

Screenshot #1: VPC Creation

Description:

This screenshot shows the newly created VPC in the AWS console. These details confirm that the VPC was created according to the assignment requirements.

```
zhangxijing — ubuntu@ip-172-31-90-71: ~/assignment8 — ssh -i ubuntu-key.pem ubuntu@54.210.100.33 — 128x55
+ owner_id = (known after apply)
+ private_dns_hostname_type_on_launch = (known after apply)
+ tags = {
  + "Name" = "public-subnet-3"
}
+ tags_all = {
  + "Name" = "public-subnet-3"
}
+ vpc_id = (known after apply)

# aws_vpc.webapp_vpc will be created
resource "aws_vpc" "webapp_vpc" {
  + arn = (known after apply)
  + cidr_block = "10.0.0.0/16"
  + default_network_acl_id = (known after apply)
  + default_route_table_id = (known after apply)
  + default_security_group_id = (known after apply)
  + dhcp_options_id = (known after apply)
  + enable_dns_hostnames = (known after apply)
  + enable_dns_support = true
  + enable_network_address_usage_metrics = (known after apply)
  + id = (known after apply)
  + instance_tenancy = "default"
  + ipv6_association_id = (known after apply)
  + ipv6_cidr_block = (known after apply)
  + ipv6_cidr_block_network_border_group = (known after apply)
  + main_route_table_id = (known after apply)
  + owner_id = (known after apply)
  + tags = {
    + "Name" = "webapp-vpc"
  }
  + tags_all = {
    + "Name" = "webapp-vpc"
  }
}

Plan: 14 to add, 0 to change, 0 to destroy.

Do you want to perform these actions?
Terraform will perform the actions described above.
Only 'yes' will be accepted to approve.

Enter a value: yes

aws_vpc.webapp_vpc: Creating...
aws_vpc.webapp_vpc: Creation complete after 1s [id=vpc-00a72bf213f3f14a5]
aws_internet_gateway.webapp_igw: Creating...
aws_subnet.public_1: Creating...
aws_subnet.private_1: Creating...
aws_subnet.private_2: Creating...
aws_subnet.public_2: Creating...
aws_subnet.public_3: Creating...
aws_route_table.private_rt: Creating...
aws_route_table.private_rt: Creation complete after 1s [id=rtb-031001b708ce16541]
```



```
zhangxijing — ubuntu@ip-172-31-90-71: ~/assignment8 — ssh -i ubuntu-key.pem ubuntu@54.210.100.33 — 128x55

}

# aws_subnet.public_1 will be created
+ resource "aws_subnet" "public_1" {
  + arn
  + assign_ipv6_address_on_creation
  + availability_zone
  + availability_zone_id
  + cidr_block
  + enable_dns64
  + enable_resource_name_dns_a_record_on_launch
  + enable_resource_name_dns_aaaa_record_on_launch
  + id
  + ipv6_cidr_block_association_id
  + ipv6_native
  + map_public_ip_on_launch
  + owner_id
  + private_dns_hostname_type_on_launch
  + tags
    + "Name" = "public-subnet-1"
  }
  + tags_all
    + "Name" = "public-subnet-1"
  }
  + vpc_id
}

# aws_subnet.public_2 will be created
+ resource "aws_subnet" "public_2" {
  + arn
  + assign_ipv6_address_on_creation
  + availability_zone
  + availability_zone_id
  + cidr_block
  + enable_dns64
  + enable_resource_name_dns_a_record_on_launch
  + enable_resource_name_dns_aaaa_record_on_launch
  + id
  + ipv6_cidr_block_association_id
  + ipv6_native
  + map_public_ip_on_launch
  + owner_id
  + private_dns_hostname_type_on_launch
  + tags
    + "Name" = "public-subnet-2"
  }
  + tags_all
    + "Name" = "public-subnet-2"
  }
  + vpc_id
}

# aws_subnet.public_3 will be created
+ resource "aws_subnet" "public_3" {
  + arn
  + assign_ipv6_address_on_creation
  + availability_zone
  + availability_zone_id
  + cidr_block
  + enable_dns64
  + enable_resource_name_dns_a_record_on_launch
  + enable_resource_name_dns_aaaa_record_on_launch
  + id
  + ipv6_cidr_block_association_id
  + ipv6_native
  + map_public_ip_on_launch
  + owner_id
  + private_dns_hostname_type_on_launch
  + tags
    + "Name" = "public-subnet-3"
  }
  + tags_all
    + "Name" = "public-subnet-3"
  }
  + vpc_id
}
```

```
zhangxijing — ubuntu@ip-172-31-90-71: ~/assignment8 — ssh -i ubuntu-key.pem ubuntu@54.210.100.33 — 128x55
+ "Name" = "public-subnet-1"
}
+ vpc_id
= (known after apply)
}

# aws_subnet.public_2 will be created
+ resource "aws_subnet" "public_2" {
+ arn
+ assign_ipv6_address_on_creation
+ availability_zone
+ availability_zone_id
+ cidr_block
+ enable_dns64
+ enable_resource_name_dns_a_record_on_launch
+ enable_resource_name_dns_aaaa_record_on_launch
+ id
+ ipv6_cidr_block_association_id
+ ipv6_native
+ map_public_ip_on_launch
+ owner_id
+ private_dns_hostname_type_on_launch
+ tags
+ "Name" = "public-subnet-2"
}
+ tags_all
+ "Name" = "public-subnet-2"
}
+ vpc_id
= (known after apply)

}

# aws_subnet.public_3 will be created
+ resource "aws_subnet" "public_3" {
+ arn
+ assign_ipv6_address_on_creation
+ availability_zone
+ availability_zone_id
+ cidr_block
+ enable_dns64
+ enable_resource_name_dns_a_record_on_launch
+ enable_resource_name_dns_aaaa_record_on_launch
+ id
+ ipv6_cidr_block_association_id
+ ipv6_native
+ map_public_ip_on_launch
+ owner_id
+ private_dns_hostname_type_on_launch
+ tags
+ "Name" = "public-subnet-3"
}
+ tags_all
+ "Name" = "public-subnet-3"
}
+ vpc_id
= (known after apply)
```

