TECHNOLOGY

Post Graduate Program in DevOps

TECHNOLOGY

Hotel-Side Hospital

simplilearn

Objectives

To create an automated provisioned infrastructure using Terraform, EKS cluster, EC2 instances, and Jenkins server

Problem Statement and Motivation

Hotel-Side Hospital, a globally renowned hospital chain headquartered in Australia, is aiming to streamline its operation by setting up an infrastructure within the hotel premises. However, in order to maintain seamless functioning and scalability, they require fully managed virtual machines (VMs) on the Amazon Web Services (AWS) platform.

The organization seeks an automated provisioned infrastructure solution that can enable them to effortlessly create new Amazon Elastic Kubernetes Service (EKS) clusters, whenever required, and promptly delete them when they are no longer needed. This will optimize resource allocation and enhance operational efficiency.

Industry Relevance

Skills used in the project and their usage in the industry are as below:

- **Terraform:** It is an infrastructure-as-code tool that allows you to define and provision resources in a cloud environment. In this project, Terraform is used to define the infrastructure components and manage their lifecycle.
- **EKS:** It is a managed Kubernetes service provided by AWS. In this project, an EKS cluster is created using Terraform, which provides a scalable and highly available environment for running containerized workloads.
- **EC2:** EC2 instances are virtual servers provided by AWS. In this project, EC2 instances are provisioned using Terraform, which allows you to specify the desired instance types, operating systems, and other configurations.

TECHNOLOGY

Industry Relevance

Skills used in the project and their usage in the industry are given below:

- **Jenkins:** It is an open-source automation server that facilitates continuous integration and continuous delivery (CI/CD) pipelines. In this project, Jenkins is used to orchestrate the CI/CD pipeline and integrates with the Terraform configurations to apply infrastructure changes and manage the deployment process.

simplilearn

Task (Activities)

1. Validate if Terraform is installed in the virtual machine
2. Install AWS CLI
3. Navigate to AWS IAM service, and get AWS Access key and Secret Key to connect AWS with the AWS CLI
4. Export the AWS Access Key, Secret Key, and Security Token to configure AWS CLI connectivity with AWS Cloud
5. Create terraform scripts to create a new VM using autoscaling which includes the following files: autoscaling.tf, VPC.tf, internetgateway.tf, subnets.tf (public subnet), routetable.tf, Route_table_association_with_public_subnets.tf

Task (Activities)

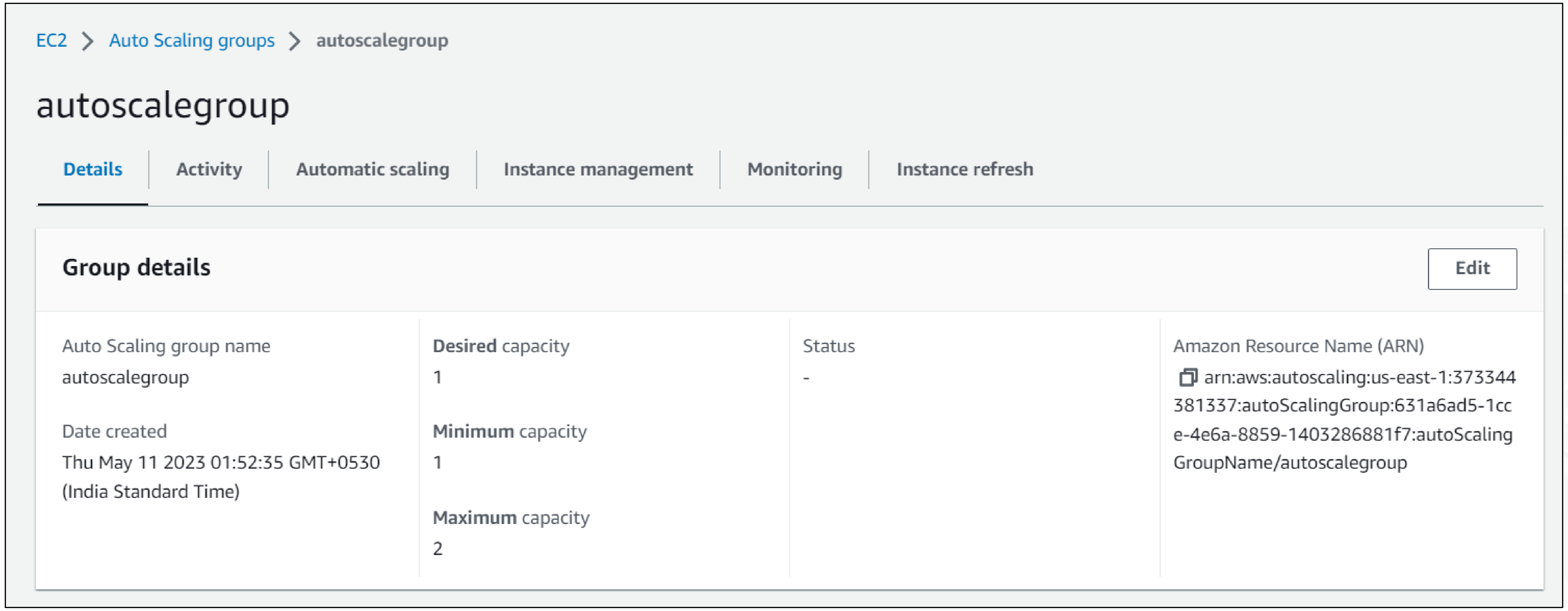
6. Execute terraform scripts
7. Connect to an instance and install the stress utility (The stress files are provided along with the problem statement document.)
8. Validate if autoscaling is working by putting load on autoscaling group

