Cloud computing 3 types:

1. IAAS :

Infrastructure as a service will provide a hard ware and networking requirement.

1. PAAS:

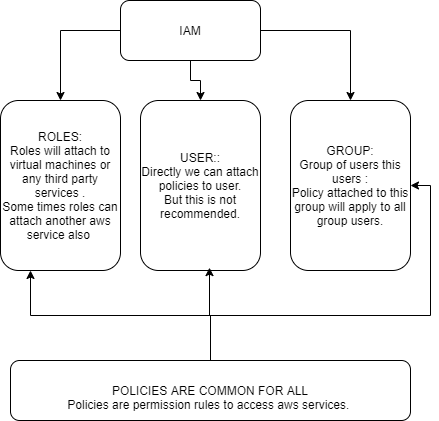
Platform as a service will provide the hardware, networking and operating system level.

Example: AWS

1. SAAS:

Software as a service will provide up to software level: Directly we can use their software . Example: gmail

IAM (Identity access management)



1. IAM is visible like global.
2. IAM user has a separate link to access AWS services , This like can be customizable

Note: At the time of creating aws account it will provide two logins

a) uid & password b) Access key & Secret key

3) MFA: Multi factor authenticator is using external keys to authenticate.

4) Common policy to automatically attach to every user is : IAMuserChangePolicy

ec2-instance(Elastic Complete Cloud)

families of ec2 instances:

Micro instance,General Purpose,Compute Optimized,Memory Optimized,Accelerated Computing Optimized,Storage Optimized,GPU optimized.

**Images id** is different for each region for same IMI image.

User Data: This is a code to run when ec2 instance is launching.

If restarted public ip address will not change. If shut down and start public ip address will change.

If terminated cpu and ram will not charge. Only Storage will charge.

Tags will help to inspector to check security reached certainexpedition.

Ec2 default storage ebs.

**Types of ec2 instance Based on pricing:**

[Amazon EC2 Spot Instances](https://aws.amazon.com/ec2/spot/): This cost will change frequently ,bidding will going on every second, If anybody bid more price ec2 instance will removed from the poll and assigned to higher bidder.

[Amazon EC2 Reserved Instances](https://aws.amazon.com/ec2/pricing/reserved-instances/): This will fixed time period 1 or 3 years , Bill will generate even it is not used.

[Amazon EC2 Dedicated Hosts](https://aws.amazon.com/ec2/dedicated-hosts/): Separate physical machine will assigned we can visible the cores.

Amazon On demand Instance (default): Immediately we Can create on demand instance

Amazon Scheduled Instance: This is applicable on ondemand instances Example: if we give scheduel on on demand instance that instance will work its maximum effort, remaining time that capacity will be utilized by amazon . For this reason amazon will give points to users. This will applicable to running instance only.

We can use ec2 instance as NAT instance.

To keep root volume after ec2 termination keeps check mark at deletion on termination.

Boot volume cannot be encrypted, To encrypt separated software like bit locker is required.

Additional volumes can be attached.

Volumes attached names will be like below:

/dev/sda or /dev/sda1

/dev/xvda or /dev/xvda1

Exam question: How to connect ec2 instance to s3 bucket

1. By login to ec2 instance and connect s3 buckets by using iam user access key and secret key. But this is not recommended because access key and secret key both are saved in ec2 machine.

Anybody can take and misuse those credentials.

1. Created one role with s3 access policies .Assign that role to ec2 instance.

Aws s3 ls: To display all s3 buckets

Aws s3 cp sample.txt s3://bucketname/sample.txt

Aws s3 cp s3://bucketname/sample.txt ./

Aws s3 sync /home/ec2-user/ s3://bucketname

Basic Monitoring : Basic monitoring is provided for AWS ecc instance

1. CPU : utilization
2. Disk : utilization
3. Network : utilization
4. Status check:
5. System status check :

Power and network status: If this status is not good restart the instance otherwise contact support

1. Instance status check:

There is a software problem installed in the instance .If status is not good check software’s.

Ingress- Inbound

Egress-Outbound

How to attach ebs volume to ec2 instance :

While attaching the EBS volume to ec2 instance ec2 instance is in same availability zone.

Process to attach EBS volume:

1. Attach from ui
2. Login to ec instance :
3. File –s /dev/xvdf : check the file system is existed
4. Mkfs –t ext4 /dev/xvdf : make the the file system if file system not exist
5. Go to root location / and created folder mkdir <fileserver>
6. Mount /dev/xvdf fileserver: Mount the volume to directory .