Screenshot #3: Internet Gateway and Route Tables

Description:

This screenshot displays:

- The Internet Gateway (webapp-igw) attached to the VPC
- The public route table with a 0.0.0.0/0 route pointing to the IGW
- The private route table without a direct route to the internet
- The subnet associations for public and private route tables

```
zhangxijing — ubuntu@ip-172-31-90-71: ~/assignment8 — ssh -i ubuntu-key.pem ubuntu@54.210.100.33 — 128x55
+ ipv6_cidr_block          = (known after apply)
+ ipv6_cidr_block_network_border_group = (known after apply)
+ main_route_table_id      = (known after apply)
+ owner_id                 = (known after apply)
+ tags                     = {
    + "Name" = "webapp-vpc"
}
+ tags_all                 = {
    + "Name" = "webapp-vpc"
}
}

Plan: 14 to add, 0 to change, 0 to destroy.

Do you want to perform these actions?
Terraform will perform the actions described above.
Only 'yes' will be accepted to approve.

Enter a value: yes

aws_vpc.webapp_vpc: Creating...
aws_vpc.webapp_vpc: Creation complete after 1s [id=vpc-00a72bf213f3f14a5]
aws_internet_gateway.webapp_igw: Creating...
aws_subnet.public_1: Creating...
aws_subnet.private_1: Creating...
aws_subnet.private_2: Creating...
aws_subnet.public_2: Creating...
aws_subnet.public_3: Creating...
aws_route_table.private_rt: Creating...
aws_route_table.private_rt: Creation complete after 1s [id=rtb-031001b708ce16541]
aws_internet_gateway.webapp_igw: Creation complete after 1s [id=igw-0f53998b2786f806c]
aws_route_table.public_rt: Creating...
aws_route_table.public_rt: Creation complete after 0s [id=rtb-0f7533af0d0cc67e4]
aws_subnet.private_1: Creation complete after 3s [id=subnet-0dc7c50d3caffbad2]
aws_route_table_association.private_1: Creating...
aws_subnet.private_2: Creation complete after 3s [id=subnet-06f64fdd5006d3401]
aws_route_table_association.private_2: Creating...
aws_route_table_association.private_1: Creation complete after 0s [id=rtbassoc-010de401dfbb49dc9]
aws_subnet.public_1: Still creating... [10s elapsed]
aws_subnet.public_3: Still creating... [10s elapsed]
aws_subnet.public_2: Still creating... [10s elapsed]
aws_subnet.public_2: Creation complete after 11s [id=subnet-005ca936dd01efb60]
aws_route_table_association.public_2: Creating...
aws_route_table_association.public_2: Creation complete after 0s [id=rtbassoc-0266ea237179190a2]
aws_subnet.public_1: Creation complete after 12s [id=subnet-09f26f8cf64f1ff83]
aws_route_table_association.public_1: Creating...
aws_route_table_association.public_1: Creation complete after 1s [id=rtbassoc-09bb49e90e5e724ef]
aws_route_table_association.private_2: Still creating... [10s elapsed]
aws_subnet.public_3: Creation complete after 13s [id=subnet-0b26385ee6ec64982]
aws_route_table_association.public_3: Creating...
aws_route_table_association.public_3: Creation complete after 0s [id=rtbassoc-09895c992a7cd8dd8]
aws_route_table_association.private_2: Creation complete after 13s [id=rtbassoc-0d1a21d6a3edc0a2c]

Apply complete! Resources: 14 added, 0 changed, 0 destroyed.
ubuntu@ip-172-31-90-71:~/assignment8$
```

Screenshot #4: ALB Security Group

Description:

Here you can see the inbound and outbound rules for the ALB's security group:

- Inbound: Allows HTTP (port 80) from anywhere (0.0.0.0/0)
- Outbound: Allows all traffic to 0.0.0.0/0

```
zhangxijing — ubuntu@ip-172-31-90-71: ~/assignment8 — ssh -i ubuntu-key.pem ubuntu@54.210.100.33 — 128x55

Terraform will perform the following actions:

# aws_security_group.alb_sg will be created
+ resource "aws_security_group" "alb_sg" {
    + arn                = (known after apply)
    + description        = "Allow HTTP from anywhere"
    + egress             = [
        + {
            + cidr_blocks      = [
                + "0.0.0.0/0",
            ]
            + description      = ""
            + from_port         = 0
            + ipv6_cidr_blocks = []
            + prefix_list_ids  = []
            + protocol          = "-1"
            + security_groups   = []
            + self               = false
            + to_port            = 0
        },
    ]
    + id                 = (known after apply)
    + ingress            = [
        + {
            + cidr_blocks      = [
                + "0.0.0.0/0",
            ]
            + description      = ""
            + from_port         = 80
            + ipv6_cidr_blocks = []
            + prefix_list_ids  = []
            + protocol          = "tcp"
            + security_groups   = []
            + self               = false
            + to_port            = 80
        },
    ]
    + name               = "alb-sg"
    + name_prefix        = (known after apply)
    + owner_id           = (known after apply)
    + revoke_rules_on_delete = false
    + tags               = {
        + "Name" = "alb-sg"
    }
    + tags_all           = {
        + "Name" = "alb-sg"
    }
    + vpc_id             = "vpc-00a72bf213f3f14a5"
}

# aws_security_group.ec2_sg will be created
+ resource "aws_security_group" "ec2_sg" {
    + arn                = (known after apply)
    + description        = "Allow HTTP (8080) from ALB SG"
```

Screenshot #5: EC2 Security Group

Description:

This screenshot shows:

- Inbound rule: Port 8080 allowed only from the ALB security group
- Outbound rule: All traffic allowed to 0.0.0.0/0

```
# zhangxijing — ubuntu@ip-172-31-90-71: ~/assignment8 — ssh -i ubuntu-key.pem ubuntu@54.210.100.33 — 128x55
}
+ vpc_id
      = "vpc-00a72bf213f3f14a5"

# aws_security_group.ec2_sg will be created
+ resource "aws_security_group" "ec2_sg" {
  + arn
    = (known after apply)
  + description
    = "Allow HTTP (8080) from ALB SG"
  + egress
    =
      +
        + cidr_blocks
          = [
            + "0.0.0.0/0",
          ]
        + description
          = ""
        + from_port
          = 0
        + ipv6_cidr_blocks
          = []
        + prefix_list_ids
          = []
        + protocol
          = "-1"
        + security_groups
          = []
        + self
          = false
        + to_port
          = 0
      ],
  + id
    = (known after apply)
  + ingress
    =
      +
        {
          + cidr_blocks
            = []
          + description
            = ""
          + from_port
            = 8080
          + ipv6_cidr_blocks
            = []
          + prefix_list_ids
            = []
          + protocol
            = "tcp"
          + security_groups
            = (known after apply)
          + self
            = false
          + to_port
            = 8080
        },
  + name
    = "ec2-sg"
  + name_prefix
    = (known after apply)
  + owner_id
    = (known after apply)
  + revoke_rules_on_delete
    = false
  + tags
    =
      +
        "Name" = "ec2-sg"
    }
  + tags_all
    =
      +
        "Name" = "ec2-sg"
    }
  + vpc_id
    = "vpc-00a72bf213f3f14a5"
}

# aws_security_group.rds_sg will be created
+ resource "aws_security_group" "rds_sg" {
  + arn
    = (known after apply)
  + description
    = "Allow DB access from EC2 SG"
  + egress
    = [
```

Screenshot #6: RDS Security Group

Description:

This screenshot highlights:

- Inbound rule: Database port (5432 for PostgreSQL or 3306 for MySQL) allowed from the EC2 security group
- Outbound rule: All traffic allowed
- The association with the correct VPC

```
zhangxijing - ubuntu@ip-172-31-90-71: ~/assignment8 - ssh -i ubuntu-key.pem ubuntu@54.210.100.33 - 128x55
+ vpc_id          = "vpc-00a72bf213f3f14a5"
}

# aws_security_group.rds_sg will be created
+ resource "aws_security_group" "rds_sg" {
  + arn           = (known after apply)
  + description   = "Allow DB access from EC2 SG"
  + egress        = [
    + {
      + cidr_blocks  = [
        + "0.0.0.0/0",
      ]
      + description   = ""
      + from_port     = 0
      + ipv6_cidr_blocks = []
      + prefix_list_ids = []
      + protocol      = "-1"
      + security_groups = []
      + self          = false
      + to_port       = 0
    },
  ]
  + id            = (known after apply)
  + ingress       = [
    + {
      + cidr_blocks  = []
      + description   = ""
      + from_port     = 3306
      + ipv6_cidr_blocks = []
      + prefix_list_ids = []
      + protocol      = "tcp"
      + security_groups = (known after apply)
      + self          = false
      + to_port       = 3306
    },
  ]
  + name          = "rds-sg"
  + name_prefix   = (known after apply)
  + owner_id      = (known after apply)
  + revoke_rules_on_delete = false
  + tags          = [
    + "Name" = "rds-sg"
  ]
  + tags_all      = [
    + "Name" = "rds-sg"
  ]
  + vpc_id        = "vpc-00a72bf213f3f14a5"
}

Plan: 3 to add, 0 to change, 0 to destroy.

Do you want to perform these actions?
Terraform will perform the actions described above.
Only 'yes' will be accepted to approve.
```