Project Reference

- **Task 1 and 2:** Course 2, Terraform Loops Built in Functions Provisioners
- **Task 3, 4, 5, 6, 8:** Course2, Terraform, Terraform Loops Built in Functions Provisioners

Output Screenshot

The autoscaling page after execution of terraform scripts:



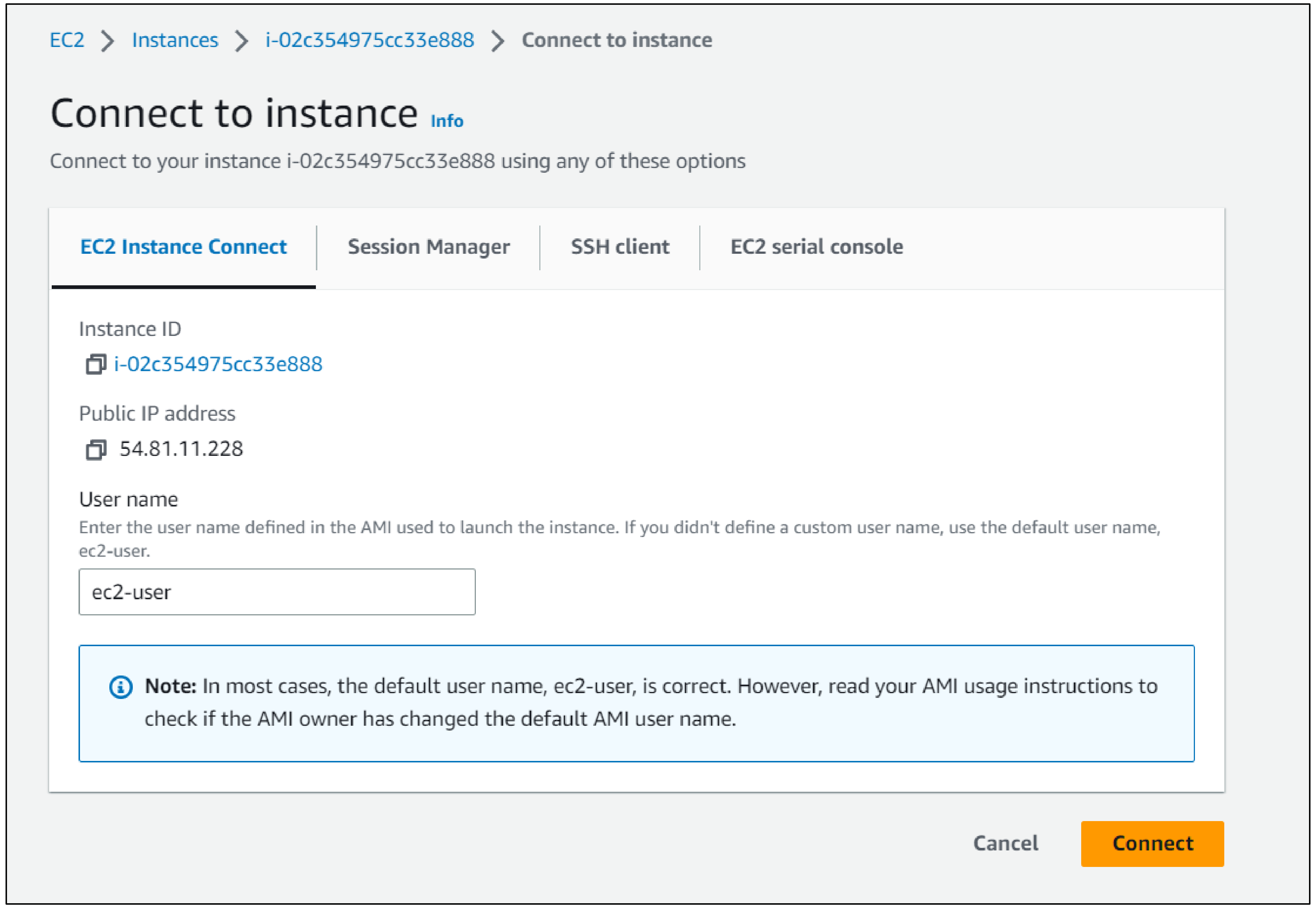
autoscalegroup

Group details

Auto Scaling group name	**Desired** capacity	Status	Amazon Resource Name (ARN)
autoscalegroup	1	-	arn:aws:autoscaling:us-east-1:373344381337:autoScalingGroup:631a6ad5-1cce-4e6a-8859-1403286881f7:autoScalingGroupName/autoscalegroup
