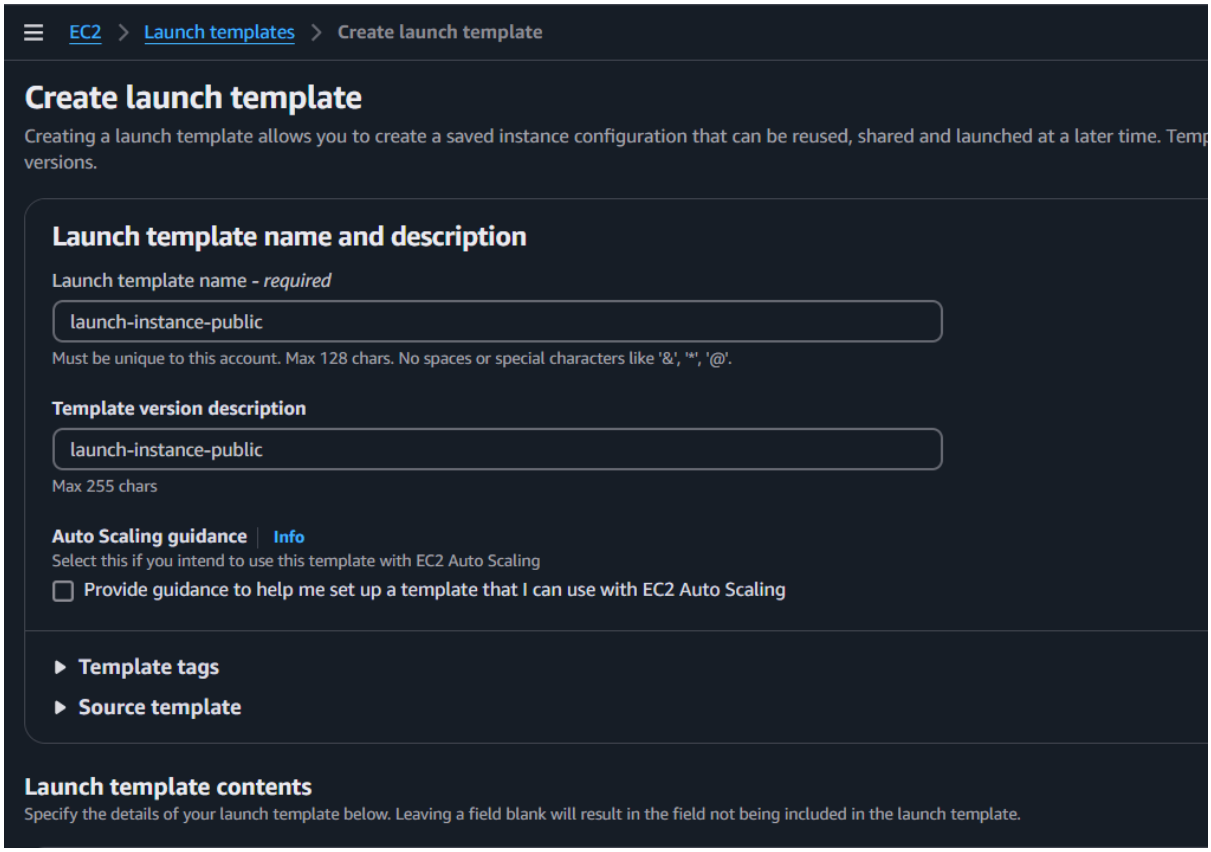
Autoscaling group

Step-1: create a VPC environment or use default. If your using own VPC. Create VPC, subnets, internet gateway, route table, NAT gateway. Create security groups, load balancer, target group.

Step-2: Create launch instance



Launch template contents

Specify the details of your launch template below. Leaving a field blank will result in the field not being included in the launch template.

Application and OS Images (Amazon Machine Image) Info

An AMI is a template that contains the software configuration (operating system, application server, and applications) required to launch your instance. Search or Browse for AMIs if you don't see what you are looking for below

Search our full catalog including 1000s of application and OS images

Recents | Quick Start

Don't include in launch template | Amazon Linux | macOS | Ubuntu | Windows | Red Hat | SUSE Linux | Debian

Browse more AMIs

Including AMIs from AWS, Marketplace and the Community

Amazon Machine Image (AMI)

Amazon Linux 2023 AMI — Free tier eligible

ami-0d2614eafc1b0e4d2 (64-bit (x86), uefi-preferred) / ami-0f6168e963366cd18 (64-bit (Arm), uefi)

Virtualization: hvm ENA enabled: true Root device type: ebs

Description

Amazon Linux 2023 is a modern, general purpose Linux-based OS that comes with 5 years of long term support. It is optimized for AWS and designed to provide a secure, stable and high-performance execution environment to develop and run your cloud applications.

