Cloud Computing (UE18CS352)
Unit 5

Aronya Baksy

April 2021

1 Proxies in the cloud

1.1 Reverse Proxy

- A reverse proxy receives HTTP connection requests from clients and routes the traffic to the application's origin server. Reverse proxies are maintained by the owner of the origin server.
- Reverse proxy servers (implemented in Apache, Nginx, Caddy) can inspect HTTP headers and route requests directed at a single IP address to any one of many internal servers based on the domain name.
- Reverse proxy servers improve security, performance and reliability
- Operation:
  1. Receive connection request from client
  2. Complete the three-way TCP handshake, terminate the orignal connection, connect with the origin server and complete the request.
- Benefits:
  - Application security
  - Load balancing when the origin server is replicated across multiple machines
  - Caching
  - SSL encryption

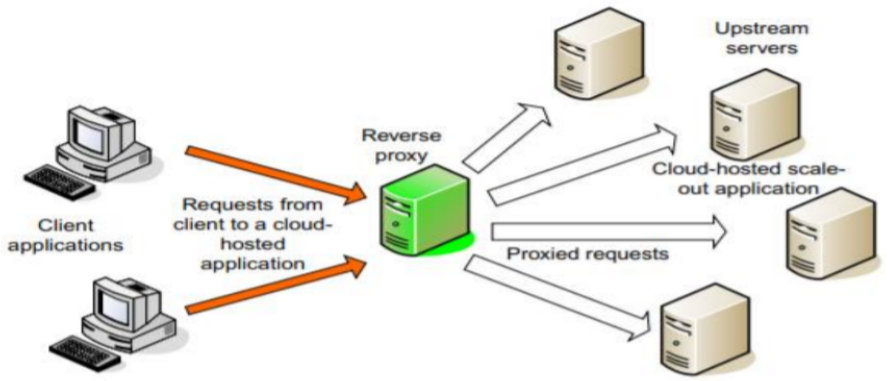

Figure 1: Reverse Proxy Configuration

1.2 Forward Proxy

- Regulates outbound traffic in accordance with certain policies in shared networks. Collects requests from clients, and interacts with servers on behalf of the client.
- Forward proxies are useful in order to :
  - Block access to certain websites for an organization, and monitor organization's online activities.
  - Block malicious traffic from reaching origin servers.
  - Cache external site content and hence reduce response times.

1.3 Nginx

- Nginx is a web server that can also be used as a reverse proxy, load balancer and HTTP cache.
- Load balancing is either done using round-robin scheduling, or the optional hash-based scheduler that chooses an upstream server based on the hash of some value (can be request URL, incoming HTTP headers, or some combination of the same)
- Scaling is done by simply changing the Nginx server configuration i.e. by adding more servers and the corresponding IP addresses in the "upstream" section

2 Scalability

- Ability to increase/decrease IT resources deployed in response to changing workloads and demands.
- Scaling can be done for data storage, compute or networking, and must be done with minimal downtime or service disruption.
- Elasticity refers to the system's ability to allocate or deallocate resources for itself in response to changing workloads
- On the other hand, scalability refers to the ability to use only existing resources to handle increased workloads
- The tradeoff between Elasticity and scalability depends on the app's workloads being predictable or highly variable.

2.1 Benefits of cloud scalability

- **Cost Saving**: pay-as-you-go model, avoid purchasing expensive hardware that soon may become obsolete
- **Disaster Recovery** costs are reduced as need for maintaining secondary data centers is eliminated
- **Convenience**: rapid provisioning of resources, customized to organization needs
- **Flexibility** in handling variable workloads with minimal costs, helps small businesses greatly.

2.2 Scaling Strategies

2.2.1 Vertical Scaling

- Adding more resources (CPU, Memory, Disk, I/O) to an existing server, or replacing an existing server with a more powerful one.
- AWS and Azure support vertical scaling by changing instance types.
- AWS and Azure cloud services have many different instance sizes, so scaling vertically is possible for many types of resources (EC2 instances, RDS databases)

2.2.2 Horizontal Scaling

- Adding more instances of the same existing configuration and splitting workloads between the new increased number of instances
- Increase number of instances instead of changing instance type

Think of it like this, vertical scaling is adding more floors to a single house, whereas horizontal scaling is building 2 more houses of the same size as the existing one.