Screenshot #7: RDS Instance Configuration

Description:

Displays the details of the RDS instance:

- Engine (MySQL)
- Instance class (e.g., db.t3.micro)
- DB name (webapp_db)
- Security group and subnet group references
- Public access disabled

The screenshot shows the AWS RDS instance configuration page for a database named 'webapp-db'. The left sidebar lists various RDS management options like Dashboard, Query Editor, and Reserved Instances. The main content area displays the 'Summary' and 'Configuration' tabs. Under 'Summary', the DB identifier is 'webapp-db', the status is 'Creating', the engine is MySQL Community, and the region is us-east-1b. Under 'Configuration', detailed settings include the DB instance ID (webapp-db), instance class (db.t3.micro), storage type (General Purpose SSD (gp2)), and monitoring type (Database Insights - Standard). Other visible parameters include engine version (8.0.40), RAM (1 GB), and IAM DB authentication (Not enabled).

Screenshot #8: Launch Template Configuration

Description:

Shows the Launch Template used to create the EC2 instances, including:

- AMI ID (ami-03a6c16a66bbe0a7a)
- Instance type (e.g., t3.micro)
- User data script (to create /usr/bin/csye6225/.env, set permissions, and start webapp.service)

The screenshot shows the AWS EC2 console with the "Launch templates" section selected. A specific launch template, "webapp-launch-template-20250331052144213900000001 (lt-0aef2d989d300d34e)", is displayed. The "Details" tab is active, showing the following configuration details:

Launch template ID	Launch template name	Default version	Owner
lt-0aef2d989d300d34e	webapp-launch-template-20250331052144213900000001	1	arn:aws:sts::132030096872:assume d-role/LabRole/- Ob001a039f6c89acd

Below the main details, the "Launch template version details" section is shown, with the "1 (Default)" version selected. It displays the following information:

Version	Description	Date created	Created by
1 (Default)	-	2025-03-31T05:21:44.000Z	arn:aws:sts::132030096872:assume d-role/LabRole/- Ob001a039f6c89acd

The "Instance details" tab is selected under the version details, showing the following instance configuration:

AMI ID	Instance type	Availability Zone	Key pair name
ami-03a6c16a66bbe0a7a	t3.micro	-	-

Under "Security groups", it lists "sg-0dc0a2e44496b8ee2".

Screenshot #9: Target Group and Health Check

Description:

Illustrates:

- Target group name (webapp-target-group)
- Protocol (HTTP) and Port (8080)
- Health check path (/healthz)
- Matcher (200 OK)

The screenshot shows the AWS EC2 Target Groups console. On the left, there is a navigation sidebar with links like 'Spot Requests', 'Savings Plans', 'Reserved Instances', 'Dedicated Hosts', 'Capacity Reservations', 'Images', 'Elastic Block Store', 'Network & Security', 'Load Balancing', and 'Auto Scaling'. The main content area displays the details of a target group named 'webapp-target-group'. The 'Details' section shows the ARN: 'arn:aws:elasticloadbalancing:us-east-1:132030096872:targetgroup/webapp-target-group/24cf69235ec64ac7'. It specifies the 'Target type' as 'Instance', 'Protocol : Port' as 'HTTP: 8080', 'Protocol version' as 'HTTP1', and the 'VPC' as 'vpc-076884b7b045e6b9c'. Below this, a table shows the target status: 0 Total targets, 0 Healthy, 0 Unhealthy, 0 Unused, 0 Initial, and 0 Draining. Under the 'Targets' tab, it says 'Registered targets (0)' and 'No registered targets'. A note states: 'Target groups route requests to individual registered targets using the protocol and port number specified. Health checks are performed on all registered targets according to the target group's health check settings. Anomaly detection is automatically applied to HTTP/HTTPS target groups with at least 3 healthy targets.' There is a 'Register targets' button.

Screenshot #10: Auto Scaling Group Settings

Description:

- The ASG name (webapp-asg)
- Minimum size (1), maximum size (3), and desired capacity (1)
- VPC subnets (public subnets)
- The Launch Template association
- The Load Balancer or Target Group attachment

The screenshot shows the AWS EC2 Auto Scaling groups page. On the left, there's a navigation sidebar with sections like EC2, Instances, Images, Elastic Block Store, Network & Security, and Load Balancing. The main content area is titled "Auto Scaling groups (1/1) Info". It shows a table with one row for "webapp-asg". The table columns include Name, Launch template/configuration, Instances, Status, and Desired capacity. The "Desired capacity" column shows a value of 1. Below this, there's a detailed view for the "webapp-asg Capacity overview". It shows "Desired capacity" as 1, "Scaling limits (Min - Max)" as 1 - 3, "Desired capacity type" as Units (number of instances), and "Status" as -. At the bottom of the page, there are links for CloudShell and Feedback, and a footer with copyright information.