Instance type Info | Get advice

Instance type

t2.micro — Free tier eligible

Family: t2 1 vCPU 1 GiB Memory Current generation: true On-Demand Linux base pricing: 0.0124 USD per Hour

On-Demand Windows base pricing: 0.017 USD per Hour On-Demand RHEL base pricing: 0.0268 USD per Hour

On-Demand Ubuntu Pro base pricing: 0.0142 USD per Hour On-Demand SUSE base pricing: 0.0124 USD per Hour

All generations

Compare instance types

Additional costs apply for AMIs with pre-installed software

Key pair (login) Info

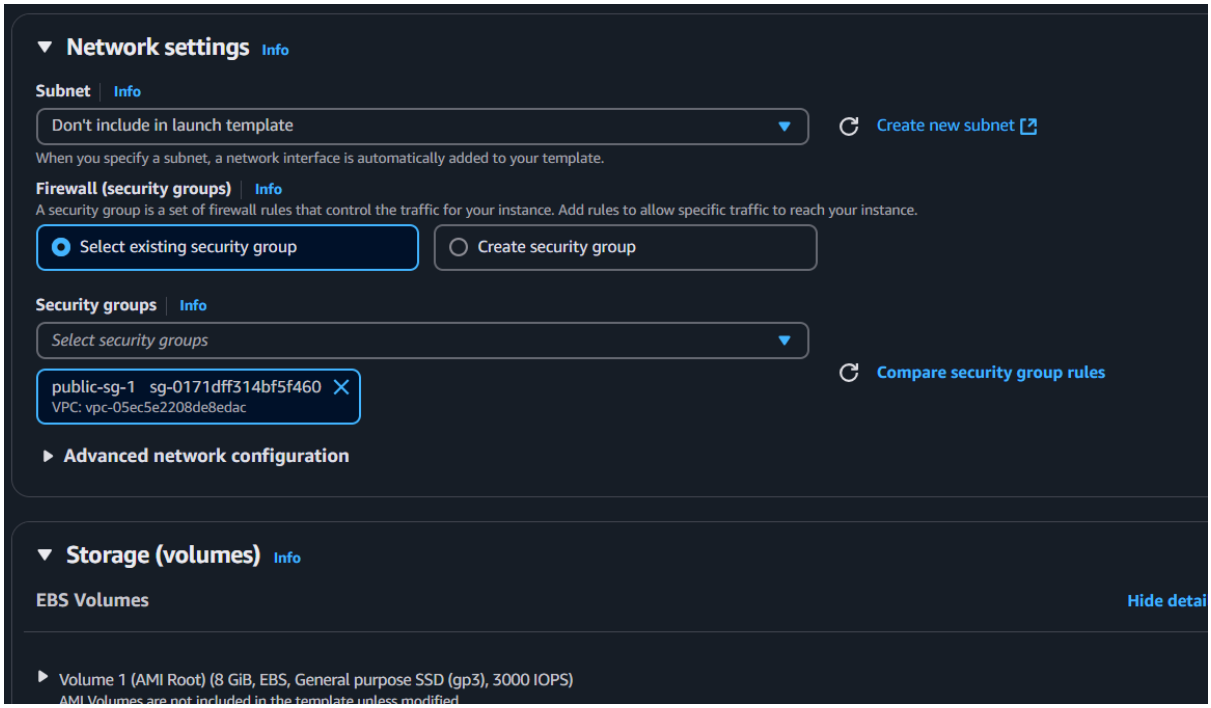
You can use a key pair to securely connect to your instance. Ensure that you have access to the selected key pair before you launch the instance.

