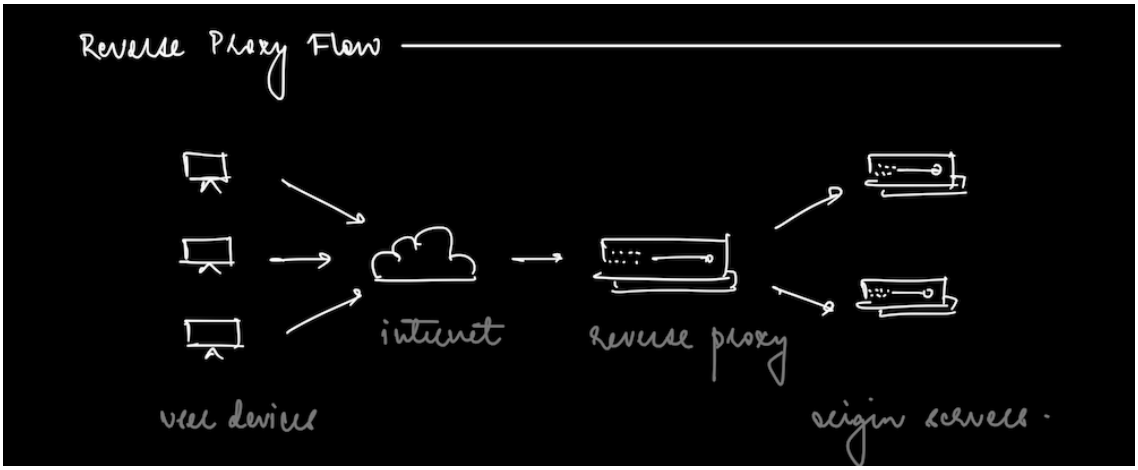
Cloud Computing V

UNIT V - SECURITY AND RISK MANAGEMENT

▼ **Proxies**

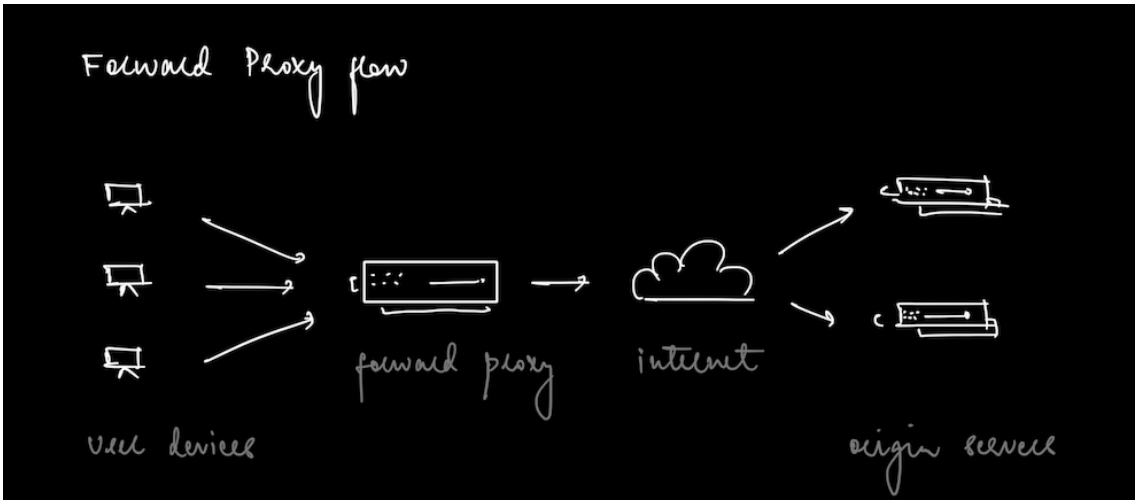
▼ Reverse proxies



Server that sits in front of web servers and forwards client requests to those servers.

- Improved online security - play a key role in building a zero trust architecture for organizations, secures sensitive business data and systems
- Safeguard backend servers from DDoS attacks
- Increased scalability and flexibility - load balancing
- Web acceleration
- Identity branding
- Caching commonly requested data

▼ Forward proxies



Sits in front of a group of client servers and intercepts requests to communicate with other web servers.

- Avoid state or institutional browsing restrictions
- Block access to certain content
- Protect online identities

Forward Proxy	**Reverse Proxy**
Forward proxy connection initiates from inside secured zone to outside unsecured global network	Reverse proxy connection comes from outside global network and destined to inside secured network
Not used for application delivery	Used for application delivery
Good for content filtering, natting, email security	Good for load balancing, TCP multiplexing, Content switching, authentication and application firewall
Restricts internal users from accessing filtered sites	Restrict outside users to have direct access to private networks

▼ **Nginx**

Open source web server that provides capabilities like reverse proxying, load balancing, caching and media streaming.

→ proxy server for email IMAP, POP3, SMTP

→ reverse proxy and load balancers for HTTP/2, TCP, UDP protocols

→ Nginx uses event driven architecture and deals with requests asynchronously.

