Openstack Interview Questions

**1.Describe OpenStack.**

Most multinational organizations define OpenStack as the future of Cloud Computing. The Internet and large volumes of data together have instigated the purpose of cloud computing, and OpenStack is one such platform to create and handle massive groups of virtual machines through a Graphical User Interface. It is a set of efficient software tools to manage private and public cloud computing platforms.  
Openstack is free, open-source software and works similar to Linux.

**2. Explain the benefits of using OpenStack.**

• Openstack is useful in developing any software-as-a-service (SAAS) applications, for new developments or to improve existing solutions.  
• Can serve as a strong foundation to deliver self-service storage to IT users.  
• Can deliver on-demand objective or block storage with higher scalability and easy-to-handle storage at lower costs.  
• Most enterprises can save bigger on licensing fees by switching virtual machines running on VMware to OpenStack.

**3.What are the key components of OpenStack?**

• Horizon: the only GUI in OpenStack; the first component administrators see and get an idea of the current operations in the cloud.  
• Nova: chief computing engine to handle multiple virtual machines and computing tasks  
• Swift: reliable and robust storage system for files and objects helping developers to refer to a unique identifier and Openstack decides where to store the info.  
• Cinder: similar to traditional computer storage system, it is a block storage system in OpenStack for accessing files at faster speed.  
• Neutron: ensures efficient connectivity between components during deployment.  
• Keystone: a central identity list of all OpenStack cloud users and provides various mapping techniques to access methods against Keystone.  
• Glance: image service provider where images are the virtual copies of hard disks. Allows using the images as templates during deployment of new instances.  
• Ceilometer: component providing billings services and other telemetry services to cloud users. Maintains an account of component system usage by each user.  
• Heat (Orchestration Engine): Allows developers to orchestrate/illustrate and store the cloud application requirements and resources needed in a file, thereby maintaining the cloud infrastructure.

**4.What storage types are allowed by OpenStack Compute?**

OpenStack supports two types of storage:  
Persistent Storage: Persistent and independent of any particular instance, created by users. This further includes three storages:  
• Object storage: to access binary objects through the REST API.  
• Block storage: offers access-to-block storage devices by affixing volumes their current VM instances.  
• Shared File System storage: provides a set of services to manage multiple files together for storage and exchange with multiple users at one time.  
Ephemeral Storage: Referring to a single instance. As the name suggests, these storage options are temporary and short-lived and disappear once the VM is terminated.

**5.Define ‘users,’ ‘role’ and ‘tenant’ in OpenStack.**

Users can be members of multiple projects  
Tenant is a group of users and an alternative term for Project/accounts where projects are organizational units in cloud processing  
Role is the position to which a user is mapped (the authorization level). Roles are usually assigned to project-user duos.

**6.Define Identity Service in OpenStack.**

Keystone is the most important and preferred Identity Service in OpenStack and executes the complete OpenStack Identity API. The Keystone Identity Service is responsible for user management and service catalog. In user management, it tracks users and their permissions while Service Catalog offers a list of services available with their API. The former provides authentication credential details of users, tenants and roles.  
Internal services like Token and Policy are also part of Keystone Identity

**7.Define the Networking Managers in OpenStack.**

• Flat Network Manager: This places all VMs on a single network utilizing the same subnet and bridge as created by the administrator. Thus, all VMs share the same network that can be interconnected and are known to have Flat Network Manager.  
• Flat DHCP Network Manager: Much similar to the above except that the IP addresses to VM are assigned via DHCP (Dynamic Host Configuration Protocol).  
• VLAN: Unlike the single network concept, VLAN facilitates more secure and separate network to VMs. It has a physical switch to offer separate virtual network and separate IP range and bridge for each tenant. This is indeed most preferable choice for multi-tenant/project environment.

**8.Name the commands used to pause and un-pause(resume) an instance**

$ novaunpause INSTANCE\_NAME  
$ nova pause INSTANCE\_NAME

**9.List the storage locations for VM images in OpenStack**

• OpenStack Object Storage  
• Filesystem  
• S3  
• HTTP  
• RBD or Rados Block Device  
• GridFS

**10.What is Token?**

Token is a type of authentication similar to password-based validation. A token gets generated once the user inserts the credentials and authenticates as a Keystone user. The token can then be used to access OpenStack services without any revalidation. It is interesting to note that a token is active for a limited period and must be renewed after regular intervals.  
To create a token, users first need to authenticate their Keystone credentials.