2.3 Scaling through Reverse Proxies

- The reverse proxy might even be configured as a load balancer.
- To the outside world there is just a single server, but the load balancer takes each request and forwards it on to an application server on the private network.
- Load balancer decides which server should receive the request based on some scheduling algorithm.

3 Hybrid Cloud and Cloud Bursting

- Combination of private and public clouds enabling expansion of local infrastructure to commercial infrastructure on a need basis
- Organizations can leverage existing infrastructure and supplement it with cloud resources as per demand
- **Cloud bursting** is an configuration between public and private clouds that allows for uninterrupted service by sending excess traffic beyound the capability of the private cloud, to the public cloud.
- Hybrid cloud enables Cloud Bursting of the private cloud by allowing the addition of extra capacity to a private infrastructure by borrowing from a public cloud
- Benefits of cloud bursting
  - Flexible and cost-effective solution to manage sudden workloads seamlessely.
  - Simple to manage scaling up/down of resources in public cloud
  - Cost savings on internal hardware procurement for an organization. Internal Compute resources Freed up for better usage in other areas.
  - Improved customer experience and customer retention levels due to uninterrupted access to the application.

4 Multi-Tenancy

- An architecture model wherein a single instance of an application or a hardware serves multiple clients.
- Three types of multi-tenancy models:
  - Shared Machine: each client has their own DB process and tables on a single shared machine
  - Shared-Process: Each client has their own tables, but only one database process executes queries for all clients
  - Shared-Table: clients share database tables and process.

4.1 Requirements of a Multi-Tenant System

- Fine-grained resource sharing: leads to greater scalability, but also necessitates better access control and security.
- Security and isolation between tenants
- Customization of tables

4.2 Types of Multi-tenant Architectures

4.2.1 Single multi-tenant database

- A single app instance, and a single database instance.
- Highly scalable. As more tenants are added, the database is scaled up by adding more storage.
- Low cost due to shared resources, but high operational complexity during design and setup

4.2.2 One Database per Tenant

- Single app instance, one DB instance per tenant.
- Higher cost and less scalable than single multi-tenant architecture, but operational complexity is low.
- Scalability may be achieved by adding more DB nodes.

4.2.3 Single-Tenant App with Single-Tenant DB

- The entire app is installed separately for each tenant. Each tenant has their own app instance and their own DB instance.
- Highest level of data isolation, but high cost due to extra hardware needed to support.

4.3 Levels of Multi-Tenancy

4.3.1 Ad-Hoc or Custom Instances

- Each tenant has their own custom version of software.
- Found in current enterprise data centers.
- Management is difficult as each customer needs specialized management support.

4.3.2 Configurable Instances

- All tenants share same version of program, but configuration is possible to an extent.
- Significant management savings as only one copy of the software is to be maintained.

4.3.3 Configurable, Multi-Tenant instances

- Only one instance of the running program is shared by all customers.
- Leads to additional efficiency in resource usage as well as management

4.3.4 Scalable, configurable multi-tenant instances

- Instances can scale up or down depending on the number of customers, and demand of each customer.
- Peformance bottlenecks and capacity limitations from other levels are eliminated here to an extent

4.4 Challenges of Multi-Tenancy

4.4.1 Authentication

- Secure sharing of resources is enforced using authentication
- In a *centralized authentication system*, auth takes place using a centralized user database. The cloud admin gives the tenants the right to manage their own accounts on this database.
- In a *decentralized authentication system*, each tenant maintains their own user database, and the tenant deploys a federation service that interfaces between the authentication services of tenant and cloud.