How to detach volume from ec2 instance :

1. Unmount
2. Detach from ui
3. Take a snapshot
4. Remove volume

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

STORAGE CLASSES

|  |  |  |  |
| --- | --- | --- | --- |
| Standard | >=3 AZ |  | Frequently access class(Default) |
| Standard-IA | >=3AZ | Min 30 days | Infrequently access class |
| Intelligent tiring | >=3AZ | Min 30 days | Combination of standard and standard –IA. Logic is maintained by Amazon. Some time it will sored in standard and sometime standard IA. |
| One zone Infrequent access | >=1AZ | Min 30 days | It will store non critical data. |
| Glacier | >=3AZ |  | Old data retrieving will take minutes to hours. |
| Reduced redundancy | >=3AZ |  | Frequently accessed non non critical data like thumbanions. |

s3:(OBJECT STORAGE) simple storage service.: High scalable, High reliable, Object based storage.

1 ) If s3 is created we can get it by another availability zone.

2) s3 is unique and universal name in aws world.

3) s3 Visibly it is to be global but it is regional specific.

4) Cross account access is work but policy should describe in role.

5) Complete static website we can host

6) Data Can be directly stored /retrieve using rest api, Exposed by s3

7) We can enable logs in s3 buckets.

8) Consistency rules: 1) Read after write consistency 2)Eventual consistency , Every get and put request sometimes refresh is required.

9) object url to access : https://s3-<regionname>.amazon.com/<bucketname>/<filename>

10) charges criteria based on 1)storage 2)Requests 3)Data Transfer

11) Enabling versioning, we can upload multiple files with same name. After enabling we cannot disable versioning, we can only suspend it only.

12) To enable cross region replication must and should we have to enable versioning

13) We can access s3 buckets with get,and put requests from browser. We can access from cli by using Access key id, Secret key id, iam role

14) We can create events, ex anything changed in s3 buckets this event will trigger. (SNS(Simple Notification Service),LAMBDA FUNCTION(this is a function when specific event occurs) ,sqsque)

15) Once object max size 5tb. In one bucket we can store n number of objects. So unlimited data.

16) Transfer Acceleration: This will enable us to upload objects to long distance buckets. This will create one endpoint near US. We will upload our objects to near endpoint. From near endpoint it goes throw amazon network. This will provide us secure.

17) We can copy object from one one region s3 to another region s3 by specifying path.

|  |  |
| --- | --- |
| Object | Bucket |
| Lock is possible | Versioning is enables for bucket level by default it is disabled |
| Object can share another Amazon account .Object can share public also. | Static website can store .  Select host option select web pages . |
| We can make object public but bucket level permission is required | Server access logs can enabled in bucket level . one bucket access logs can store in another bucket. |
| Object level logging is possible with cloud trail ,Cloud trail is required to record events .  This service is available in management and governance level. | Bucket level logging is possible we can store the logs in another bucket |
| Encryption is possible | Encryption is possible |
| We can attach tags in object level. Tags is useful to to identify the object. | Tags can attach bucket level also |

Bucket life cycle Management:

1 ) Life cycle we can apply to bucket level as well as object level also. While applying object level we need to select object pattern like regex.

2)Minimum 30 days is required for infrequent access and 1zone infrequent access, intelegent tyring .

Replication of s3:

1. We can share the owner ship of the bucket to another account.
2. IAM role is required to use s3 bucket by another s3 bucket.
3. Object is replicated after cross region replication is enabled. Old object will not be replicated
4. To enable cross region replication must and should we have to enable versioning
5. While creating replication rules source and destination bucket name should specified with object paths. Complete s3 replication is not possible only object replication is possible.

It required specifying source and destination buckets names along with regex of objects.

Analytics: This analytics are based on daily, weekly, report of s3 bucket this is called inventory.

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

VPC(Virtual Private Cloud)

1. While creating VPC CIDR block should specify ,While subnet is creating cidr for sub net will come from range of vpc cidr range.
2. Private Subnet and public subnet both have route table and Nacl , But public subnet have IGW.
3. /16 is highest number in vpc in aws world.

**When creating AWS account we will get default vpc in our account. Same way default subnet also present. Default routetable also present.This default route table attached IGW**.

By default ec2 instance will launch in default VPC.

Default VPC can be deleted, If want again contact amazon customer care.

Dedicated/default: Dedicated the resource to us. Default means sharing the resource with isolated.

VPC is region specific.

DHCP will asign ip address to Resources.

Availability zones to save from disaster.

VPC service is free.

By default, one route table is created for every VPC. This route table(default route table) connected with IGW.(internet Gate Way)

<https://www.ipaddressguide.com/cidr>

VPC-Peering

We can connect multiple VPC with VPC peering.

With VPC-peering we can use one VPC resources(ex : ec2) in another VPC.

This can be possible in same account and different account.

This can be possible in same region with different region.

Transitive peering is not supporting.

Example: If vpc1 is peering with vpc2 , vpc2 is peering with vpc3 .