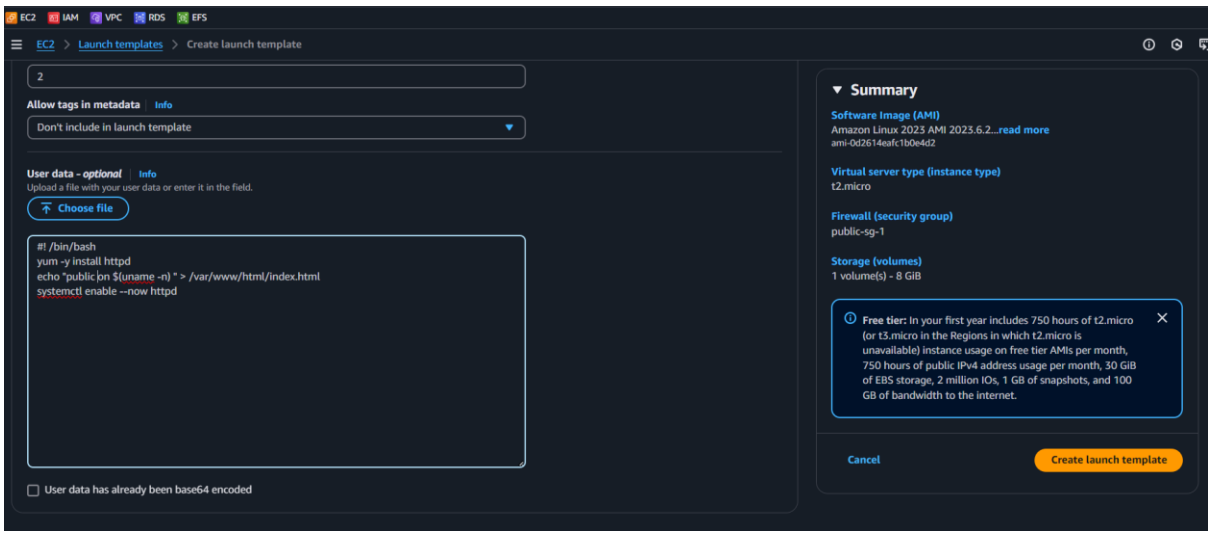
Key pair name

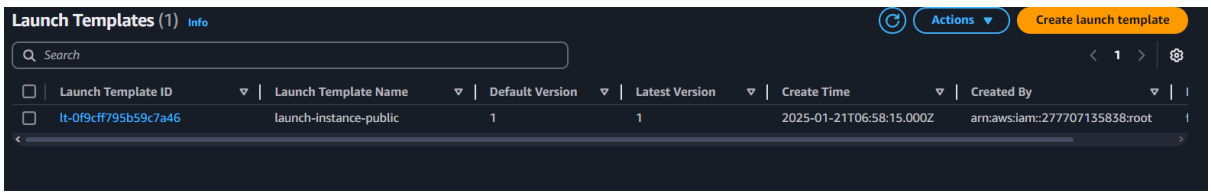
Key_2.1.25

Create new key pair

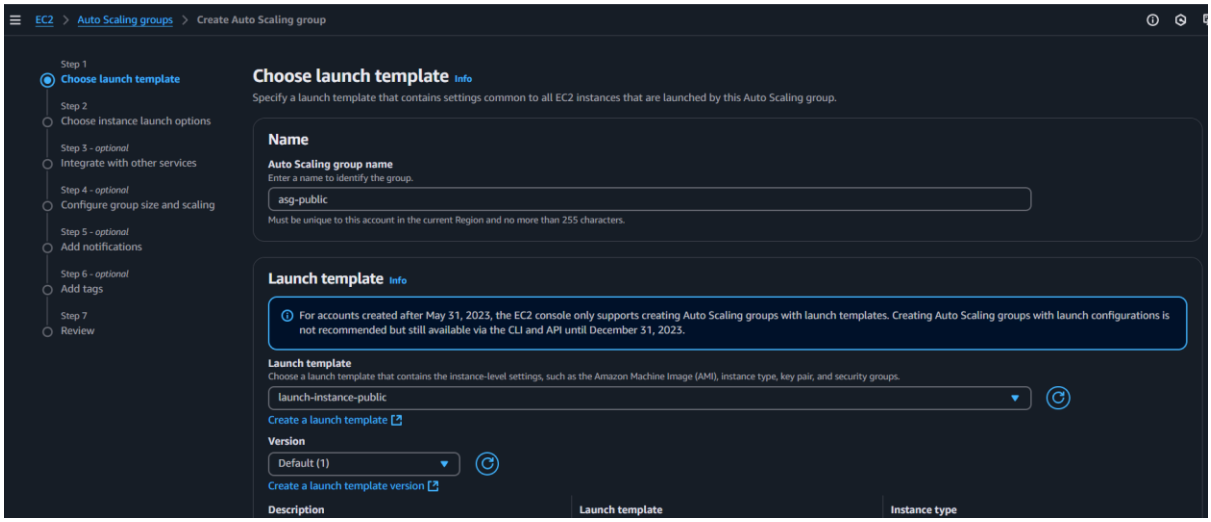
Network settings Info

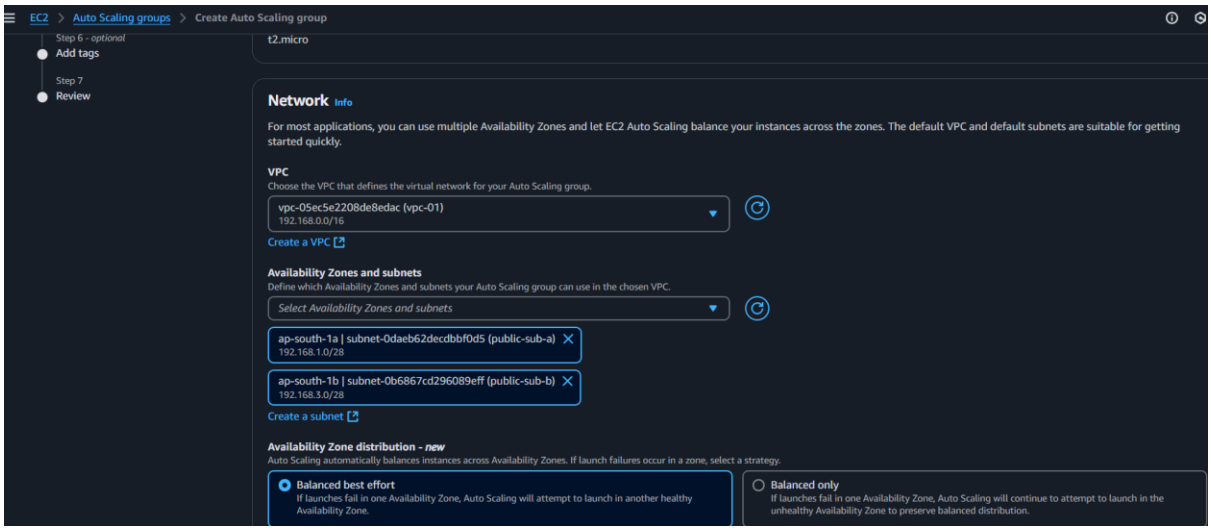


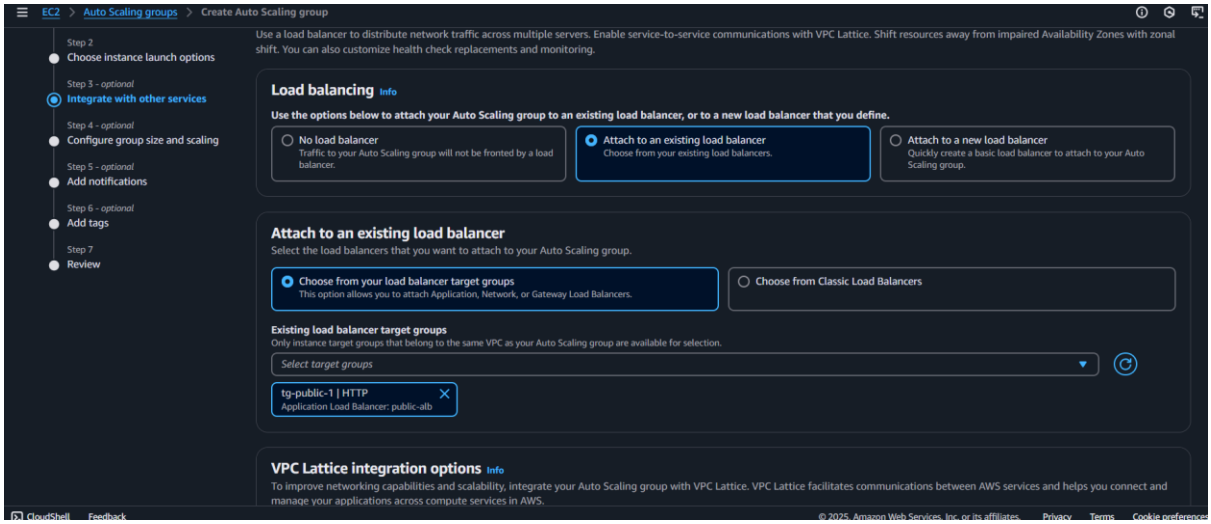


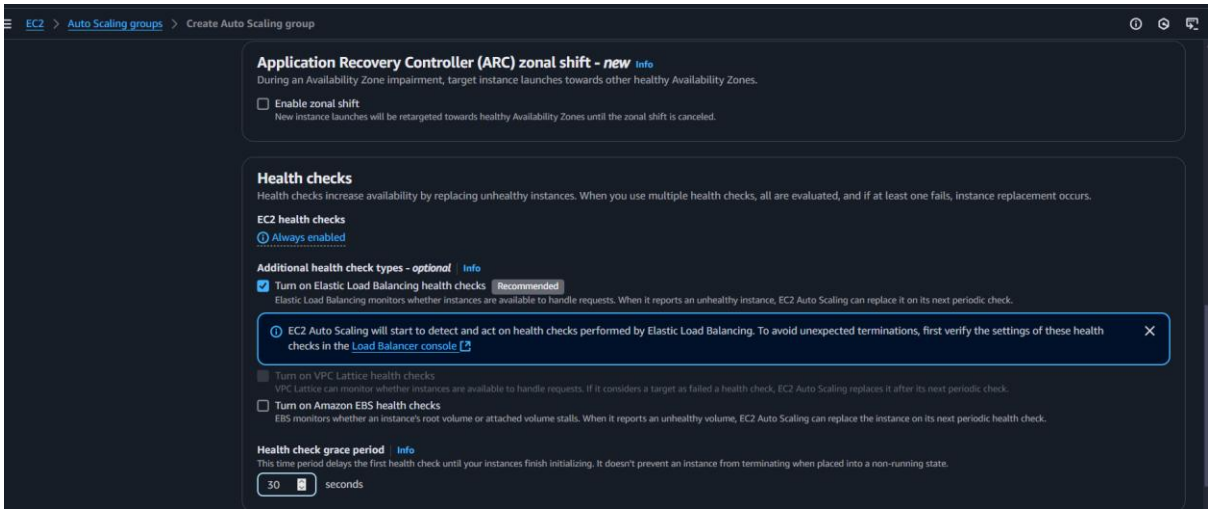


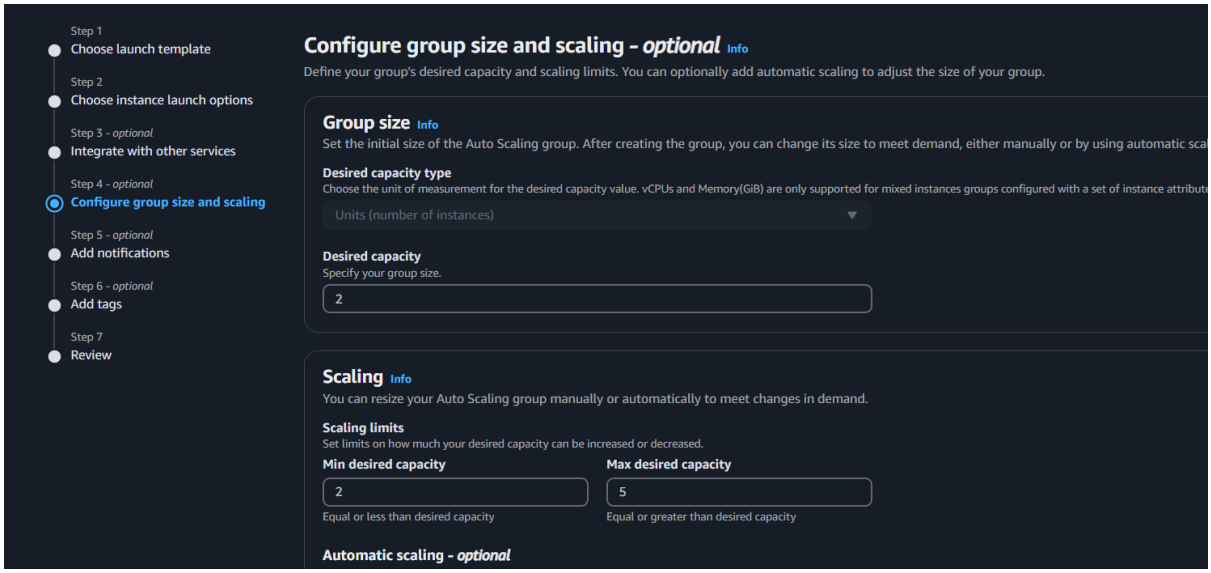
Step-3: Create autoscaling group

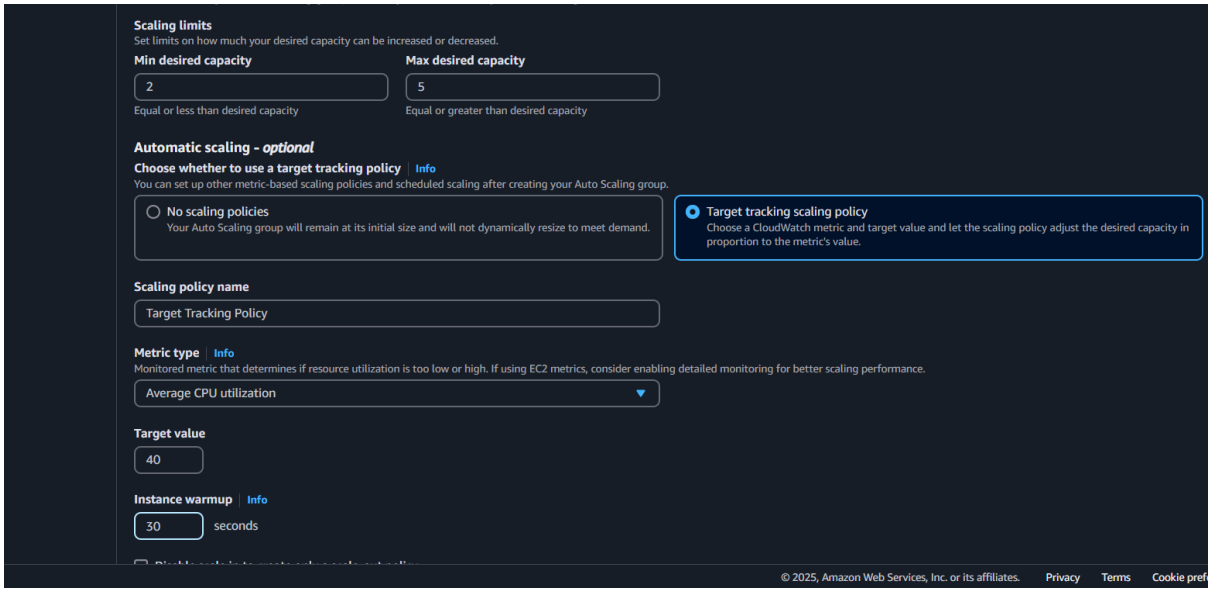

EC2 > Auto Scaling groups > Create Auto Scaling group
Step 1
Choose launch template
Step 2
Choose instance launch options
Step 3 - optional
Integrate with other services
Step 4 - optional
Configure group size and scaling
Step 5 - optional
Add notifications
Step 6 - optional
Add tags
Step 7
Review
Choose launch template Info
Specify a launch template that contains settings common to all EC2 instances that are launched by this Auto Scaling group.
Name
Auto Scaling group name
Enter a name to identify the group.
asg-public
Must be unique to this account in the current Region and no more than 255 characters.
Launch template Info
For accounts created after May 31, 2023, the EC2 console only supports creating Auto Scaling groups with launch templates. Creating Auto Scaling groups with launch configurations is not recommended but still available via the CLI and API until December 31, 2023.
Launch template
Choose a launch template that contains the instance-level settings, such as the Amazon Machine Image (AMI), instance type, key pair, and security groups.
launch-instance-public
Create a launch template
Version
Default (1)
Create a launch template version
Description
Launch template
Instance type