4.4.2 Implementing Resource Sharing

- Access control is provided using **roles** and **business rules**.
- A *role* is associated with a set of permissions specific to it. The ability to set permissions for roles is also attached to a certain small set of roles.
- A *business rule* is a policy that provides fine-grained access control, based on the context of the running application (e.g. in a banking app, limit the amt of money withdrawn in a single transaction, or limit the time during which transaction can take place)
- Business rules are implemented using policy engines like Drools Guvnor and Drools Expert
- Two types of access control:
  - **Access Control List**: Each object associated with a set of permissions for each role
  - **Capability-based Access Control**: If a user holds a reference or capability (called a **key**) to an object, they have access to the object.

4.4.3 Sharing Storage Resources

- In a **shared table** approach, all tenant's data is stored in a single table. A *metadata table* stores info about tenants.
- Shared table is more space efficient but requires multiple select statements for a query (to join between metadata table and actual data table)
- In a **dedicated table** approach, each tenant has their own table. Access to other tenant's tables is restricted.

5 Cloud Security

- A set of control-based safeguards and technologies that protect cloud resources from online theft, leakage or data loss.
- Cloud security is partitioned into the physical and virtual domains. Basic objectives of cloud security are confidentiality, integrity and availability

5.1 Physical Security

- Physical security involves protection against physical threats like intruders, natural disasters and human error (e.g. forgot to turn on the AC)
- Multi-layered physical security system involves:
  1. Central monitoring and control center with dedicated staff
  2. Monitoring for each type of physical threat
  3. Training of staff in response to threat situations
  4. Manual or automated backup systems to mitigate damage caused
  5. Secure access to facility

5.2 Virtual Security: Best practices

5.2.1 Cloud Time Service

- Synchronize all nodes in the data center to the same clock.
- Synchronization is needed for correct ordering of operations, as well as analysis of system logs across geographically distributed locations
- Network Time Protocol (NTP) is used for this. Encryption is used to avoid fake reference sources.

5.2.2 Identity and Access Management

- IM must be scalable, federated, allow single identity and single sign-in and must satisfy legal and policy requirements.
- Access Management allows access to cloud facilities only to authorized users.
- In addition, it controls the access of cloud management personnel, implements 2-factor authentication, disallows shared accounts and white lists IP addresses to allow remote access.

5.2.3 Break-glass protocols

- In case of emergencies, bypass normal security controls and allow an alarm to be triggered
- Such a protocol must ensure that it can be executed only in emergencies under controlled situations and that the alarm is triggered properly.

5.2.4 Key Management

- Secure facilities for the generation, assignment, revocation, and archiving of keys.
- Also generate procedures for recovering from compromised keys

5.2.5 Auditing

- Capture all security-related events, together with data needed to analyze the event
- This data includes time, system on which the event occurred, and userid that initiated the event.
- The audit log is centralized and secure
- It must be possible to create a sanitized or stripped-down version to share with cloud customers for further analysis.

5.2.6 Security Monitoring and Testing

- Monitoring involves a system-wide anomaly and intrusion detection system installed on network and host nodes.
- At times, cloud users can create their own anomaly and intrusion detection systems.
- All software (releases or patches) are tested in a test bed environment before deployment to production environment.
- Testing happens on a continuous ongoing basis that identifies vulnerabilities in the cloud system.

5.3 Risk Management in Cloud

- Risks in data security and resource outages can be crippling for dependent organizations.
- Risk management is the process of identifying, evaluating, monitoring and controlling risks in a business environment.
- Risk management is domain dependent, and must tradeoff between risk impact and cost of risk mitigation measure.
- A **security control** is a safeguard that detects, responds or prevents a security risk. There are three broad categories of security controls: technical, operational and Management, with each further divided into 18 families.
- Security breaches are classified as low-impact, medium-impact or high-impact based on the requirement for security control.
- **Low Impact Systems** are those where a security breach causes *limited degradation in capability*, but the system can still perform its primary functions.