**11.What is OpenStack Python SDK?**

Python SDK (Software Development Kit) helps users to write applications for performing automation tasks in Python by calling Python objects. It provides a platform to work with multiple OpenStack services at one place. It consists of language bindings to access OpenStack clouds, complete API reference, easy interaction with REST API and sample code for initial applications.

**12. Describe the function of Filter Scheduler.**

The Filter Scheduler facilitates filtering and weighting to notify where a new instance can be created. It supports working with Compute Nodes. Filter Scheduler firstly creates an unfiltered dictionary of hosts and then filter them using related properties and makes the final selection of hosts for the number of instances as needed.

**13.Define the Networking option in OpenStack.**

• AvalabilityZoneFilter: filters hosts by their availability zone.  
• CapacityFilter: filtering based on volume host’s capacity consumption  
• DifferentBackendFilter: Scheduling volumes to a different back-end  
• DriverFilter: filters based on ‘filter function’ and ‘metrics’  
• InstanceLocalityFilter  
• JSONFIlter  
• RetryFilter: Filter the previously attempted hosts  
• SameBackendFilter

**14. List down the Networking hardware in OpenStack.**

• Networks  
• Routers  
• Subnets  
• Ports Vendor Plugins

**15.Define Hypervisor**

For all cloud computing paltforms, Hypervisor is a term to define virtual machine monitor (VMM) including hardware, software and firmware components running on a virtual machine. Host machine is the one having hypervisor with one or more virtual machines.  
OpenStack Compute allows multiple hypervisors. There are functionalities to choose one among them for a specific purpose.

**16.List down the type of Hypervisors supported by OpenStack.**

• KVM (Kernel-based Virtual machine)  
• LXC: Linux Containers having Linux-based VMs  
• QEMU: Quick EMUlator used for development purposes  
• UML: User Mode Linux used for development purposes  
• VMware vSphere: VMware-based Linux and Windows via vCenter server connection.  
• Hyper-V: Server virtualization with Microsoft’s Hyper-V

**17.Explain in brief the modular architecture of OpenStack.**

The three important components of OpenStack modular architecture are:  
• OpenStack Compute: For managing large networks on the virtual machine  
• Image Service: The delivery service provides discovery and registration for virtual disk images  
• OpenStack Object Storage: A storage system that provides support for both block storage and object storage

**18.What command manages floating IP addresses in OpenStack**

nova floating-ip-\*

**19.Define bare-metal node.**

Bare-metal node grants access to control bare-metal driver that handles the provisioning of OpenStack Compute physical hardware utilizing the standard cloud APIs and tools like Heat. It is generally used for single tenant clouds like high-performance cluster computing. For using the bare-metal driver, a network interface must be created with the bare-metal node inserted into it. Afterwards, users can launch an instance from the node. Users can also list and delete bare-metal nodes by removing the associated network instances

**20.List down the components of OpenStack Compute**

Nova (Compute) Cloud comprises following components:  
• API server  
• Message Queue (Rabbit-MQ Server)  
• Compute Workers (Nova-Compute)  
• Network controller (Nova-Network)  
• Volume Worker  
• Scheduler

**21.Define the role of API Server.**

It provides an interface for the external world to interact with the cloud infrastructure.

**22.List the commands to generate Key pairs.**

• ssh-keygen  
• cd .ssh  
• nova keypair-add –pub\_key id\_rsa.pub mykey

**23.Define Flavor**

Flavors are virtual hardware templates present in OpenStack, which define the memory sizes of RAM, hard disk, etc. Flavors illustrate a number of parameters like ID, Name, Memory\_MB, Disk and others, giving a choice of Virtual Machine to the user just like having a physical server. OpenStack dashboard also allows users to modify a flavor by deleting the existing one and creating a new with the similar name and parameters.

**24.How to create a user in OpenStack?**

sudo nova-manage user create user-name

**25.How to assign a project/tenant to a user?**

By using the command sudo nova-manage user create user-name

**26.Can we see the list of roles and associated IDs in OpenStack environment?**

Yes, by using keystone role-list

# EC2 Interview – AWS Interview – Cloud Interview – 8 Questions

**1. Explain Elastic Block Storage?  What type of performance can you expect?  How do you back it up?  How do you improve performance?**

EBS is a virtualized SAN or storage area network.  That means it is RAID storage to start with so it's redundant and fault tolerant.  If disks die in that RAID you don't lose data.  Great!  It is also virtualized, so you can provision and allocate storage, and attach it to your server with various API calls.  No calling the storage expert and asking him or her to run specialized commands from the hardware vendor.