It was designed to use a non blocking event driven connection algorithm

Benefits of Nginx:

1. Load balancing - provides better SLA for application
2. Security
3. Caching
4. Logging - centralized logging for backend server and provides a single place to audit and log for troubleshooting issues.
5. TLS/SSL support - Secure communication between client server
6. Protocol support - Supports both ipv4 and ipv6 along w http, https, http/1.1, http/2

▼ Cloud scalability

→ Cloud scalability refers to the ability to increase of decrease IT resources as needed to meet changing demand.

→ Cloud elasticity refers to a system's ability to grow or shrink dynamically in response to changing demands

Benefits of cloud scalability:

1. Cost savings
2. Disaster recovery

3. Convenience
4. Flexibility and speed

→ **Vertical scaling** refers to adding more resources to the server, making the server more powerful. (larger instance size)

→ **Horizontal scaling** refers to provisioning additional servers and splitting workloads. (more instances provisioned)

▼ Hybrid cloud and cloud bursting

→ Primary benefit of a hybrid cloud is AGILITY

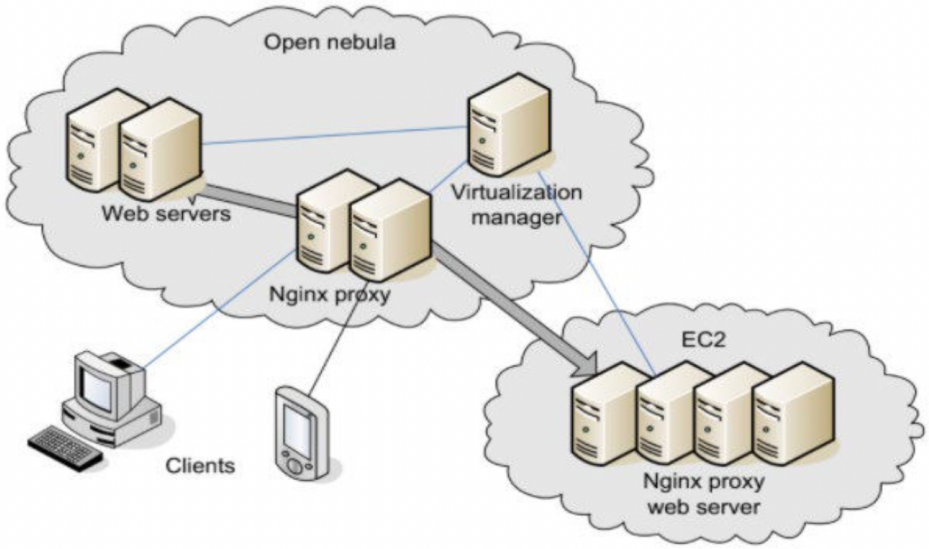
1. Dynamic or frequently changing workloads
2. Separating critical workloads from less sensitive workloads
3. Big data processing
4. Moving to the cloud incrementally
5. Temporary processing capacity needs
6. Flexibility for the future
7. Best of both worlds

Cloud Bursting

If any organization uses a private cloud consumes 100 percent of the available resource then the overflow traffic is directed to the public cloud to avoid any interruption of the service.

1. Software development - DevOps and CI-CD
2. Marketing campaigns
3. Big data modelling and queries
4. Seasonal businesses

OpenNebula with reverse proxy for cloud bursting



▼ **Multitenancy**

Better use of resources and lower costs

Requirements -

→ Fine grain resource sharing

→ Security and isolation between customers

→ Customization of tables

▼ Architecture types

1. Single multitenant database
2. One database per client
3. Standalone single tenant app with single tenant database

▼ Multitenancy levels

1. **AdHoc/Custom instances** - Each customer has their own custom version of the software

2. **Configurable instances** - All customers share the same version of the program and customization is possible based on configuration. Only one copy of the software needs to be maintained.
3. **Configurable multitenant instances** - Cloud systems have only one instance of the program running which is shared among all the customers.
4. **Scalable, configurable, multitenant efficient instances** - Software is hosted on a cluster, allowing the capacity to scale limitlessly.

▼ Resource sharing

Access control - **roles**(permissions) and **business rules**(policies that provide fine grained access control)

Access Control Models -

1. Access control lists
2. Capacity based access control (if a user has reference to the object, that guarantees access to the object)

Resource sharing - Storage and Servers

1. Sharing storage resources (dedicated table and shared table)
2. Sharing compute resources
3. Customization
    - Preallocated columns
    - Name value paris
    - XML method

▼ **Failure detection and application recovery**

→ Heartbeats and probes (lightweight service request)

1. **Failure monitoring**

- Push model (im alive)

- Pull model (are you alive? yeap)
- Dual scheme (are you alive? yeap,,,,im alive)

2. **Redirection**

- Checkpoint/restart

▼ Cloud security requirements

1. Physical security - Secure against physical threats
2. Virtual security
    - *Cloud Time Service* - All systems in the datacenter are synchronized to the same clock. Use NTP (Network time protocol) and encrypt the protocol message.
    - *Identity management* - Establishes a single identity and single sign on across multiple systems. Foundation for achieving confidentiality, integrity and availability.
    - *Access management* - Allow access to cloud facilities only to authorized users.
    - *Break glass procedures* - Allow procedures that bypass normal security controls in emergency situations.
    - *Key management* - Encryption, retrieval and recovery of keys
    - *Auditing* - Audits should capture all security related events, centrally maintained and secure
    - *Security monitoring*
    - *Security testing*