- **Medium Impact Systems** are those where the system is still capable of performing its primary functions but there is a *significant degradation in the capabilities.*
- **High-Impact Systems** are those where a security breach causes *inability to perform primary functions.*

5.3.1 Risk Management Process

- Categorize information resources on the basis of criticality (impact in case of failure) and sensitivity (confidentiality)
- Select security controls appropriate to the levels of criticality and sensitivity chosen
- Evaluate the chosen security controls. Upgrade them if found insufficient against anticipated threats.
- Implement chosen security controls. These may be administrative, technical or physical.
- Monitor effectiveness of deployed security controls
- Periodic review of all security controls must take place to protect against new threats, and to account for operational changes in system design etc.

5.4 Security Design Patterns

5.4.1 Defense in Depth

- Layered defenses protect sensitive resources.
- e.g.: Remote access to a cloud allowed only through VPN. Access could be allowed only from certain whitelisted IP addresses. Admins may be needed to further provide a OTP for access.

5.4.2 Honeypot

- Honeypots are systems that disguise themselves as valuable targets, while being monitored by security personnel.
- While an attacker attempts to control the honeypot, the sysadmins monitoring the honeypot can trap and stop the attack.
- Honeypot VMs can be deployed by the cloud provider or the cloud customers.

5.4.3 Sandboxing

- Execution of software inside a controlled environment within an operating system.
- Within the sandbox, the software has access only to the bare minimum resources it needs to function properly. Hence any attacker gaining control of the software does not have unrestricted access to the entire system.
- Sandboxes also provide defense in depth as any attacker is also needed to overcome the sandbox in order to gain unrestricted access.

5.5 Network Design Patterns for Security

5.5.1 VM Isolation

- Encryption of traffic between VMs, or tightened network controls on VMs (using ACLs, restricting port numbers)

5.5.2 Subnet Isolation

- Separate subnets for admin traffic, user traffic and storage network traffic.
- Physically separate networks are preferred as virtual LANs (VLAN) that are not physically separate are hard to configure correctly.
- Routing between the networks is handled by firewalls.

5.5.3 Common Management DB

- a database that contains information regarding the components of an IT system (inventory of components, present config and status)
- Simplifies implementation and management of IT services, allows all admins to have a consistent view of the IT system.