Performance on EBS can exhibit variability.  That is it can go above the SLA performance level, then drop below it.  The SLA provides you with an average disk I/O rate you can expect.  This can frustrate some folks especially performance experts who expect reliable and consistent disk throughput on a server.  Traditional physically hosted servers behave that way.  Virtual AWS instances do not.

Backup EBS volumes by using the snapshot facility via API call or via a GUI interface like elasticfox.

Improve performance by using Linux software raid and striping across four volumes.

**2. What is S3?  What is it used for?  Should encryption be used?**

S3 stands for Simple Storage Service.  You can think of it like ftp storage, where you can move files to and from there, but not mount it like a filesystem.  AWS automatically puts your snapshots there, as well as AMIs there.  Encryption should be considered for sensitive data, as S3 is a proprietary technology developed by Amazon themselves, and as yet unproven vis-a-vis a security standpoint.

**3. What is an AMI?  How do I build one?**

AMI stands for Amazon Machine Image.  It is effectively a snapshot of the root filesystem.  Commodity hardware servers have a bios that points the the master boot record of the first block on a disk.  A disk image though can sit anywhere physically on a disk, so Linux can boot from an arbitrary location on the EBS storage network.

Build a new AMI by first spinning up and instance from a trusted AMI.  Then adding packages and components as required.  Be wary of putting sensitive data onto an AMI.  For instance your access credentials should be added to an instance after spinup.  With a database, mount an outside volume that holds your MySQL data after spinup as well.

**4. Can I vertically scale an Amazon instance?  How?**

Yes.  This is an incredible feature of AWS and cloud virtualization.  Spinup a new larger instance than the one you are currently running.  Pause that instance and detach the root ebs volume from this server and discard.  Then stop your live instance, detach its root volume.  Note the unique device ID and attach that root volume to your new server.   And the start it again.  Voila you have scaled vertically in-place!!

**5. What is auto-scaling?  How does it work?**

Autoscaling is a feature of AWS which allows you to configure and automatically provision and spinup new instances without the need for your intervention.  You do this by setting thresholds and metrics to monitor.  When those thresholds are crossed a new instance of your choosing will be spun up, configured, and rolled into the load balancer pool.  Voila you've scaled horizontally without any operator intervention!

**6. What automation tools can I use to spinup servers?**

The most obvious way is to roll-your-own scripts, and use the AWS API tools.  Such scripts could be written in bash, perl or other language or your choice.  Next option is to use a configuration management and provisioning tool like puppet or better it's successor Opscode Chef.  You might also look towards a tool like Scalr.  Lastly you can go with a managed solution such as Rightscale.

**7. What is configuration management?  Why would I want to use it with cloud provisioning of resources?**

Configuration management has been around for a long time in web operations and systems administration.  Yet the cultural popularity of it has been limited.  Most systems administrators configure machines as software was developed before version control - that is manually making changes on servers.  Each server can then and usually is slightly different.  Troubleshooting though is straightforward as you login to the box and operate on it directly.  Configuration management brings a large automation tool into the picture, managing servers like strings of a puppet.  This forces standardization, best practices, and reproducibility as all configs are versioned and managed.  It also introduces a new way of working which is the biggest hurdle to its adoption.

Enter the cloud, and configuration management becomes even more critical.  That's because virtual servers such as amazons EC2 instances are much less reliable than physical ones.  You absolutely need a mechanism to rebuild them as-is at any moment.  This pushes best practices like automation, reproducibility and disaster recovery into center stage.

**8. Explain how you would simulate perimeter security using Amazon Web Services model?**

Traditional perimeter security that we're already familiar with using firewalls and so forth is not supported in the Amazon EC2 world.  AWS supports security groups.  One can create a security group for a jump box with ssh access - only port 22 open.  From there a webserver group and database group are created.  The webserver group allows 80 and 443 from the world, but port 22 \*only\* from the jump box group.  Further the database group allows port 3306 from the webserver group and port 22 from the jump box group.  Add any machines to the webserver group and they can all hit the database.  No one from the world can, and no one can directly ssh to any of your boxes.