Here vpc1 resource cannot used by vpc 3.This is called Transient peering;this is not possible.

While peering VPC, CYDER block range will not the same

Process: Assume Two vpc are there VPC1 and VPC2.

Modify the route table of one VPC by placing another VPC ' s CIDR block. Do this for both VPC.

**Subnets**

5 ip addresses are required for amazon internal use, when creating subnets.

Private Subnet: IGW not attached

Public Subnet: IGW attached.

If new subnet is created that new sub net is associated with default route table.

We can provide internet to private sub net with one-way internet with using NAT instance,NAT gate way.

**How are you communicating yours communicate your corporate network with VPC?**

VPC(aws).



CORPORATE NETWORK.(on premise)

Ec2-with privateip.

Ec2-with public IP

Ec2-with privateip.

Ec2-with private ip

Corporate Network

Ec2-with private ip

@Note data can be encrypted and decrypted.

Ece-with private ip

@vpc is like VPN in or premise

Ec2-private

Ec2-private

Ec2-private

Corporate Network

Private wired communication between Corporation Network and VPC

DIRECT CONNECT

NACL: Default one NACL is created for VPC that default nacl will applicable all subnets in that vpv. This default nacl will allow all traffic to subnet.

If new custom nacl is created that nacl will stop all traffic .

One NACL can apply multiple subnets.

Rules are in multiple of hundred. Because rules will apply from precedence.

Example: If 100 rule is allowing and 101 is not allowing . In this situation 100 rule allow is work.

**In bound and out bound should mention because it is stateless.**

**While specifying outbound rules in nacl we should specify aperimal-ports from 1024-65535. Because out going traffic will go from this.**

Flow Logs

1. Applicable to VPC
2. We can send this logs to s3 and cloud watch(to send logs to cloud watch cloud watch group is required)
3. Role is required to attach the flow log.

Difference between NACL & SUCURITY GROUP.

|  |  |
| --- | --- |
| This is virtual fire wall of the sub net.  This is applicable to subnet level.  Stateless(Inbound rule cannot be remembered)  We can control ip level blocking  By default, every subnet has it own NACL.  This will allow all traffic inside and outside.  Only one NACL is allowed for one subnet.  It can control in bound and out bounds.  Example: This will accept http traffic inside but it can stop outbound traffic of http.  Note: Rule format should be 100,200,300. Because we can add another rule between these numbers.  Rule will be applicable from less number to higher number. | This is virtual fire wall for ec2,elb,alb,AutoScalling Groups.  State full (In bound rules will remember)  No ip level control.  At least one security group is required for one ENI(Electronic network interface). For one ENI max 5 security groups can be attached.  This controls Inbound but cannot control out bounds.  Example: This will accept http traffic inside but it cannot stop outbound traffic of http.  Note: If it accepts http request ,it automatically accept out traffic of http. |

Difference between NAT gate way and NAT instance

|  |  |
| --- | --- |
| This is always in public subnet.  It just forward the traffic from/to the private subnet. It will scale upto 45GBps  Elastic Ip address is required for Nat-Gate Way  It just forward the traffic from/to the private subnet. It will scale upto 45GBps.  Nat-gate way took private and public ip address.  Bandwidth is maintained by amazon  Private ip address will assign from available subnet range.  This is best for production  SSh is not possible because this is not a machine.  Shut down is not possible.  Bill will be calculated based on amount of date transferred.  We cannot configure security groups, But this gate way is in Public subnet so NACL(Subnet level) rule will applicable.  Eip is required for this. | This is always in public subnet.  This bandwidth is depending on instance type which we selected.  Port forwarding is possible.  We can change the elastic ip address to this instance  Best for testing purpose,because if bandwidth wants to increase we should shut down and start another NAT-instance(ec2)  We can stop/start this NAT-instance  Ssh is possible Nat instance is a machine(ec2-instance).  Shut down is possible.  We can configure security group because this is ec2 instance(machine in cloud).  Data coming from aws cloud is free and data coming from out side world is chargeable.  We can disable ip address of nat instance.  NAT AMI is available in community AMI of amazon.  Disable the source and destination check will disable to prevent checking.  Assign one elastic public ip address to nat instance is required. |

Baston Host

1. This is like a step-up server
2. This will useful to connect private subnet instance from public subnet.
3. The basten host contains separate access list for separate group.
4. This is a way to access the team to the private sub net.

AWS RDS (Relational Database Back Up).

|  |  |  |
| --- | --- | --- |
| Free | paid | From aws |
| mysql | Ms-sql | Maria db |
| Postgrace sql | oracle | aurora |

Above all are relational data bases.

One non relational data base is called Dynamo db by aws.