5.6 Example of PaaS Security

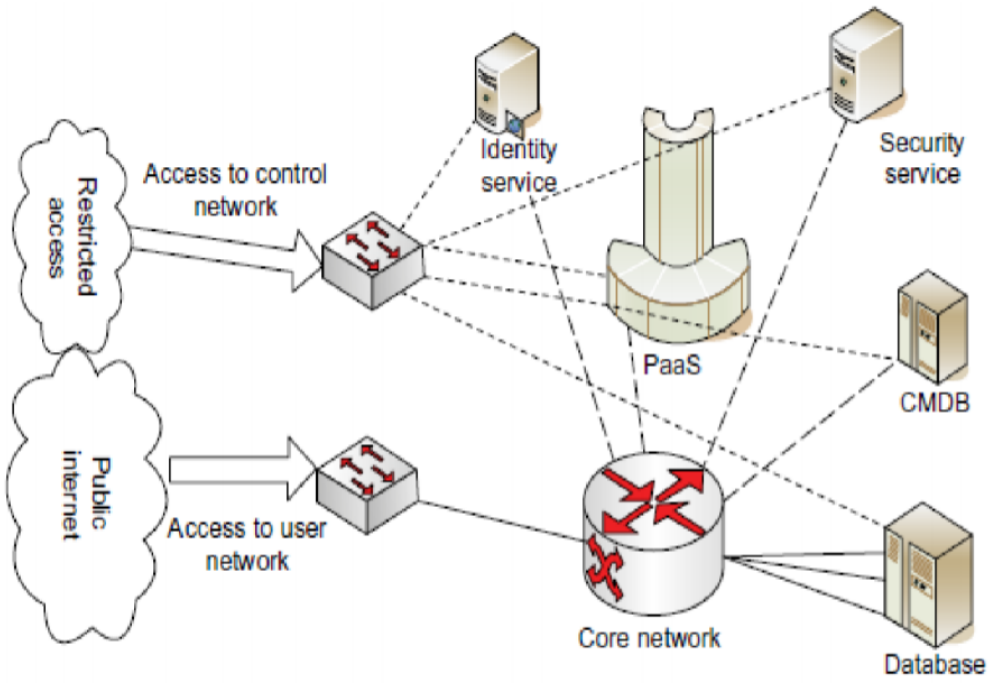

Figure 2: Security Architecture for PaaS System

5.6.1 External Network Access

- Distinct interfaces for distinct physical networks, one for admins and one for cloud users.
- Access to control network is limited to whitelisted IP addresses only.
- Multi-factor authentication can be made mandatory for increasing secure access to administrative functions.
- The access to the public network is via two switches, to increase availability via redundancy.

5.6.2 Internal Network Access

- Separate physical networks for admin control functions and one for cloud user functions. Protects control network from unauthorized access.
- The DBMS is connected to the public network via an aggregated set of links to provide increased bandwidth and availability.
- PaaS service is accessible from public and private networks. But the security server need not be accessible from the public network.

5.6.3 Database Server Security

- The identity server handles access management to the database.
- Database is further secured by restricting the allowed ports on which internet traffic is allowed.
- Additional security is implemented by checking the validity of the ODBC connection from the client to the database

5.6.4 Security Service

- The diagram also includes a security server to perform security services such as
  - Auditing of security
  - Monitoring for security threats
  - Hosting a security operations center
  - Security scanning of the cloud infrastructure

5.7 Standards for Security Architecture

5.7.1 SSE-CMM

- System Security Engineering Capability Maturity Model, adaptation of CMM for software projects by CMU
- Defines 5 capability levels for an organization
- Allows organizations to plan and implement processes for self improvement

5.7.2 ISO/IEC 27001-27006

- Set of related standards under the ISO/IEC 27000 family that provides an Info. Security Management System.
- Specifies requirements to be satisfied by all organizations, and processes for evaluating security risks
- Not specific to cloud

5.7.3 ENISA

- European Network and Info. Security Agency provides a **Cloud Computing Information Assurance Framework**.
- The framework is a set of assurance criteria designed to assess the risk of adopting cloud services, compare different Cloud Provider offers, obtain assurance from the selected cloud providers, and reduce the assurance burden on cloud providers

5.7.4 ITIL Security Management

- ITIL is a comprehensive set of standards used for ITSM, based on ISO/IEC 27002.
- Shallow learning curve due to the fact that ITIL is already adopted in many data centers.

5.7.5 COBIT

- Control OBjectives for Information related Technologies, developed by the ISACA.
- A set of best practices for linking business and IT goals, with metrics and maturity models.
- Broader scope than ISO/IEC 27000

5.7.6 US NIST

- US National Institute for Standards and Technology releases many whitepapers in the Security Management and Assurance working group.
- Targeted at US Federal Agencies (CIA, FBI etc.) , but apply to many organizations as well.

5.8 Legal and Regulatory Issues with the cloud

- Local, national and international laws apply due to distributed nature of the cloud, as well as the presence of a third party (i.e. the cloud provider)
- Such laws must specify who is responsible for security and accuracy of the data stored on cloud.
- Issues to consider when framing laws:
  - Cover all risks arising from a third party's presence
  - Need to ensure data security
  - Obligations of the cloud provider during any litigation

5.9 Legal Issues

5.9.1 Due Diligence

- Client must define scope of service provided, as well as regulations and compliance standards to be followed
- Consider any risks arising from stability and reliability of the cloud provider, as well as the criticality of the business function outsourced to the cloud.

5.9.2 Contract Negotiation

- Cloud services may have one-click standard agreements that are not customizable. Such agreements are acceptable for low-risk scenarios.
- Cloud service providers can avoid negotiating custom agreements with each customer through external accreditations

5.9.3 Implementation

- Enterprise must ensure that the safeguards laid out in the contract are actually being followed
- It is also important to continuously re-evaluate the system periodically to check for changed circumstances (increased data sensitivity, revoked external certifications)

5.9.4 Contract Termination

- Identify alternate service provider, ensure timely and secure transfer of services
- Also ensure that sensitive data if any, is completely deleted from the original provider's systems.