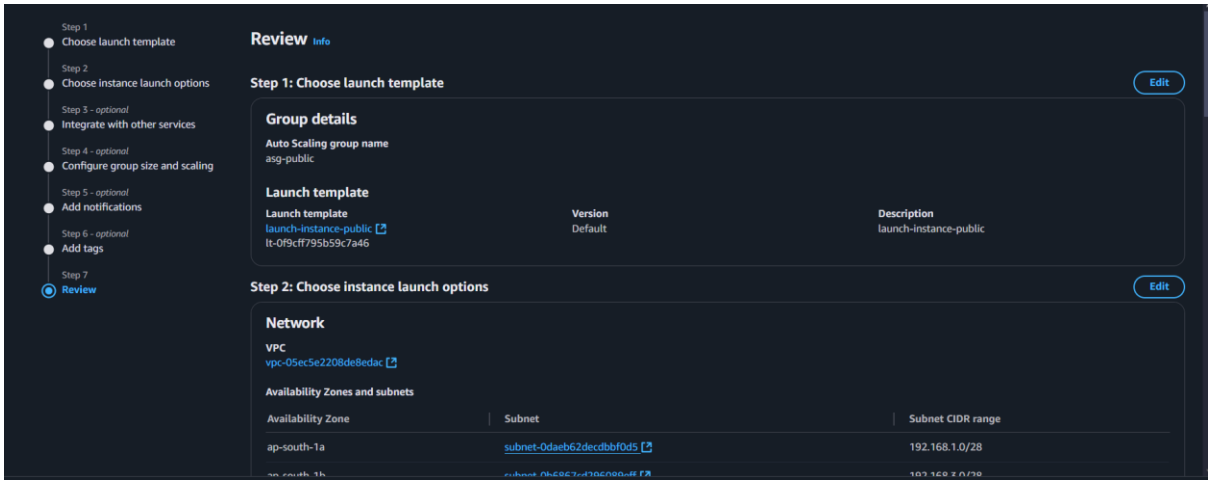

EC2 > Auto Scaling groups > Create Auto Scaling group
Step 6 - optional
Add tags
Step 7
Review
t2.micro
Network Info
For most applications, you can use multiple Availability Zones and let EC2 Auto Scaling balance your instances across the zones. The default VPC and default subnets are suitable for getting started quickly.
VPC
Choose the VPC that defines the virtual network for your Auto Scaling group.
vpc-05ec5e2208de8edac (vpc-01)
192.168.0.0/16
Create a VPC
Availability Zones and subnets
Define which Availability Zones and subnets your Auto Scaling group can use in the chosen VPC.
Select Availability Zones and subnets
ap-south-1a | subnet-0daeb62decdbbf0d5 (public-sub-a)
192.168.1.0/28
ap-south-1b | subnet-0b6867cd296089eff (public-sub-b)
192.168.3.0/28
Create a subnet
Availability Zone distribution - new
Auto Scaling automatically balances instances across Availability Zones. If launch failures occur in a zone, select a strategy.
Balanced best effort
If launches fail in one Availability Zone, Auto Scaling will attempt to launch in another healthy Availability Zone.
Balanced only
If launches fail in one Availability Zone, Auto Scaling will continue to attempt to launch in the unhealthy Availability Zone to preserve balanced distribution.

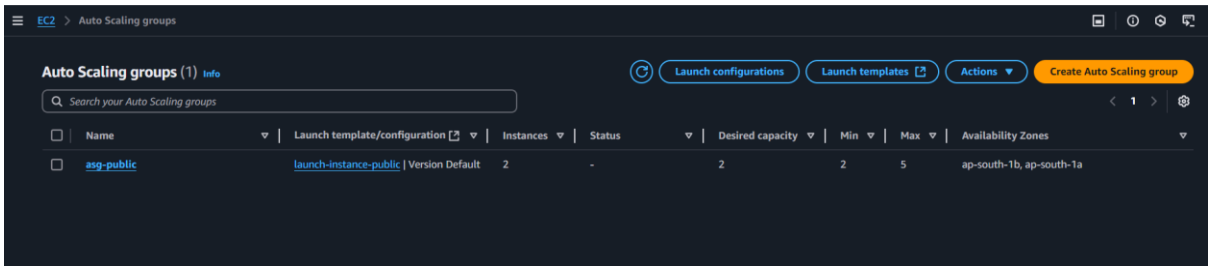

EC2 > Auto Scaling groups > Create Auto Scaling group
Step 2
Choose instance launch options
Step 3 - optional
Integrate with other services
Step 4 - optional
Configure group size and scaling
Step 5 - optional
Add notifications
Step 6 - optional
Add tags
Step 7
Review
Use a load balancer to distribute network traffic across multiple servers. Enable service-to-service communications with VPC Lattice. Shift resources away from impaired Availability Zones with zonal shift. You can also customize health check replacements and monitoring.
Load balancing Info
Use the options below to attach your Auto Scaling group to an existing load balancer, or to a new load balancer that you define.
No load balancer
Traffic to your Auto Scaling group will not be fronted by a load balancer.
Attach to an existing load balancer
Choose from your existing load balancers.
Attach to a new load balancer
Quickly create a basic load balancer to attach to your Auto Scaling group.
Attach to an existing load balancer
Select the load balancers that you want to attach to your Auto Scaling group.
Choose from your load balancer target groups
This option allows you to attach Application, Network, or Gateway Load Balancers.
Choose from Classic Load Balancers
Existing load balancer target groups
Only instance target groups that belong to the same VPC as your Auto Scaling group are available for selection.
Select target groups
tg-public-1 | HTTP
Application Load Balancer: public-alb
VPC Lattice integration options Info
To improve networking capabilities and scalability, integrate your Auto Scaling group with VPC Lattice. VPC Lattice facilitates communications between AWS services and helps you connect and manage your applications across compute services in AWS.
CloudShell
Feedback
© 2025, Amazon Web Services, Inc. or its affiliates.
Privacy
Terms
Cookie preferences