CREATING-DB

Select db engine(ex-mysql)>select version>DB instance class(nano,micro)>select vpc,subnet>select encrypted or not>select uidpwd> select Backup retention period(7 -35)>enable monitoring (optional)>Automatic version upgard option>Deletion protection option

1. We can enable monitoring by default error logs will send to cloud watch(cloud watch group is required)
2. Automatically amazon will take back up of db instance. But we should specify the retention period of the db from 7 to 35 days.
3. We can see logs in db side no need to go to cloud watch.

|  |  |
| --- | --- |
| Automatic Backup | Manual Backup |
| 1)At any time backup will available in amazon  2)Automatic day backup is a full daily snapshots and will also store transaction logs throughout the day  3)When recovery aws will take most recent daily backup and then apply transaction log relevant to thatday.  This allow us to do a point in time recovery down to a second with in the retention period.  4)Automatic backup are enabled by default.  5) We will |  |

**Horizontal scaling** means that you **scale** by adding more machines into your pool of resources.

Vertical Scaling means increasing CPU and RAM to existing machine.

Chargable:

Load Balancers :Application Load Balancers(ALB),Network Load Balancers.(NLB),Elastic Load Balancers.(ELB)

Application Load Balancer: (layer 7)

This will support http,https.This is specific to vpc. Used for micro services.

AlB follows context path ec2 based routing.

Target Group: Logical group of servers associated with your load balancer.

Based on rules request will forwarded to certain target group. In target group request is processed and response will give to load balancers.

Existing machine.We can attach security group to ALB.

LISTNER1

We can enable termination protection to ALB.

EC2

ALB

We can enable access logs.We can enable multiple request types.

EC2

EC22

EC2

LISTNER2

States Of ALB: Provisioning, active, Free

ELASTIC LOAD BALANCER(ELB)

This is specific to vpc. But we can do with multiple regions with the help of route53.

ELB can secured with security group.

All the servers contain same content.

Load will be distributed among all servers.

If will perform health checks then it will send traffic to healthyresource.

Unhealthy Thresh hold: This is a numeric value we have to give in ELB configuration .

example: If value is 2 , If two conjugative health checks fails. This resource ec2 instance, will mark as unhealthy and resource will remove from the service.

Healthy Thresh hold: This is a numeric value we have to give in ELB configuration

example: If value is 2 , If two conjugative health checks pass. This resource will mark as healthy and resource will added to service.

ELP policy: Round Robin.

Note: We can add or remove ec2 instance at any time.

Cross Zone Load Balancing: Load will distributed equally among the zones.

Connection Draining: Time to complete active request to the servers. After completing active requests LB will remove server from the service.

We can access load balancers with end points.

Enable Load Balancer Generated cookie: Request from specific Clint is bound with same server .

All request will goes to same server until the specific time period.

Load Balancers follows round Robin algorithm.

Ideal Time Out: Waiting for a response from a ec2 instance , If ec2 instance is not responded with in a time Load balancer will send connection time out error to end user.

Acces Log: To Monitors the request and response , We should provide a s3 bucket name in this region.

Auto Scalling

This is free.

launch Configuration: This is a template this contains all configuration for new resource.

1)AMI 2)Security Group 3)volume type 4)IAM role(some time ec2instance will talk to s3 so its required) 5) Product Key

It can be integrated with load balancers

**Alarm is required for auto scaling**

Increase group size decrease group size required alarm:

Example: If cpu utilization is increase more than 50%es alarm fires , If alarm fires instance will up based on launch configuration.

We can send notifications. AS will check server ec2 health checks.

Desired capacity will change based on policies.

Straight options to test increase group size, decrease group size we can manually click the button and check.

We can schedule auto scale capacity.

EBS & SNAP SHOTS

Elastic Block Storage:

5types 3 types we can use as boot devise.

1) General Purpose: ssd

2) Provisioned IOPS: These will work as boot volume.

3) Magnetic:

4) Cold HDD: These are not used for boot volume.

5) Throw put optimized.

We can increase EBS volume when ec2 instance is running, but we can’t decrease the EBS volume.

Multiple EBS can attach with single instance (ec2)

Snap Shots:

1. We can take a snap shot of root volume
2. From root volume snap shot we can convert it as image.
3. Above image is available while creating ec2 instance.
4. We can copy snap shots from one region to another region with encrypted option.
5. We can share snap shot to another account by specifying another account id.

How to increase the volume:

1. Stop all IO operations on that volume
2. Created snap shot and recreate the snap shot with big volume .
3. Mount that volume.

AMI(Amazon Machine Images)

1. AMI contains all data and applications of the one ec2 instances .
2. While crating the images from ec2 instance automatically ec2 instance will shut down by default. Creating the image without shutdown will leads to data inconsistency.
3. If AMI is created from one ec2 instance that AMI will specific to that region only.
4. We can copy AMI one region to another region with encrypted option.
5. AMI image can be shared to specific aws account id . We can make it as public also.
6. While creating the AMI snap shot of the volumes will created in that region.