5.9.5 Data Privacy and Secondary use of Data

- Use collected data only for intended purpose, and such data cannot be sold to third parties
- Privacy laws often state that individuals can access their own data and modify or delete it
- Enterprises must ensure that cloud service providers do not use the data for data mining or other secondary usage.

5.9.6 Data Location

- Data handling laws differ between countries, hence transferring data between countries is a challenging process.
- Allow for the location of data to be known in advance so that such scenarios can be planned (e.g. AWS allows selection of regions for all services)
- The enterprise must obey the most stringent of all the laws that apply across the countries where data is stored.

5.9.7 Business Continuity Planning

- BCP is used to implement actions to keep a business running in the face of natural disasters that affect infrastructure, including those maintained on a third-party cloud.
- BCP typically involves identifying the possible catastrophes, carrying out Business Impact Analysis, and using the results of the analysis to formulate a recovery plan
- Disaster Recovery planning (DRP) is a part of BCP used for recovery of IT operations.
- BCP and DRP are made before deploying apps to cloud, and implemented during the deployment. Some cloud providers provide features (e.g. multi-locations) that help in BCP and DRP

5.9.8 Security Breaches

- In case of a breach, cloud provider's disclosure policy is important.
- Disclosure policy defines how quickly a customer is notified of a breach, so that corrective action can be taken.
- To avoid ambiguity, the service agreement should specify the actions to be taken during a breach

5.9.9 Litigations

- During a litigation against an enterprise or a cloud provider, the provider must be able to make available any data that is needed for this litigation.
- This is important as enterprises (not cloud providers) are responsible for responding to such requests.
- In case a cloud provider is directly requested to provide data, then the affected business must be contacted and must be given the opportunity to oppose the request.

6 Cloud Authentication: Keystone

- Keystone is an OpenStack service that provides:
  - API client authentication
  - Service discovery
  - Ditributed multi-tenant authorization

  using OpenStack's Identity API
- The fundamental purpose of Keystone is to be a registry of projects and decide on access to projects.

6.1 Terminologies

6.1.1 Project

- An abstraction used to group resources (servers, machine images etc.)
- Users or user groups are given access to projects using role assignments.
- The specific role assigned outlines the type of access and capabilites that a user/user group is entitled to.

6.1.2 Domain

- AN abstraction that isolates the visibility of a set of projects and users (or user groups) to a single organization
- Domains enable splitting cloud resources into silos that can be used by each organization.
- Domains represents logical divisions within an enterprise, or maybe entirely different enterprises

6.1.3 Users and User Groups

- Also known as actors, they are the ones who utilize the cloud resources.
- User groups are groups of users that have some shared responsibility.

6.1.4 Relationship between the three

Domains are a collection of Users, Groups, and Projects. Roles are globally unique. Users may have membership in many Groups.

6.1.5 Roles

- "Assigned to" an user and "assigned on" a project.
- Convey a sense of authority, a particular responsibility to be fulfilled by an actor.
- A role assignment is a triple of actor, target (may be a project or a domain), and a role.
- Role assignments can be granted, revoked and inherited between users/projects.

6.1.6 Token

- Each API call authenticated by Keystone requires the passing of a token.
- Tokens are generated by Keystone upon successful authentication of an user against the service.
- A token has both an unique ID (unique per cloud) and a payload (data about the user)

6.1.7 Service Catalog

- List of endpoints and URLs for different services on a cloud.
- Used mainly for service discovery and access (such as creating VMs, storage allocation etc.)
- Each endpoint is broken down into a public URL, an internal URL and an admin URL (all may be the same or not)

6.2 Identity in Keystone

6.2.1 SQL

- Identity of actors (name, password, metadata) and groups stored on an SQL database (MySQL, PostgreSQL, DB2)
- Keystone in this case serves as the identity provider
- Pros:
  - Easy setup
  - Manage users and groups via OpenStack APIs
- Cons:
  - Keystone should not be identity provider as well as authenticator
  - Weak password support: no password rotation or recovery
  - Does not integrate with existing enterprise LDAP servers