Date created	**Minimum** capacity		
Thu May 11 2023 01:52:35 GMT+0530 (India Standard Time)	1		
	Maximum capacity		
	2		

Output Screenshot

EC2 instance connect page:



EC2 > Instances > i-02c354975cc33e888 > Connect to instance

Connect to instance Info

Connect to your instance i-02c354975cc33e888 using any of these options

Instance ID

i-02c354975cc33e888

Public IP address

54.81.11.228

User name

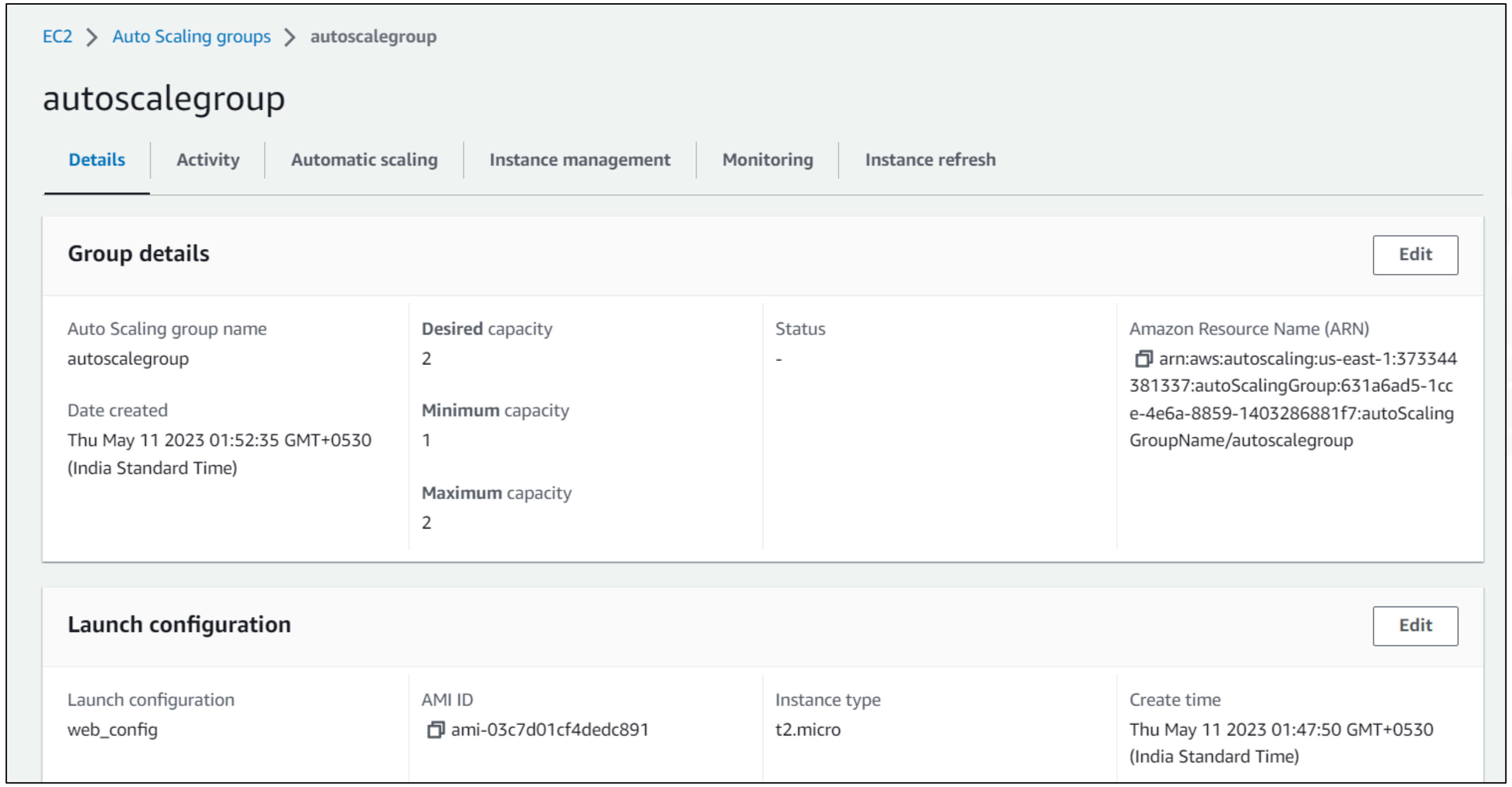
Enter the user name defined in the AMI used to launch the instance. If you didn't define a custom user name, use the default user name, ec2-user.