▼ Risk management process

Security control - low (limited degradation), moderate (significant degradation), or high impact(greatest requirements in terms of security functions)

Risk management process:

→ Information resource categorization - Criticality and Sensitivity

→ Select security controls

→ Risk assessment

→ Implement security controls

→ Operational monitoring

→ Periodic review

▼ **Security design patterns**

▼ Defence in depth

Defences should be layered. Can be done using VPN, white listing IP addresses, otps

▼ Honeypots

Decoy computer system that appears attractive to the attacker

▼ Sandboxes

Software is executed in a restricted environment inside the operating system

▼ Network patterns

- VM Isolation - Encryption of traffic and tightened security controls
- Subnet Isolation - Physically separate traffic for administrative network traffic, customer elements
- Common management database - DB that contains informations regarding the components of an IT system.

▼ **Security architecture standards**

- SSE-CMM
  System Security Engineering Capability Maturity Model. Defines 5 capability levels for any organization to assess themselves and put in place processes to improve their levels.

- ISO/IEC
  Information security management system. Specifies a set of requirements that organizations must satisfy.
- ENISA
  European Networks and Intelligence Security Agency. Set of assurance criteria designed to assess the risk of adopting cloud services, compare cloud providers.
- ITIL security management
  Information Technology Infrastructure library. Comprehensive set of standards for IT service management
- COBIT
  Control objectives for IT
- NIST
  National institute of Standards and Technology

▼ **Legal issues**

1. Third party/Contractual issues
   - Due diligence - Specify regulations and compliance standards
   - Contract negotiation
   - Implementation
   - Termination
2. Data handling
   - Data privacy
   - Data location
   - Secondary use of data
   - Business continuity planning and disaster recovery
   - Security breaches
3. Litigation related issues

▼ **Keystone**

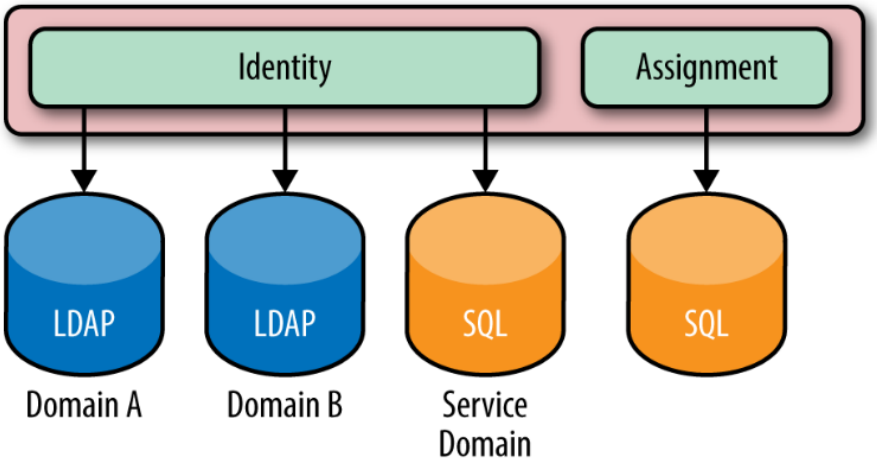
Keystone is an OpenStack service that provides API client authentication, service discovery and distributed multitenant authorization by implementing openstack’s identity api.

▼ Keystone concepts

1. Project - an abstraction to group and isolate resources
2. Domain - isolate the visibility set of projects and users to a specific organization. Serves as a logical division between different portions of enterprises or separate enterprises.
3. Users and User groups - Entities that are given access to resources
4. Roles - used to convey a sense of authorization
5. Assignment - (actor, target, role) triple
6. Target - Projects and Domains
7. Tokens - ID and payload. In order for a user to prove who they are, they pass an OpenStack token into the API call.
8. Catalogue - Service catalogue containing URLs and endpoints of different cloud services.

▼ Identities

- SQL
- LDAP (Lightweight Directory Access Protocol) - retrieve and store actors
- Multiple backends



Authentication - Password, scope, payload, user section, token, RBAC (role based access control)

▼ **Cloud security defence strategies**

1. Basic cloud security
2. Security challenges in VMs
3. Cloud defence methods
4. Defence with virtualization
5. Privacy and copyright protection

▼ **Attacks on cloud**

- **Denial of Service (DoS)**
  Oversaturate the capacity of the targeted machine.
  → Buffer overflow attacks (memory buffer overflow)
  → Flood attacks (attacker has more bandwidth than target)
- **Distributed denial of service (DDoS)**
  Overwhelm using a flood of internet traffic
- **Economic Denial of Sustainability (EDoS)**
  Increase in utilization of computing resource of destination user and results in

huge costs. Concentrates on maximising the financial costs of cloud based consumers.