EC2 > Auto Scaling groups > Create Auto Scaling group
Application Recovery Controller (ARC) zonal shift - new Info
During an Availability Zone impairment, target instance launches towards other healthy Availability Zones.
Enable zonal shift
New instance launches will be retargeted towards healthy Availability Zones until the zonal shift is canceled.
Health checks
Health checks increase availability by replacing unhealthy instances. When you use multiple health checks, all are evaluated, and if at least one fails, instance replacement occurs.
EC2 health checks
Always enabled
Additional health check types - optional Info
Turn on Elastic Load Balancing health checks Recommended
Elastic Load Balancing monitors whether instances are available to handle requests. When it reports an unhealthy instance, EC2 Auto Scaling can replace it on its next periodic check.
EC2 Auto Scaling will start to detect and act on health checks performed by Elastic Load Balancing. To avoid unexpected terminations, first verify the settings of these health checks in the Load Balancer console
Turn on VPC Lattice health checks
VPC Lattice can monitor whether instances are available to handle requests. If it considers a target as failed a health check, EC2 Auto Scaling replaces it after its next periodic check.
Turn on Amazon EBS health checks
EBS monitors whether an instance's root volume or attached volume stalls. When it reports an unhealthy volume, EC2 Auto Scaling can replace the instance on its next periodic health check.
Health check grace period Info
This time period delays the first health check until your instances finish initializing. It doesn't prevent an instance from terminating when placed into a non-running state.
30
seconds

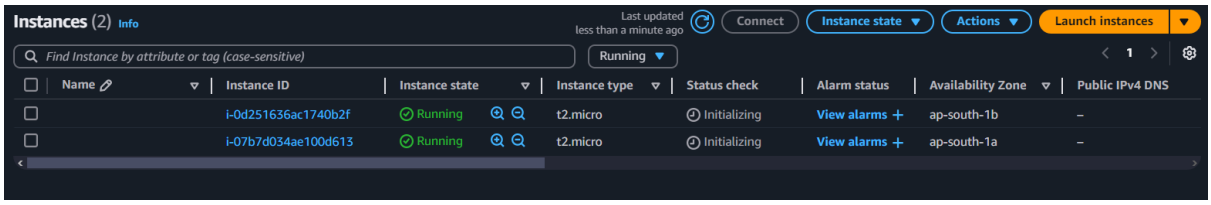

Step 1
Choose launch template
Step 2
Choose instance launch options
Step 3 - optional
Integrate with other services
Step 4 - optional
Configure group size and scaling
Step 5 - optional
Add notifications
Step 6 - optional
Add tags
Step 7
Review
Configure group size and scaling - optional Info
Define your group's desired capacity and scaling limits. You can optionally add automatic scaling to adjust the size of your group.
Group size Info
Set the initial size of the Auto Scaling group. After creating the group, you can change its size to meet demand, either manually or by using automatic sca
Desired capacity type
Choose the unit of measurement for the desired capacity value. vCPUs and Memory(GiB) are only supported for mixed instances groups configured with a set of instance attribut
Units (number of instances)
Desired capacity
Specify your group size.
2
Scaling Info
You can resize your Auto Scaling group manually or automatically to meet changes in demand.
Scaling limits
Set limits on how much your desired capacity can be increased or decreased.
Min desired capacity
2
Equal or less than desired capacity
Max desired capacity
5
Equal or greater than desired capacity
Automatic scaling - optional


Scaling limits
Set limits on how much your desired capacity can be increased or decreased.
Min desired capacity
2
Equal or less than desired capacity
Max desired capacity
5
Equal or greater than desired capacity
Automatic scaling - optional
Choose whether to use a target tracking policy Info
You can set up other metric-based scaling policies and scheduled scaling after creating your Auto Scaling group.
No scaling policies
Your Auto Scaling group will remain at its initial size and will not dynamically resize to meet demand.
Target tracking scaling policy
Choose a CloudWatch metric and target value and let the scaling policy adjust the desired capacity in proportion to the metric's value.
Scaling policy name
Target Tracking Policy
Metric type Info
Monitored metric that determines if resource utilization is too low or high. If using EC2 metrics, consider enabling detailed monitoring for better scaling performance.
Average CPU utilization
Target value
40
Instance warmup Info
30
seconds
© 2025, Amazon Web Services, Inc. or its affiliates.
Privacy
Terms
Cookie pref