ec2-user

Note: In most cases, the default user name, ec2-user, is correct. However, read your AMI usage instructions to check if the AMI owner has changed the default AMI user name.

Output Screenshot

New instance getting launch after executing stress commands in the EC2 instance:



autoscalegroup

Group details

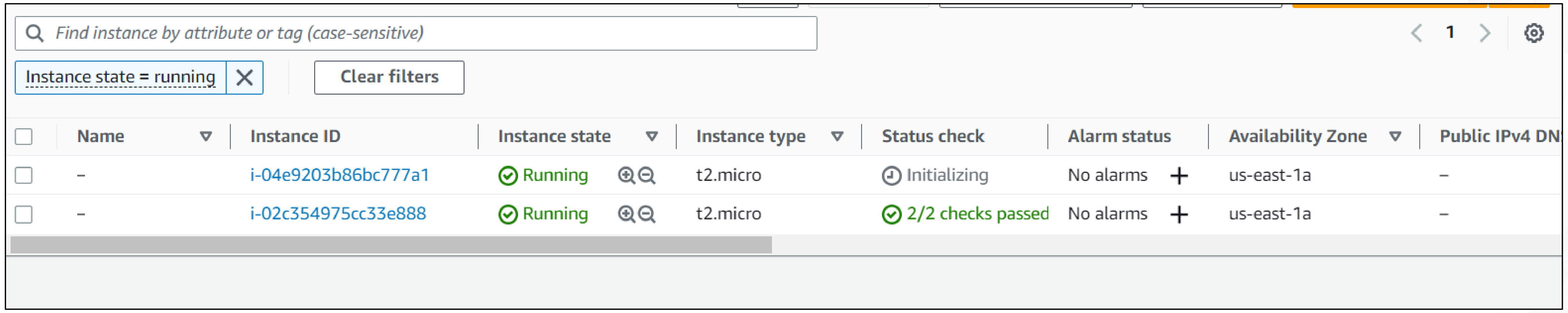
Auto Scaling group name autoscalegroup	Desired capacity 2	Status -	Amazon Resource Name (ARN) arn:aws:autoscaling:us-east-1:373344381337:autoScalingGroup:631a6ad5-1cce-4e6a-8859-1403286881f7:autoScalingGroupName/autoscalegroup
Date created Thu May 11 2023 01:52:35 GMT+0530 (India Standard Time)	Minimum capacity 1		
	Maximum capacity 2		

Launch configuration

Launch configuration web_config	AMI ID ami-03c7d01cf4dedc891	Instance type t2.micro	Create time Thu May 11 2023 01:47:50 GMT+0530 (India Standard Time)

Output Screenshot

The newly launched instance:



Find instance by attribute or tag (case-sensitive)

Instance state = running

Clear filters

	Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IPv4 DN
	–	i-04e9203b86bc777a1	Running	t2.micro	Initializing	No alarms	us-east-1a	–
	–	i-02c354975cc33e888	Running	t2.micro	2/2 checks passed	No alarms	us-east-1a	–

TECHNOLOGY

Thank You

simplilearn