EFS

This is like NFS in lan.

File Based storage.

This disk can be used across network, across region,

It can be mounted to a premise server as well (OVER VPN (or) Direct Connect.

Can be mounted to multiple ec2 instance at a time.

Sizening is not required.(It scales Automatically)

EFS will take ip address from subnet range.

Efs will take electronic network interface which contains security group.

EFS mounting is a os level activity so its not possible to mount with cli or web console.

terraform:(0.11.3)

Aws Authentication types in terra form.

1) static credentials : These are mention in the terraform file itself(hard coded).

2)Environment Variable: These credentials are exported in current machine.

3) shared credentials : These are written in another files and refer in this main terraform file.

4)Ec2 roles.

Infrastructure as a code.

support planning from execution.

Terraform state files: Every run of terrafom, state of the terraform will store in statefiles

1).tfstate

2).tfstat backup

To Access the attribute of the terraform we us below syntax.

Syntax: $(resource\_type.logicalname.parameter\_name)

example:

Terraform will identify what resource is created in earlier execution.

Terraform will not change the ec2instance key, it will remove and recreate entire ec2 instance.

terraform config: To provide accesskeyid and secretaccess key to terraform engine.

terraform plan : This will display what changes to b apply on environment.

terraform init : To update all plug-in to connect providers.

terraform apply : To apply the terraform to executing in Infrastructure .

terraform fmt : To check the code properly alligned or not.

terraform console: To check interpolation(replacing the values) out put in a console.

terraform import : To get resource into terraform control from outside

Terraform support different types of datatypes.

string : "sample\_string"

List : ["sanjay" ,"Thanvik", "vaishnavi", " vhinnu"]

Map : {

key1=value

key2=value

}

#####################################################################################

Variable Declaration in terraform:

String:

variable "region" {

type="string"

default="south\_india"

}

List:

variable "fruits" {

type="List"

default=["mango" ,"apple","pinapple"]

} Map:

varaible "imageid" {

type="Map"

default={

"andhrapradesh"="amaravathi"

"telengana" = "hyderabad"

}

&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&

Input variables of a terraform is :

varaible "imageid" {}: This will expect input from user or machine.

output variable of a terraform is:

output "ami\_name"{

value="sanjaykumarreddy"

}

$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$

To using variable values: ${vars.imageid} ,${vars.fruits},${vars.region}

To loop the code we can use: count variable.

Length (list): This method will take list and return length of the list.

Data source: This will useful when dynamic values required at the time of running.

lookupmethod(map,key): This method will return the value from map with specific key .

**templates**: Templates is structured file for reusing by reusing in terraform we can replace the values with our values.

Modules: Collection of terraform templates.

To import the modules:

**Collection Of terra form folder is called a module**.

modules "samole\_module"{

source="path\_of\_the\_module(folder)"

}

After importing we can use all variables in that modules by using below syntax.

module.modulename.variable\_name

# How to Install Kubernetes (k8s) 1.7 on CentOS 7 / RHEL 7

by [Pradeep Kumar](https://www.linuxtechi.com/author/pradeep/) · Published September 4, 2017 · Updated December 12, 2017

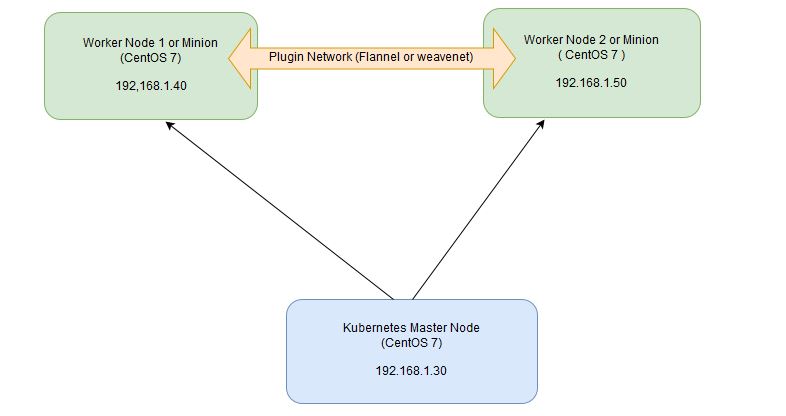
Kubernetes is a **cluster** and **orchestration** engine for docker containers. In other words Kubernetes is  an open source software or tool which is used to orchestrate and manage docker containers in cluster environment. Kubernetes is also known as k8s and it was developed by Google and donated to “Cloud Native Computing foundation”

In Kubernetes setup we have one master node and multiple nodes. Cluster nodes is known as worker node or Minion. From the master node we manage the cluster and its nodes using ‘**kubeadm**‘ and ‘**kubectl**‘  command.

