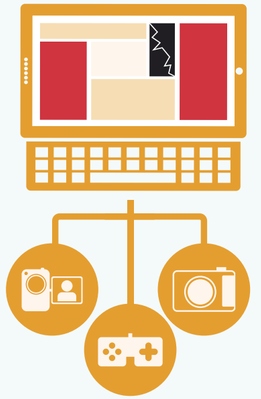
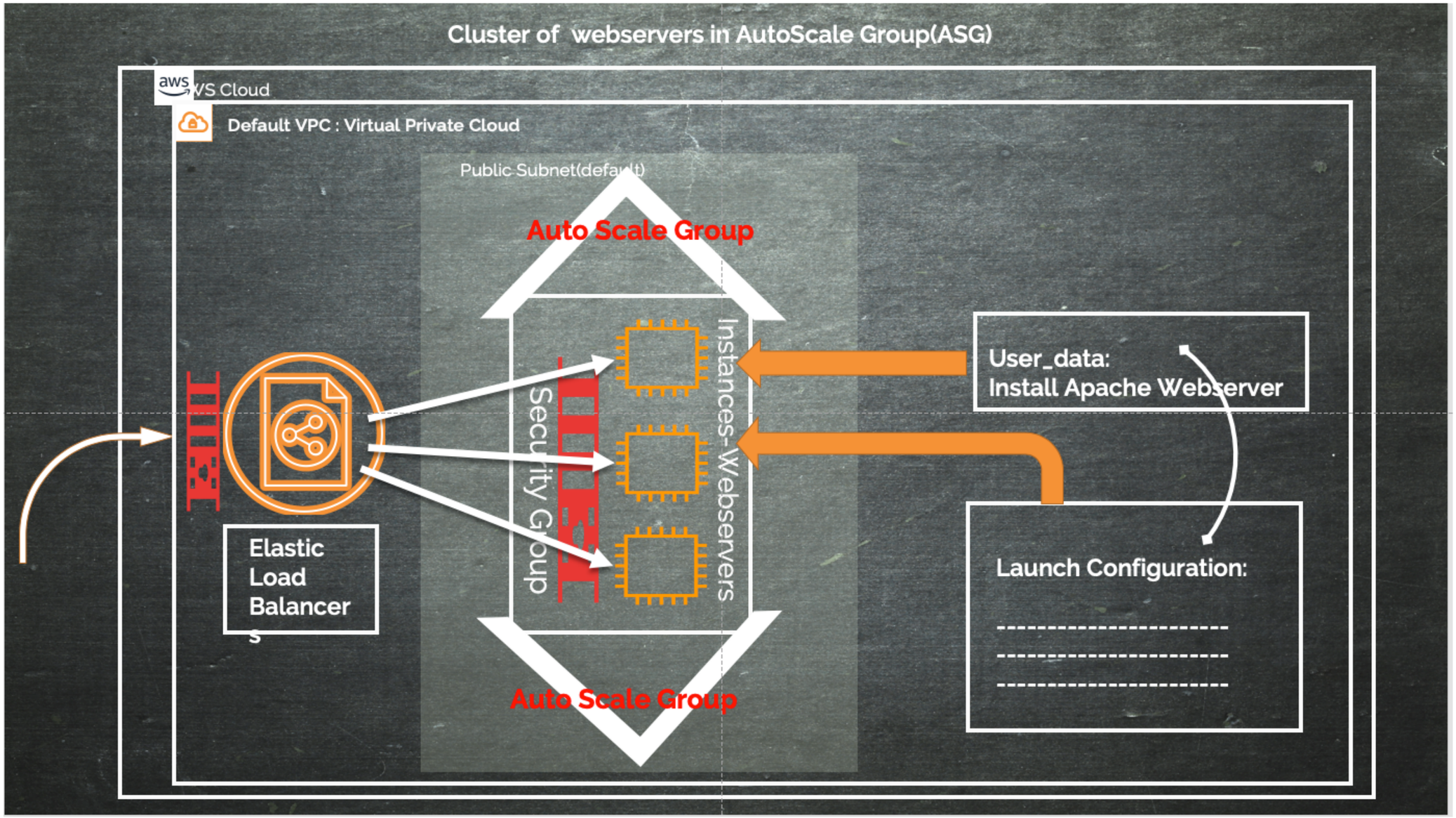
Deploy cluster of web servers in Auto Scaling Group with Elastic Load Balancer

Topics covered:

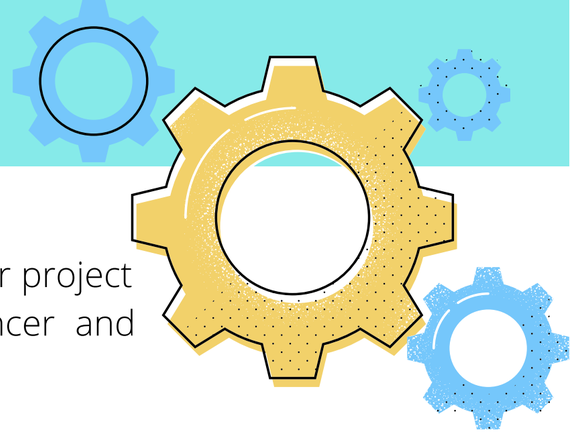
- Discuss Architecture of cluster of web servers in Auto Scaling Group with Elastic Load Balancer
- Update the last code for ASG and demonstrate to create ASG with ELB cluster
- Declare output variable and interpolate the attribute value of dns_name of ELB , once resource is created



Architecture : cluster of web servers in Auto Scaling Group with Elastic Load Balancers



Elastic Load Balancer



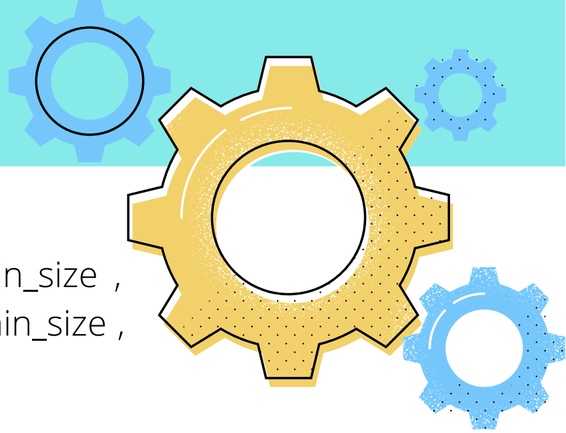
Elastic Load Balancer allow to balance the load across the nodes ASG cluster.ELB also helps to manage SSL cert if your project require https access. There are three types of ELB , AWS offers(classic Load Balancer , Network Load Balancer and Application Load Balancer) . We are creating Classic Load Balancer for this course.

Listener and health check resources are key to ELB. Notice the declared resources below. Security group rules allow access to ELB.

```
resource "aws_elb" "my_first_elb" {
    name = "terraform-elb"
    availability_zones = var.azs
    security_groups=[ aws_security_group.elb_sg.id ]
    listener {
        lb_port=80
        lb_protocol ="http"
        instance_port = var.server_port
        instance_protocol= "http"
    }
    health_check {
        healthy_threshold = 2
        unhealthy_threshold = 2
        timeout=3
        interval = 30
        target = "HTTP:${var.server_port}/"
    }
}
```

Note: propagate_at_launch is argument which allows resource update(in this case ,tag) while launching.

Auto Scaling Group



Auto Scaling Group allow us to scaling up and scaling down the resources based of usage. Notice the min_size , max_size and desired capacity value in arguments. You can create 1 instance in ASG assigning 1 to value for min_size , max_size and desired_capacity

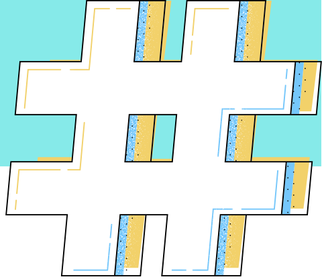
```
resource "aws_autoscaling_group" "my_first_asg" {
launch_configuration = aws_launch_configuration.my-first-launch-conf.id
availability_zones = var.azs

min_size = 2
max_size = 10
desired_capacity = 3

tag {
key = "Name"
value = "terraform-asg"
propagate_at_launch = true
}
}
```

Note: propagate_at_launch is argument which allows resource update(in this case ,tag) while launching.

Auto Scaling Policy



key feature of Auto Scaling Group is to scale up or scale down resources based on Auto Scaling Policy we attach.Notice policy_type and predefined_metric_type in below auto scaling policy resource.This policy will monitor the cpu usage and calculate number of ec2 instance to spin up to scale up ec2 instances if average cpu goes higher that target_value=60. And scale down the instances if usage below the average target_value=60. Awesome!

```
resource "aws_autoscaling_policy" "my_asg_policy" {
  name = "webservers_autoscale_policy"
  policy_type = "TargetTrackingScaling"
  autoscaling_group_name = aws_autoscaling_group.my_first_asg.name

  target_tracking_configuration {
  predefined_metric_specification {
    predefined_metric_type = "ASGAverageCPUUtilization"
  }
  target_value = "60"
  }
}
```

Output the resource attributes when its created

Declare a output variable with any meaningful name (you can specify any name, example "elb_endpoint") and interpolate the attribute you want as output.

```
output "elb_endpoint" {
    value = [ "${aws_elb.my_first_elb.dns_name}"]
}
```

How can we get the public IP assigned to our lab system

Run below command to get public ip assigned to your lab system..

$ curl https://checkip.amazonaws.com
73.241.51.131
$

OR

Search in google.com "what is my ip "

OR

Go to https://www.ipchicken.com/ and note current IP address