your application as well as web servers and other resources.

2. AWS supposedlt named the service Route 53 because it refers to TCP or UDP port 53 and handles all DNS request through port 53.

3. Amazon Route 53 is designed to work well with other AWS features and offerings. You can use Amazon Route 53 to map domain names to your Amazon EC2 instances, Amazon S3 buckets, Amazon CloudFront distributions, and other AWS resources. By using the AWS Identity and Access Management (IAM) service with Amazon Route 53, you get fine grained control over who can update your DNS data. You can use Amazon Route 53 to map your zone apex (example.com versus www.example.com) to your Elastic Load Balancing instance, Amazon CloudFront distribution, AWS Elastic Beanstalk environment, or Amazon S3 website bucket using a feature called Alias record.

4. A domain is a general DNS concept. Domain names are easily recognizable names for numerically addressed Internet resources. For example, amazon.com is a domain. A hosted zone is an Amazon Route 53 concept. A hosted zone is analogous to a traditional DNS zone file; it represents a collection of records that can be managed together, belonging to a single parent domain name. All resource record sets within a hosted zone must have the hosted zone's domain name as a suffix. For example, the amazon.com hosted zone may contain records named www.amazon.com, and www.aws.amazon.com, but not a record named www.amazon.ca. You can use the Route 53 Management Console or API to create, inspect, modify, and delete hosted zones. You can also use the Management Console or API to register new domain names and transfer in existing domain names into Route 53's management.

5. The time for which a DNS resolver caches a response is set by a value called the time to live (TTL) associated with every record. Amazon Route 53 does not have a default TTL for any record type. You must always specify a TTL for each record so that caching DNS resolvers can cache your DNS records to the length of time specified through the TTL.

6. There is no minimum fee
Hosted Zones $0.50 per hosted zone / month for the first 25 hosted zones $0.10 per hosted zone / month for additional hosted zones

The monthly hosted zone prices listed above are not prorated for partial months. A hosted zone is charged upon set-up and on the first day of each subsequent month. To allow testing, a hosted zone that is deleted within 12 hours of creation is not charged; however, any queries on that zone will be charged at the rates below.

Traffic Flow $50.00 per policy record / month A policy record represents the application of an Amazon Route 53 Traffic Flow policy to a specific DNS name (such as www.example.com) in order to use the traffic policy to manage traffic for that DNS name. The monthly price listed above is prorated for partial months. There is no charge for traffic policies that are not associated with a DNS name via a policy record.

Standard Queries $0.400 per million queries – first 1 Billion queries / month $0.200 per million queries – over 1 Billion queries / month

Latency Based Routing Queries $0.600 per million queries – first 1 Billion queries / month $0.300 per million queries – over 1 Billion queries / month

Geo DNS Queries $0.700 per million queries – first 1 Billion queries / month $0.350 per million queries – over 1 Billion queries / month

The query prices listed above are prorated; for instance, a hosted zone with 100,000 standard queries would be charged $0.040 and a hosted zone with 100,000 Latency Based Routing queries would be charged $0.060.

Queries to Alias records that are mapped to Elastic Load Balancers, Amazon CloudFront distributions, AWS Elastic Beanstalk environments, and Amazon S3 website buckets are free. Alias records can be created for all query types: standard queries, latency-based routing queries, and geo queries. These queries are listed as “Intra-AWS-DNS-Queries”, “Intra-AWS-LBR-Queries”, or “Intra-AWS-Geo-Queries” on the Amazon Route 53 usage report.

(Reference: https://aws.amazon.com/route53/, https://aws.amazon.com/route53/faqs/, https://en.wikipedia.org/wiki/Amazon_Route_53)

8) Take a look at https://www.twistlock.com/2018/11/13/open-source-clouddiscovery-tool/ and learn what the Cloud Discovery service is. Explain how the tool works. What does it do? (Just research your answer and explain how you understand it.)

Incidentally, this is the tool Amazon uses. Does Route 53 provide a similar service? If so, how? What are the differences?

Sol: **What is Cloud Discovery?**

Cloud Discovery is an open source tool that helps infrastructure, operations, and security teams identify all the cloud native platform services, such as container registries, managed Kubernetes platforms, and serverless services used across your cloud providers, accounts, and regions. Cloud Discovery is a powerful tool for audit and security practitioners that want a simple way to discover all the 'unknown unknowns' across environments without having to manually login to multiple provider consoles, click through many pages, and manually export the data.

How it works?

Cloud Discovery connects to cloud providers' native platform APIs to discover services and their metadata and requires only read permissions. Cloud Discovery also has a network discovery option that uses port scanning to sweep IP ranges and discover cloud native infrastructure and apps, such as Docker Registries and Kubernetes API servers, with weak settings or authentication. This capability is useful for discovering 'self-installed' cloud native components not provided as a service by a cloud provider, such as a Docker Registry running on an EC2 instance.

Cloud Discovery is provided as a simple Docker container image that can be run anywhere and works well for both interactive use and automation. Today, Cloud Discovery supports asset identification on AWS, Azure, and Google Cloud Platform but it's designed to be easily pluggable with support for more cloud platforms coming soon.

A quick walkthrough with sample outputs

The tool runs as as a container and requires the following environment variables:
BASIC_AUTH_USERNAME: This variable determines the username to use for basic authentication.
BASIC_AUTH_PASSWORD: This variable determines the password to use for basic authentication. (You can, of course, use Twistlock to secure inject this secret at runtime from secrets managers like Vault, CyberArk, or AWS Secrets Manager!)
TLS_CERT_PATH: This variable determines the path to the TLS certificate inside the container. By default the service generates self-signed certificates for localhost usage.
TLS_CERT_KEY: This variable determines the path to the TLS certificate key inside the container.

Starting the Cloud Discovery container

First, users would run the Cloud Discovery container using the following command:

```
docker run -d --name cloud-discovery --restart=always \ -e BASIC_AUTH_USERNAME=admin -e
BASIC_AUTH_PASSWORD=pass -e PORT=9083 -p 9083:9083 twistlock/cloud-discovery
```

We built Cloud Discovery as service, rather than a locally run tool, to make automation easy. You can deploy Cloud Discovery as just another app on Kubernetes (or whatever platform you use for managing your containers) then have multiple tools and workflows call into it to generate up to date results on demand. For example, if you want to generate a weekly report on new services, you'd simply call the endpoints (as described below) and diff the current results from the previous one.

Amazon Route 53 has same features as Cloud Discovery Service such as Management Console Amazon Route 53 works with the AWS Management Console. This web-based, point-and-click, graphical user interface lets you manage Amazon Route 53 without writing any code at all. AWS Cloud Map automates DNS configuration and simplifies the provision of instances for services such as Amazon Elastic Container Service (Amazon ECS), Fargate, and Kubernetes. You can create a hosted zone with AWS Cloud Map using the AWS SDK or the AWS CLI:

1. Create a DNS namespace (for which a hosted zone is automatically created) to define your service naming scheme.
2. Create your service.
3. Register an instance to your service.

(References: https://aws.amazon.com/premiumsupport/knowledge-center/service-discovery-route53-auto-naming/, https://www.twistlock.com/2018/11/13/open-source-clouddiscovery-tool/)