Screenshot #11: CloudWatch Alarms for Scaling Policies

Description:

Displays the two alarms:

1. Scale Up: CPU > 70%
2. Scale Down: CPU < 30%

The screenshot shows the AWS CloudWatch Alarms interface. On the left, there's a navigation sidebar with links like Dashboards, AI Operations, Alarms, Logs, Metrics, X-Ray traces, Events, Application Signals, Network Monitoring, and Insights. The main area is titled 'Alarms (2)' and lists two alarms:

Name	State	Last state update (UTC)	Conditions
low-cpu-alarm	In alarm	2025-03-31 05:36:00	CPUUtilization < 30 for 2 datapoints within 4 minutes
high-cpu-alarm	OK	2025-03-31 05:35:19	CPUUtilization > 70 for 2 datapoints within 4 minutes

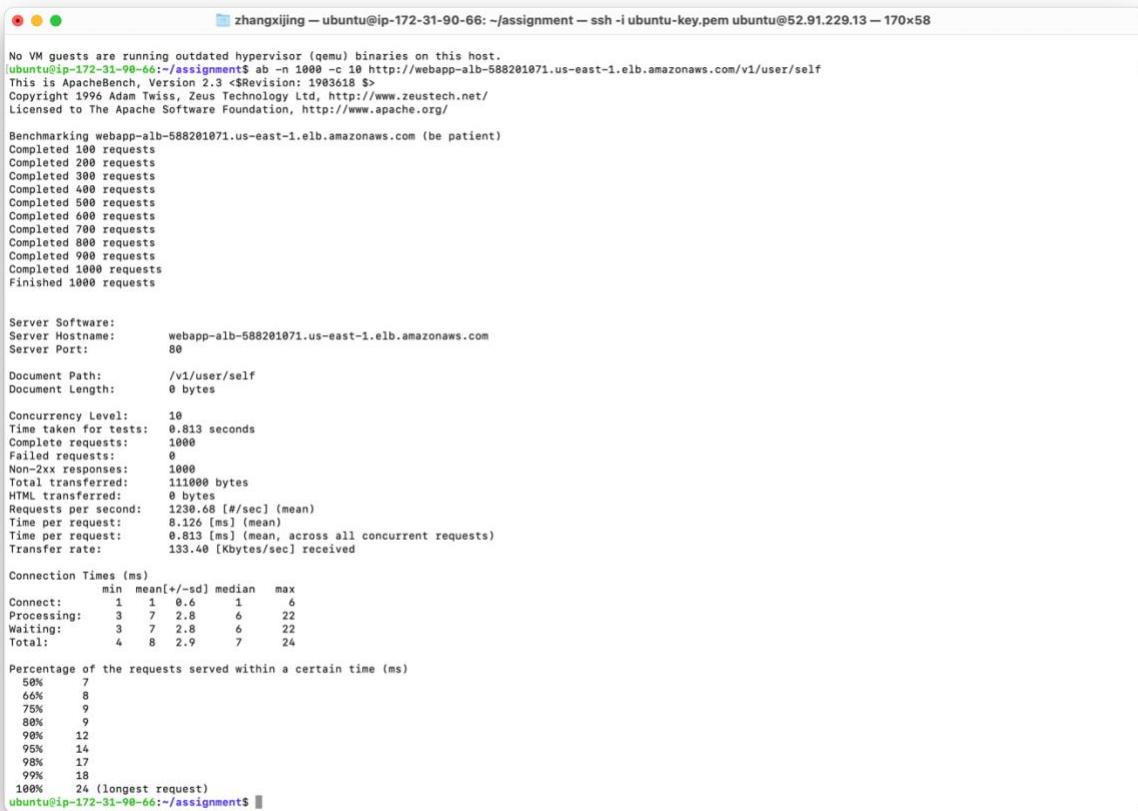
Screenshot #12: ASG Scaling Up/Down During Load Test

Description:

Demonstrates how the ASG responds to high CPU usage by:

- Adding new instances (scale up) when the CPU crosses the threshold
- Removing instances (scale down) once the load decreases

Show the AWS console's "Activity" or "Instances" view in the ASG during load testing, proving dynamic scaling.