Want to further lock this configuration down?  Only allow ssh access from specific IP addresses on your network, or allow just your subnet.

**1)** What is Amazon EC2 service?

**Ans:** EC2 uses Xen virtualization. Each virtual machine, called an “instance”. You can use Amazon EC2 to launch as many or as couple of virtual servers as you need, design security and networking, and manage storage. Amazon EC2 empowers you to scale up or down to handle changes in requirements.

**2)** What is Amazon Machine Image (AMI)?

**Ans:** An Amazon Machine Image (AMI) is a template that contains a software configuration (for example, an operating system, an application server, and applications). From an AMI, we launch an instance, which is a copy of the AMI running as a virtual server in the cloud. We can launch multiple instances of an AMI.

**3)** What is the relation between Instance and AMI?

**Ans:** We can launch different types of instances from a single AMI. An instance type essentially determines the hardware of the host computer used for your instance. Each instance type offers different compute and memory capabilities.

After we launch an instance, it looks like a traditional host, and we can interact with it as we would any computer. We have complete control of our instances; we can use sudo to run commands that require root privileges.

**4)** What are the Security Best Practices for Amazon EC2?

**Ans:** There are several best practices for secure Amazon EC2. Following are few of them.

 Use AWS Identity and Access Management (IAM) to control access to your AWS resources.

 Restrict access by only allowing trusted hosts or networks to access ports on your instance.

 Review the rules in your security groups regularly, and ensure that you apply the principle of least

 Privilege — only open up permissions that you require.

 Disable password-based logins for instances launched from your AMI. Passwords can be found or cracked, and are a security risk.

**5)** Explain Stopping, Starting, and Terminating an Amazon EC2 instance?

**Ans:**

**Stopping and Starting an instance:** When an instance is stopped, the instance performs a normal shutdown and then transitions to a stopped state. All of its Amazon EBS volumes remain attached, and you can start the instance again at a later time. You are not charged for additional instance hours while the instance is in a stopped state.

**Terminating an instance:** When an instance is terminated, the instance performs a normal shutdown, then the attached Amazon EBS volumes are deleted unless the volume’s deleteOnTermination attribute is set to false. The instance itself is also deleted, and you can’t start the instance again at a later time.

**6)** How to use Amazon SQS?

**Ans:** Amazon SQS is a message passing mechanism that is used for communication between different connectors that are connected with each other. It also acts as a communicator between various components of Amazon. It keeps all the different functional components together. This functionality helps different components to be loosely coupled, and provide an architecture that is more failure resilient system.

**7)** How buffer is used in Amazon web services?

**Ans:** Buffer is used to make the system more resilient to burst of traffic or load by synchronizing different component. The components always receive and process the requests in unbalanced way. Buffer keeps the balance between different components and makes them work at the same speed to provide faster services.

**8)** How does cloud computing provides on-demand functionality?

**Ans:** Cloud computing is a metaphor used for internet. It provides on-demand access to virtualized IT resources that can be shared by others or subscribed by you. It provides an easy way to provide configurable resources by taking it from a shared pool. The pool consists of networks, servers, storage, applications and services.

**9)** What is the difference between scalability and elasticity?

**Ans:** Scalability is a characteristic of cloud computing through which increasing workload can be handled by increasing in proportion the amount of resource capacity. It allows the architecture to provide on demand resources if the requirement is being raised by the traffic. Whereas, elasticity is being one of the characteristic provide the concept of commissioning and decommissioning of large amount of resource capacity dynamically. It is measured by the speed by which the resources are coming on demand and the usage of the resources.

**10)** What are the different layers of cloud computing?

**Ans:** Cloud computing consists of 3 layers in the hierarchy and these are as follows:

**1.** Infrastructure as a Service (IaaS) provides cloud infrastructure in terms of hardware like memory, processor speed etc.

**2.** Platform as a Service (PaaS) provides cloud application platform for the developers.

**3.** Software as a Service (SaaS) provides cloud applications which are used by the user directly without installing anything on the system. The application remains on the cloud and it can be saved and edited in there only.

**11)** How to secure your data for transport in cloud?

**Ans:** Cloud computing provides very good and easy to use feature to an organization, but at the same time it brings lots of question that how secure is the data, which has to be transported from one place to another in cloud. So, to make sure it remains secure when it moves from point A to point B in cloud, check that there is no data leak with the encryption key implemented with the data you sending.