6.2.2 LDAP

- Keystone can retrieve and store actors (Users and Groups) in Lightweight Directory Access Protocol (LDAP).
- LDAP should be restricted to only read operations (searching) and authentication (bind).
- Keystone needs a minimal amount of privilege to use the LDAP (read access to attrs defined in the configuration, as well as an anonymous access)
- Pros:
  - No need to maintain copies of user accounts
  - Keystone no longer acts as identity provider
- Cons:
  - Service accounts need to be stored somewhere (may not be desirable to have them on LDAP server)
  - Keystone is still seeing user passwords in the request messages. Ideally Keystone should never see user passwords.

6.2.3 Multiple backends

- Allow one identity source per Keystone domain.
- Allows service accounts and employee accounts to be separated, and allows use of multiple LDAPs for flexibility in organization of departments.
- Pros:
  - Support multiple LDAPs for various user accounts, SQL for service accounts
  - Leverage existing LDAP identity
- Cons:
  - Complex set up
  - User authentication must be domain-scoped

6.2.4 Identity Providers

- An identity provider is a service that abstracts the identity service backed and translates user information into some standard federated identity protocol.
- Keystone uses Apache modules for consuming authentication info from multiple Identity Providers.
- Such users never stored in Keystone, not permanent, users will have their attributes mapped into group-based role assignments
- From a Keystone perspective, an identity provider is a source for identities; it may refer to software that is backed by various backends or Social Logins
- Pros:
  - Leverage existing infra & software for user authentication
  - Separation between Keystone service and user info
  - Keystone never sees any user passwords
  - Type of authentication (certificate-based, 2-factor) is abstracted away from keystone
- Con: most complex setup

6.3 Authentication in Keystone

6.3.1 Password

- User or service provides a password for authentication
- The payload of the request contains information needed to find where the user exists, authenticate the user, and optionally, retrieve a service catalog based on the user's permissions on a scope
- The user section identifies the user (either on a domain, or using a globally unique user ID),
- The scope section identifies the project being worked on, and hence is used to retrieve the service catalog. Must contain information to identify a project and the owning domain.

6.3.2 Token

- A user may also request a new token by providing a current token.
- The payload contains the current token ID.
- This allows refreshing a token that will soon expire, or changing a token type from unscoped to scoped.

6.3.3 Access Management

- Keystone manages access to APIs using **role-based access control**.
- Consists of policies stored in JSON form at each API endpoint.
- Rules in JSON form consists of target:rule pairs.
- At the top of the file, targets are established that can be used for evaluation of other targets.
- Here the meaning of admin, owner and other roles are defined.