The screenshot shows a terminal window titled "zhangxijing — ubuntu@ip-172-31-90-66: ~/assignment — ssh -i ubuntu-key.pem ubuntu@52.91.229.13 — 170x58". The terminal displays the output of an ApacheBench load test. The test was run with 10 threads (ab -n 1000 -c 10) against a service at port 80 (<http://webapp-alb-588201071.us-east-1.elb.amazonaws.com/v1/user/self>). The test completed 1000 requests. The output includes performance metrics like requests per second, connection times, and request distribution percentages.

```
No VM guests are running outdated hypervisor (qemu) binaries on this host.  
ubuntu@ip-172-31-90-66:~/assignment$ ab -n 1000 -c 10 http://webapp-alb-588201071.us-east-1.elb.amazonaws.com/v1/user/self  
This is ApacheBench, Version 2.3 <Revision: 1903618 $>  
Copyright 1996 Adam Twiss, Zeus Technology Ltd, http://www.zeustech.net/  
Licensed to The Apache Software Foundation, http://www.apache.org/  
  
Benchmarking webapp-alb-588201071.us-east-1.elb.amazonaws.com (be patient)  
Completed 100 requests  
Completed 200 requests  
Completed 300 requests  
Completed 400 requests  
Completed 500 requests  
Completed 600 requests  
Completed 700 requests  
Completed 800 requests  
Completed 900 requests  
Completed 1000 requests  
Finished 1000 requests  
  
Server Software:        webapp-alb-588201071.us-east-1.elb.amazonaws.com  
Server Hostname:       webapp-alb-588201071.us-east-1.elb.amazonaws.com  
Server Port:          80  
  
Document Path:         /v1/user/self  
Document Length:      0 bytes  
  
Concurrency Level:    10  
Time taken for tests: 0.813 seconds  
Complete requests:   1000  
Failed requests:     0  
Non-2xx responses:  1000  
Total transferred:  111000 bytes  
HTML transferred:   0 bytes  
Requests per second: 1230.68 (#/sec) (mean)  
Time per request:   8.126 [ms] (mean)  
Time per request:   0.813 [ms] (mean, across all concurrent requests)  
Transfer rate:       133.48 [Kbytes/sec] received  
  
Connection Times (ms)  
              min  mean[+/-sd] median   max  
Connect:       1    1  0.6      1    6  
Processing:    3    7  2.8      6   22  
Waiting:      3    7  2.8      6   22  
Total:        4    8  2.9      7   24  
  
Percentage of the requests served within a certain time (ms)  
  50%    7  
  66%    8  
  75%    9  
  80%    9  
  90%   12  
  95%   14  
  98%   17  
  99%   18  
 100%  24 (longest request)  
ubuntu@ip-172-31-90-66:~/assignment$
```

Google Translate | Google Translate | webapp- | Sandbox | Workbenches | Instances | webapp- | Google Translate | Terraform | Relaunch to update

us-east-1.console.aws.amazon.com/ec2/home?region=us-east-1#Instances:instanceState=running

aws Search [Option+S] United States (N. Virginia) v vocabs/user3813293=Zhang_XJing @ 1952-6085-6908 v

EC2 Instances

Dashboard EC2 Global View Events

Instances (3) Info Last updated less than a minute ago Connect Instance state Actions Launch instances

Find instance by attribute or tag (case-sensitive)

All states

Instance state = running Clear filters

	Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Ava
<input type="checkbox"/>	Bastion Host	i-0b9ece92c761027c9	Running	t2.micro	2/2 checks passed	View alarms +	us-e
<input type="checkbox"/>	zhang-server	i-026e2cbb0b0a0c2b1	Running	t2.micro	2/2 checks passed	View alarms +	us-e
<input type="checkbox"/>	webapp-instance	i-0398b9a51f4f9d166	Running	t3.micro	3/3 checks passed	View alarms +	us-e

Select an instance

CloudShell Feedback © 2025, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

Google Translate | Google Translate | webapp- | Sandbox | Workbenches | Instances | webapp- | Google Translate | Terraform | Relaunch to update

us-east-1.console.aws.amazon.com/ec2/home?region=us-east-1#Instances:

Search [Option+S] United States (N. Virginia) v vocabs/user3813293=Zhang_XJing @ 1952-6085-6908 v

EC2 Instances

EC2 Dashboard EC2 Global View Events Instances Instances Instance Types Launch Templates Spot Requests Savings Plans Reserved Instances Dedicated Hosts Capacity Reservations Images AMIs AMI Catalog Elastic Block Store Volumes Snapshots Lifecycle Manager Network & Security Security Groups Elastic IPs Placement Groups Key Pairs Network Interfaces Load Balancing CloudShell Feedback