Step-4: Check whether instances is add or not

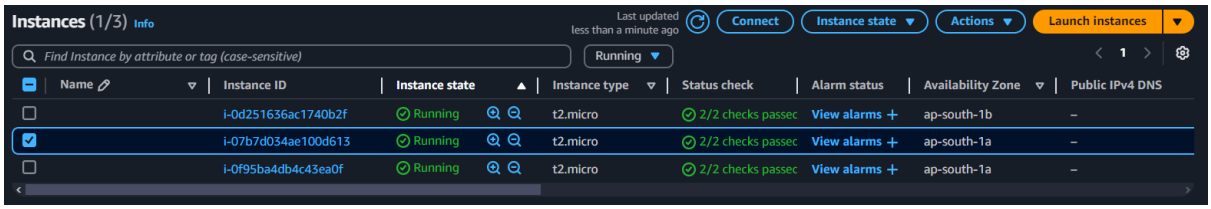


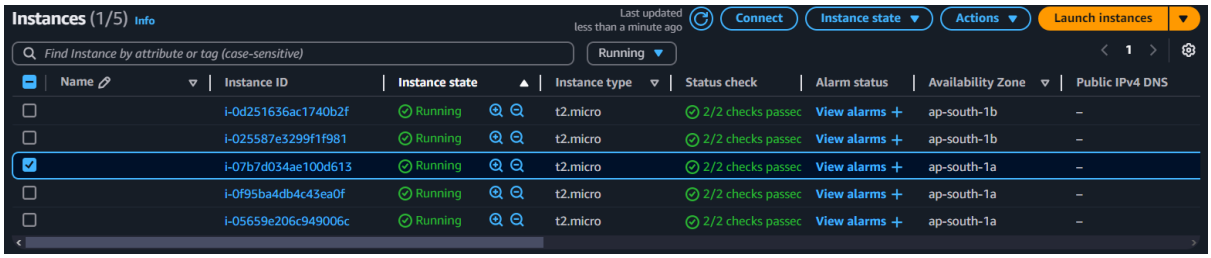
Step-5: Increase the load and check whether it is creating new instance or not

```
[root@ip-192-168-1-14 ~]# dd if=/dev/zero of=/dev/null &
[1] 26417
[root@ip-192-168-1-14 ~]# dd if=/dev/zero of=/dev/null &
[2] 26418
[root@ip-192-168-1-14 ~]# dd if=/dev/zero of=/dev/null &
[3] 26419
[root@ip-192-168-1-14 ~]# dd if=/dev/zero of=/dev/null &
[4] 26420
[root@ip-192-168-1-14 ~]# dd if=/dev/zero of=/dev/null &
[5] 26421
[root@ip-192-168-1-14 ~]#
```

```
top - 07:07:21 up 3 min,  2 users,  load average: 1.41, 0.48, 0.19
Tasks: 117 total,   6 running, 111 sleeping,   0 stopped,   0 zombie
%Cpu(s): 58.2 us, 40.5 sy,  0.0 ni,  0.0 id,  0.0 wa,  0.0 hi,  0.0 si,  1.3 st
MiB Mem :    949.5 total,    483.5 free,    158.7 used,    307.3 buff/cache
MiB Swap:      0.0 total,      0.0 free,      0.0 used.    647.5 avail Mem

    PID USER      PR  NI    VIRT    RES    SHR S  %CPU  %MEM     TIME+ COMMAND
  26417 root      20   0  221400   1012    924 R  20.1   0.1   0:06.32 dd
  26418 root      20   0  221400   1016    924 R  20.1   0.1   0:04.51 dd
  26420 root      20   0  221400   1016    924 R  20.1   0.1   0:04.07 dd
  26421 root      20   0  221400   1016    924 R  19.7   0.1   0:03.90 dd
  26419 root      20   0  221400   1016    924 R  19.4   0.1   0:04.20 dd
      1 root      20   0  106204  17412  10596 S   0.0   1.8   0:01.02 systemd
      2 root      20   0       0      0      0 S   0.0   0.0   0:00.00 kthreadd
      3 root       0 -20       0      0      0 I   0.0   0.0   0:00.00 rcu_gp
      4 root       0 -20       0      0      0 I   0.0   0.0   0:00.00 rcu_par_gp
      5 root       0 -20       0      0      0 I   0.0   0.0   0:00.00 slub_flushwq
      6 root       0 -20       0      0      0 I   0.0   0.0   0:00.00 netns
      7 root      20   0       0      0      0 I   0.0   0.0   0:00.00 kworker/0:0-cgroup_destroy
      8 root       0 -20       0      0      0 I   0.0   0.0   0:00.00 kworker/0:0H-events_highpri
      9 root      20   0       0      0      0 I   0.0   0.0   0:00.11 kworker/u30:0-writeback
     10 root       0 -20       0      0      0 I   0.0   0.0   0:00.00 mm_percpu_wq
     11 root      20   0       0      0      0 I   0.0   0.0   0:00.00 rcu_tasks_kthread
     12 root      20   0       0      0      0 I   0.0   0.0   0:00.00 rcu_tasks_rude_kthread
     13 root      20   0       0      0      0 I   0.0   0.0   0:00.00 rcu_tasks_trace_kthread
     14 root      20   0       0      0      0 S   0.0   0.0   0:00.15 ksoftirqd/0
```





Step-last: when load reduces it terminate the instance