Kubernetes can be installed and deployed using following methods:

* Minikube ( It is a single node kubernetes cluster)
* Kops ( Multi node kubernetes setup into AWS )
* Kubeadm ( Multi Node Cluster in our own premises)

In this article we will install latest version of Kubernetes 1.7 on CentOS 7 / RHEL 7 with kubeadm utility. In my setup I am taking three CentOS 7 servers with minimal installation. One server will acts master node and rest two servers will be minion or worker nodes.

[](https://www.linuxtechi.com/wp-content/uploads/2017/09/Kubernetes-settup-Diagram.jpg)

#### On the Master Node following components will be installed

* **API Server**  – It provides kubernetes API using Jason / Yaml over http, states of API objects are stored in etcd
* **Scheduler**– It is a program on master node which performs the scheduling tasks like launching containers in worker nodes based on resource availability
* **Controller Manager** – Main Job of Controller manager is to monitor replication controllers and create pods to maintain desired state.
* **etcd** – It is a Key value pair data base. It stores configuration data of cluster and cluster state.
* **Kubectl utility** – It is a command line utility which connects to API Server on port 6443. It is used by administrators to create pods, services etc.

#### On Worker Nodes following components will be installed

* **Kubelet** – It is an agent which runs on every worker node, it connects to docker  and takes care of creating, starting, deleting containers.
* **Kube-Proxy** – It routes the traffic to appropriate containers based on ip address and port number of the incoming request. In other words we can say it is used for port translation.
* **Pod** – Pod can be defined as a multi-tier or group of containers that are deployed on a single worker node or docker host.

### Installations Steps of Kubernetes 1.7 on CentOS 7 / RHEL 7

**Perform the following steps on Master Node**

#### Step 1: Disable SELinux & setup firewall rules

Login to your kubernetes master node and set the hostname and disable selinux using following commands

~]# hostnamectl set-hostname 'k8s-master'

~]# exec bash

~]# setenforce 0

~]# sed -i --follow-symlinks 's/SELINUX=enforcing/SELINUX=disabled/g' /etc/sysconfig/selinux

Set the following firewall rules.

[root@k8s-master ~]# firewall-cmd --permanent --add-port=6443/tcp

[root@k8s-master ~]# firewall-cmd --permanent --add-port=2379-2380/tcp

[root@k8s-master ~]# firewall-cmd --permanent --add-port=10250/tcp

[root@k8s-master ~]# firewall-cmd --permanent --add-port=10251/tcp

[root@k8s-master ~]# firewall-cmd --permanent --add-port=10252/tcp

[root@k8s-master ~]# firewall-cmd --permanent --add-port=10255/tcp

[root@k8s-master ~]# firewall-cmd --reload

[root@k8s-master ~]# modprobe br\_netfilter

[root@k8s-master ~]# echo '1' > /proc/sys/net/bridge/bridge-nf-call-iptables

**Note:** In case you don’t have your own dns server then update /etc/hosts file on master and worker nodes

192.168.1.30 k8s-master

192.168.1.40 worker-node1

192.168.1.50 worker-node2

#### Step 2: Configure Kubernetes Repository

Kubernetes packages are not available in the default CentOS 7 & RHEL 7 repositories, Use below command to configure its package repositories.

[root@k8s-master ~]# cat <<EOF > /etc/yum.repos.d/kubernetes.repo

> [kubernetes]

> name=Kubernetes

> baseurl=https://packages.cloud.google.com/yum/repos/kubernetes-el7-x86\_64

> enabled=1

> gpgcheck=1

> repo\_gpgcheck=1

> gpgkey=https://packages.cloud.google.com/yum/doc/yum-key.gpg

>         https://packages.cloud.google.com/yum/doc/rpm-package-key.gpg

> EOF [root@k8s-master ~]#

#### Step 3: Install Kubeadm and Docker

Once the package repositories are configured, run the beneath command to install kubeadm and docker packages.