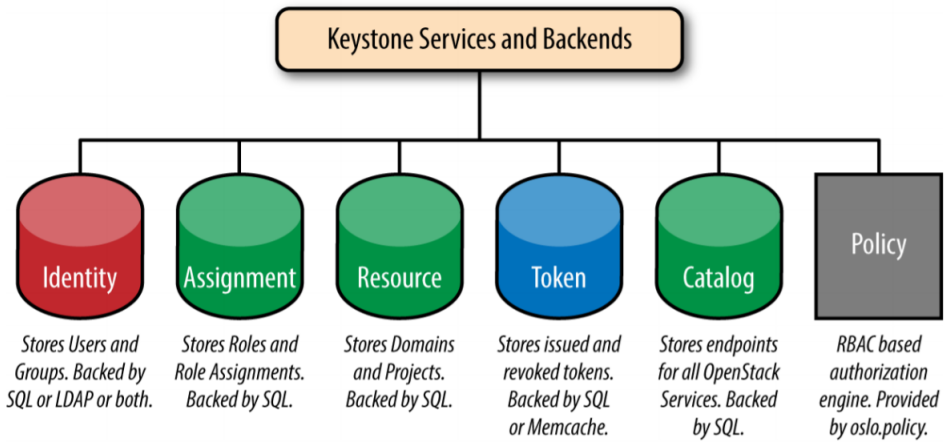

Figure 3: Keystone Services and Backends

7 Cloud Security Threats: Denial of Service

- A type of cyber attack in which a malicious actors aim to render a device unavailable to its intended users by interrupting the device's normal functioning.
- DoS attacks typically function by overwhelming or flooding a targeted machine with requests until normal traffic is unable to be processed, resulting in denial-of-service to addition users.
- The focus of a DoS attack is to overpower the capacity of a targeted machine, resulting in denial-of-service to additional requests.
- Types of DoS attacks
  - **Buffer Overflow**: force a machine to consume resources until capacity runs out. Leads to slow response time, system crashes and hence DoS
  - **Flood Attacks**: Overpower the server by targeting a server with a massive number of packets. For this attack the attacker must have greater bandwidth than the target.

7.1 DDoS Attacks

- **Distributed DoS (aka DDoS) attacks** utilize multiple computers as sources of attack traffic.
- DDoS attacks are carried out with networks of internet-connected machines that have been infected with a malware that allows an attacker to control them remotely.
- Such a network of machines is called a botnet (with individual machines being called bots or zombies). Each bot is a legitimate machine on the internet hence it is difficult to separate attack traffic from actual traffic.
- The botnet floods the victim with requests, overwhelming the capacity and causing denial of service.

7.2 EDoS: Economic Denial of Sustainability

- EDoS targets the vulnerabilities of cloud computing's utility pricing model.
- EDoS attackers steadily send illegitimate traffic to gradually consume cloud resources such as VMs, network devices, security devices and DBs so that it can trigger auto scaling features of cloud
- Consequently, due to the additional resource usage, the target consumer is billed for additional charges, causing financial problems.
- The other side effect of this attack is the persistent degradation of services faced by benign cloud users.

DDoS Attack	EDoS Attack
Degrade/block cloud services	Make cloud resources economically infeasible
Short attack period	Long attack period
Attacks occur above EDoS region	Attacks occur between normal data traffic zone and DDoS attack zone

7.3 Intrusion Detection Systems (IDS)

- Signature matching IDS and anomaly detection can be implemented on VMs that are dedicated to building IDS'.
- Network anomaly detection reveals abnormal traffic patterns, such as unauthorized episodes of TCP connection sequences, against normal traffic patterns.

7.3.1 Defense against DDoS Attack

- Use **successive attack transit** routers in the network along the tree.
- This mechanism is based on change-point detection across each router.
- Based on the anomaly pattern detected in covered network domains, the scheme detects a DDoS attack before the victim is overwhelmed

7.3.2 Data Integrity and Privacy Protection

- Special APIs for authentication, e-mail communication
- Fine-grained access control to deter hackers.
- Personal firewalls at user ends to keep shared data sets from Java, JavaScript, and ActiveX applets
- A privacy policy consistent with the cloud service provider's policy, to protect against identity theft, spyware, and web bugs
- VPN channels between resource sites to secure transmission of critical data objects

7.3.3 Data Colouring

- Data colouring is a watermarking technique that secures data. Each data object is labelled with an unique colour.
- User identification is also coloured to correspond with the data coloured.
- This color matching process can be applied to implement different trust management events.
- Cloud storage provides a process for the generation, embedding, and extraction of the watermarks in colored objects
- Data coloring takes a minimal number of calculations to color or decolor the data objects (compared to encryption/decryption)

7.3.4 Data Lock-In

- Data lock-in is caused by inability to move data from one cloud platform to another to do some other computation
- Causes for data lock-in are *lack of interoperability* (the lack of standard APIs for access) and *lack of application compatibility* (applicaitons are not standard across all clouds)
- Standardized cloud APIs can be built, but this requires providers to build infrastructure that adhere to OVF aa platform-independent, efficient, extensible, and open format for VMs)
- This will enable efficient, secure software distribution, facilitating the mobility of VMs.