Last updated less than a minute ago

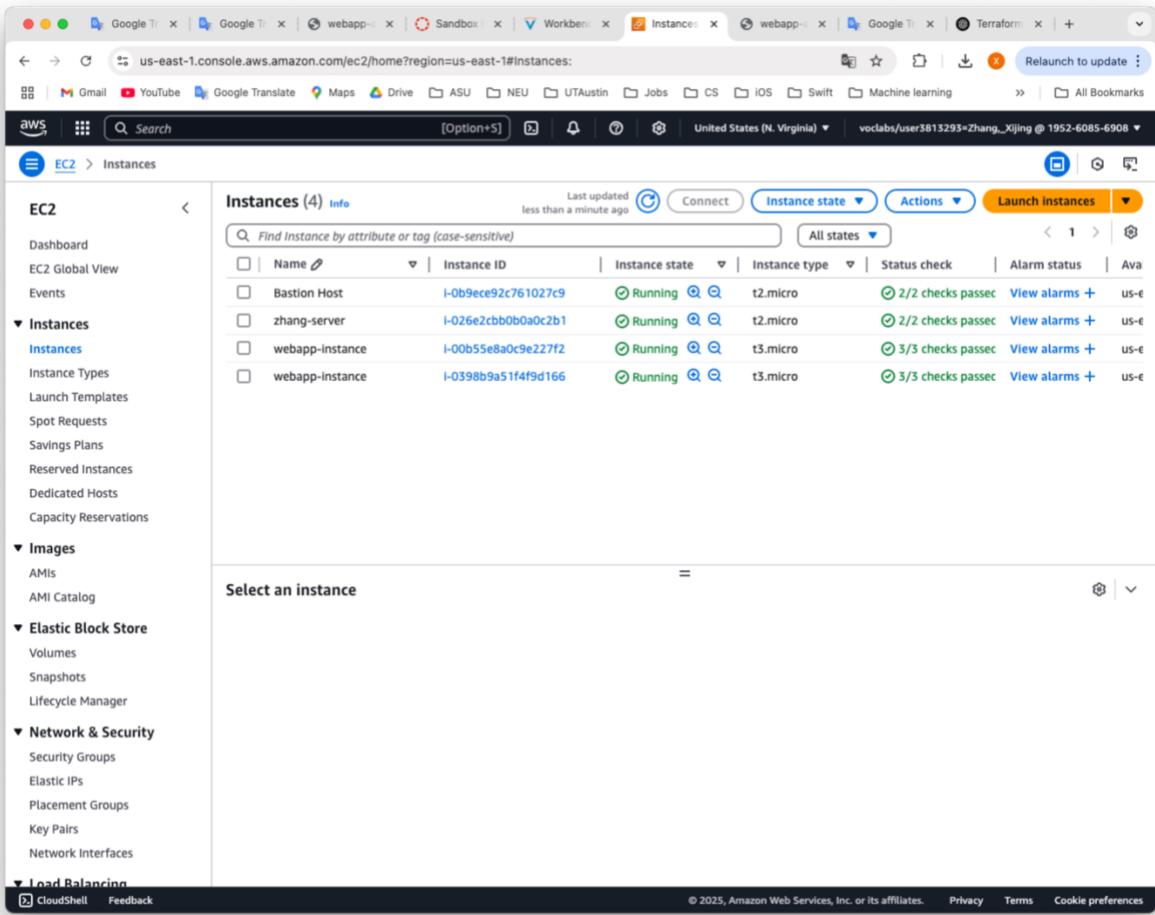
Find Instance by attribute or tag (case-sensitive)

Instance state Actions Launch instances

Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Ava
Bastion Host	I-0b9ece92c761027c9	Running	t2.micro	2/2 checks passed	View alarms +	us-e
zhang-server	I-026e2ccb0b0a0c2b1	Running	t2.micro	2/2 checks passed	View alarms +	us-e
webapp-instance	I-00b55e8a0c9e227f2	Running	t3.micro	3/3 checks passed	View alarms +	us-e
webapp-instance	I-0398b9a51f4f9d166	Running	t3.micro	3/3 checks passed	View alarms +	us-e

Select an instance

© 2025, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences



Screenshot #13: API Requests and Responses

Description:

Includes:

1. Create User (POST /v1/user) request and response (e.g., 201 Created).
2. Get User (Self) (GET /v1/user/self with BasicAuth) request and response (e.g., 200 OK).



```
zhangxijing — ubuntu@ip-172-31-93-79: ~ — ssh -i ubuntu-key.pem ubuntu@35.171.19.177 — 149x58
[ubuntu@ip-172-31-93-79:~]$ curl -i http://webapp-alb-1568480669.us-east-1.elb.amazonaws.com/healthz
HTTP/1.1 200 OK
Date: Tue, 01 Apr 2025 01:52:37 GMT
Content-Length: 0
Connection: keep-alive
Cache-Control: no-cache

ubuntu@ip-172-31-93-79:~$
```

```
zhangxijing — ubuntu@ip-172-31-93-79: ~ — ssh -i ubuntu-key.pem ubuntu@35.171.19.177 — 149x58
[ubuntu@ip-172-31-93-79:~]$ curl -i http://webapp-alb-1568480669.us-east-1.elb.amazonaws.com/healthz
HTTP/1.1 200 OK
Date: Tue, 01 Apr 2025 01:52:37 GMT
Content-Length: 0
Connection: keep-alive
Cache-Control: no-cache

[ubuntu@ip-172-31-93-79:~]$ curl -i -X POST http://webapp-alb-1568480669.us-east-1.elb.amazonaws.com/v1/user \
-H "Content-Type: application/json" \
-d '{
    "email": "john.doe@example.com",
    "password": "MyPassword123!",
    "first_name": "John",
    "last_name": "Doe"
}'

HTTP/1.1 201 Created
Date: Tue, 01 Apr 2025 01:53:05 GMT
Content-Type: application/json
Content-Length: 211
Connection: keep-alive

{"ID":"0423e154-ce73-4764-8354-d55f4d36c709","email":"john.doe@example.com","first_name":"John","last_name":"Doe","account_created":"2025-04-01T01:53:05.477839684Z","updated_at":"2025-04-01T01:53:05.477839684Z"}[ubuntu@ip-172-31-93-79:~]$
```

```
zhangxijing — ubuntu@ip-172-31-93-79: ~ — ssh -i ubuntu-key.pem ubuntu@35.171.19.177 — 149x58
[ubuntu@ip-172-31-93-79:~]$ curl -i http://webapp-alb-1568480669.us-east-1.elb.amazonaws.com/healthz
HTTP/1.1 200 OK
Date: Tue, 01 Apr 2025 01:52:37 GMT
Content-Length: 0
Connection: keep-alive
Cache-Control: no-cache

[ubuntu@ip-172-31-93-79:~]$ curl -i -X POST http://webapp-alb-1568480669.us-east-1.elb.amazonaws.com/v1/user \
-H "Content-Type: application/json" \
-d '{
  "email": "john.doe@example.com",
  "password": "MyPassword123!",
  "first_name": "John",
  "last_name": "Doe"
}'
HTTP/1.1 201 Created
Date: Tue, 01 Apr 2025 01:53:05 GMT
Content-Type: application/json
Content-Length: 211
Connection: keep-alive

{"ID": "0423e154-ce73-4764-8354-d55f4d36c709", "email": "john.doe@example.com", "first_name": "John", "last_name": "Doe", "account_created": "2025-04-01T01:53:05.477839684Z", "updated_at": "2025-04-01T01:53:05.477839684Z"}[ubuntu@ip-172-31-93-79:~]$ curl -i \
-u "john.doe@example.com:MyPassword123!" \
http://webapp-alb-1568480669.us-east-1.elb.amazonaws.com/v1/user/self
HTTP/1.1 200 OK
Date: Tue, 01 Apr 2025 01:53:23 GMT
Content-Type: application/json
Content-Length: 205
Connection: keep-alive

{"ID": "0423e154-ce73-4764-8354-d55f4d36c709", "email": "john.doe@example.com", "first_name": "John", "last_name": "Doe", "account_created": "2025-04-01T01:53:05.4778392Z", "updated_at": "2025-04-01T01:53:05.4778392Z"}[ubuntu@ip-172-31-93-79:~]$
```

```
zhangxijing — ubuntu@ip-172-31-93-79: ~ — ssh -i ubuntu-key.pem ubuntu@35.171.19.177 — 149x58
Date: Tue, 01 Apr 2025 01:52:37 GMT
Content-Length: 0
Connection: keep-alive
Cache-Control: no-cache

ubuntu@ip-172-31-93-79:~$ curl -i -X POST http://webapp-alb-1568480669.us-east-1.elb.amazonaws.com/v1/user \
-H "Content-Type: application/json" \
-d '{
    "email": "john.doe@example.com",
    "password": "MyPassword123!",
    "first_name": "John",
    "last_name": "Doe"
}'

HTTP/1.1 201 Created
Date: Tue, 01 Apr 2025 01:53:05 GMT
Content-Type: application/json
Content-Length: 211
Connection: keep-alive

{"ID":"0423e154-ce73-4764-8354-d55f4d36c709","email":"john.doe@example.com","first_name":"John","last_name":"Doe","account_created":"2025-04-01T01:53:05.477839684Z","updated_at":"2025-04-01T01:53:05.477839684Z"}ubuntu@ip-172-31-93-79:~$ curl -i \
-u "john.doe@example.com:MyPassword123!" \
http://webapp-alb-1568480669.us-east-1.elb.amazonaws.com/v1/user/self

HTTP/1.1 200 OK
Date: Tue, 01 Apr 2025 01:53:23 GMT
Content-Type: application/json
Content-Length: 205
Connection: keep-alive

{"ID":"0423e154-ce73-4764-8354-d55f4d36c709","email":"john.doe@example.com","first_name":"John","last_name":"Doe","account_created":"2025-04-01T01:53:05.477839684Z","updated_at":"2025-04-01T01:53:05.477839684Z"}ubuntu@ip-172-31-93-79:~$ curl -i -X POST http://webapp-alb-1568480669.us-east-1.elb.amazonaws.com/v1/user \
-H "Content-Type: application/json" \
-d '{
    "email": "xijing.zhang@example.com",
    "password": "MyPassword123!",
    "first_name": "Xijing",
    "last_name": "Zhang"
}'

HTTP/1.1 201 Created
Date: Tue, 01 Apr 2025 01:54:19 GMT
Content-Type: application/json
Content-Length: 219
Connection: keep-alive

{"ID":"d0c1636a-2ff0-4771-9813-55df83448157","email":"xijing.zhang@example.com","first_name":"Xijing","last_name":"Zhang","account_created":"2025-04-01T01:54:19.075280057Z","updated_at":"2025-04-01T01:54:19.075280057Z"}ubuntu@ip-172-31-93-79:~$ curl -i \
-u "xijing.zhang@example.com:MyPassword123!" \
http://webapp-alb-1568480669.us-east-1.elb.amazonaws.com/v1/user/self

HTTP/1.1 200 OK
Date: Tue, 01 Apr 2025 01:54:23 GMT
Content-Type: application/json
Content-Length: 211
Connection: keep-alive

{"ID":"d0c1636a-2ff0-4771-9813-55df83448157","email":"xijing.zhang@example.com","first_name":"Xijing","last_name":"Zhang","account_created":"2025-04-01T01:54:19.0752802Z","updated_at":"2025-04-01T01:54:19.0752802Z"}ubuntu@ip-172-31-93-79:~$
```