[root@k8s-master ~]# yum install kubeadm docker -y

Start and enable kubectl and docker service

[root@k8s-master ~]# systemctl restart docker && systemctl enable docker

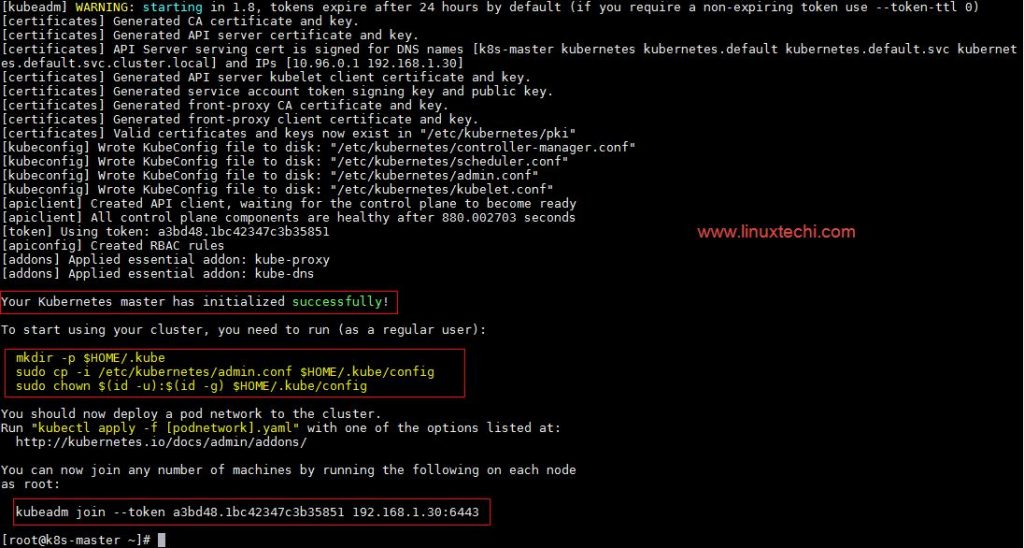
[root@k8s-master ~]# systemctl  restart kubelet && systemctl enable kubelet

#### Step 4: Initialize Kubernetes Master with ‘kubeadm init’

Run the beneath command to  initialize and setup kubernetes master.

[root@k8s-master ~]# kubeadm init

Output of above command would be something like below

[](https://www.linuxtechi.com/wp-content/uploads/2017/09/kubeadm-init-output.jpg)

As we can see in the output that kubernetes master has been initialized successfully. Execute the beneath commands to use the cluster as root user.

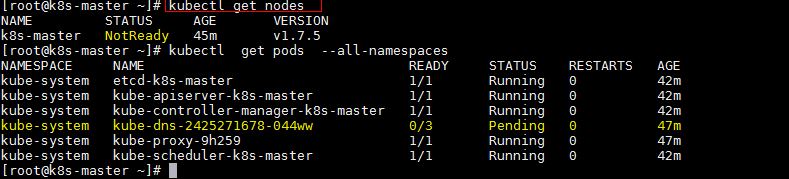
[root@k8s-master ~]# mkdir -p $HOME/.kube

[root@k8s-master ~]# cp -i /etc/kubernetes/admin.conf $HOME/.kube/config

[root@k8s-master ~]# chown $(id -u):$(id -g) $HOME/.kube/config

#### Step 5: Deploy pod network to the cluster

Try to run below commands to get status of cluster and pods.

[](https://www.linuxtechi.com/wp-content/uploads/2017/09/kubectl-get-nodes.jpg)

To make the cluster status ready and kube-dns status running, deploy the pod network so that containers of different host communicated each other.  POD network is the overlay network between the worker nodes.

Run the beneath command to deploy network.

[root@k8s-master ~]# export kubever=$(kubectl version | base64 | tr -d '\n')

[root@k8s-master ~]# kubectl apply -f "https://cloud.weave.works/k8s/net?k8s-version=$kubever"

serviceaccount "weave-net" created

clusterrole "weave-net" created

clusterrolebinding "weave-net" created

daemonset "weave-net" created

[root@k8s-master ~]#

Now run the following commands to verify the status

[root@k8s-master ~]# kubectl get nodes

NAME         STATUS    AGE       VERSION

k8s-master   Ready     1h        v1.7.5

[root@k8s-master ~]# kubectl  get pods  --all-namespaces

NAMESPACE     NAME                                 READY     STATUS    RESTARTS   AGE

kube-system   etcd-k8s-master                      1/1       Running   0          57m

kube-system   kube-apiserver-k8s-master            1/1       Running   0          57m

kube-system   kube-controller-manager-k8s-master   1/1       Running   0          57m

kube-system   kube-dns-2425271678-044ww            3/3       Running   0          1h

kube-system   kube-proxy-9h259                     1/1       Running   0          1h

kube-system   kube-scheduler-k8s-master            1/1       Running   0          57m

kube-system   weave-net-hdjzd                      2/2       Running   0          7m

[root@k8s-master ~]#

Now let’s add worker nodes to the Kubernetes master nodes.

### Perform the following steps on each worker node

#### Step 1: Disable SELinux & configure firewall rules on both the nodes

Before disabling SELinux set the hostname on the both nodes as ‘worker-node1’ and ‘worker-node2’ respectively

~]# setenforce 0

~]# sed -i --follow-symlinks 's/SELINUX=enforcing/SELINUX=disabled/g' /etc/sysconfig/selinux

~]# firewall-cmd --permanent --add-port=10250/tcp

~]# firewall-cmd --permanent --add-port=10255/tcp

~]# firewall-cmd --permanent --add-port=30000-32767/tcp

~]# firewall-cmd --permanent --add-port=6783/tcp

~]# firewall-cmd  --reload

~]# echo '1' > /proc/sys/net/bridge/bridge-nf-call-iptables

#### Step 2: Configure Kubernetes Repositories on both worker nodes

~]# cat <<EOF > /etc/yum.repos.d/kubernetes.repo

> [kubernetes]

> name=Kubernetes

> baseurl=https://packages.cloud.google.com/yum/repos/kubernetes-el7-x86\_64

> enabled=1

> gpgcheck=1

> repo\_gpgcheck=1

> gpgkey=https://packages.cloud.google.com/yum/doc/yum-key.gpg

>         https://packages.cloud.google.com/yum/doc/rpm-package-key.gpg

> EOF

#### Step 3: Install kubeadm and docker package on both nodes

[root@worker-node1 ~]# yum  install kubeadm docker -y

[root@worker-node2 ~]# yum  install kubeadm docker -y

Start and enable docker service

[root@worker-node1 ~]# systemctl restart docker && systemctl enable docker

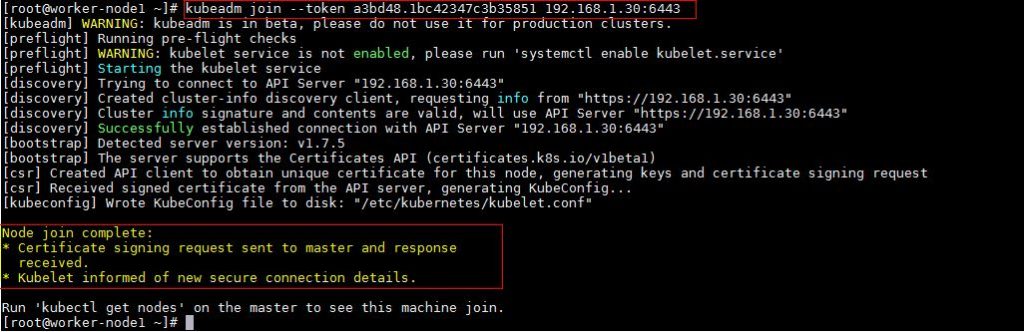
[root@worker-node2 ~]# systemctl restart docker && systemctl enable docker

#### Step 4: Now Join worker nodes to master node

To join worker nodes to Master node, a token is required. Whenever kubernetes master initialized , then in the output we get command and token.  Copy that command and run on both nodes.

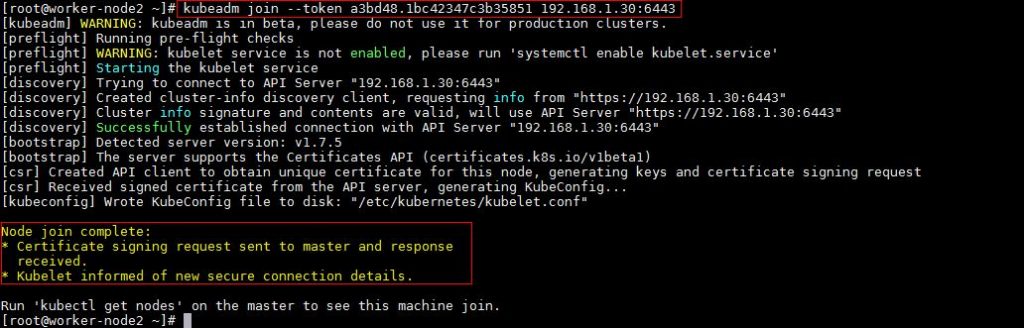
[root@worker-node1 ~]# kubeadm join --token a3bd48.1bc42347c3b35851 192.168.1.30:6443

Output of above command would be something like below

[](https://www.linuxtechi.com/wp-content/uploads/2017/09/kubeadm-node1.jpg)

[root@worker-node2 ~]# kubeadm join --token a3bd48.1bc42347c3b35851 192.168.1.30:6443

Output would be something like below

[](https://www.linuxtechi.com/wp-content/uploads/2017/09/kubeadm-join-node2.jpg)

Now verify Nodes status from master node using kubectl command

[root@k8s-master ~]# kubectl get nodes

NAME           STATUS    AGE       VERSION

k8s-master     Ready     2h        v1.7.5

worker-node1   Ready     20m       v1.7.5

worker-node2   Ready     18m       v1.7.5

[root@k8s-master ~]#

As we can see master and worker nodes are in ready status. This concludes that kubernetes 1.7 has been installed successfully and also we have successfully joined two worker nodes.  Now we can create pods and services.