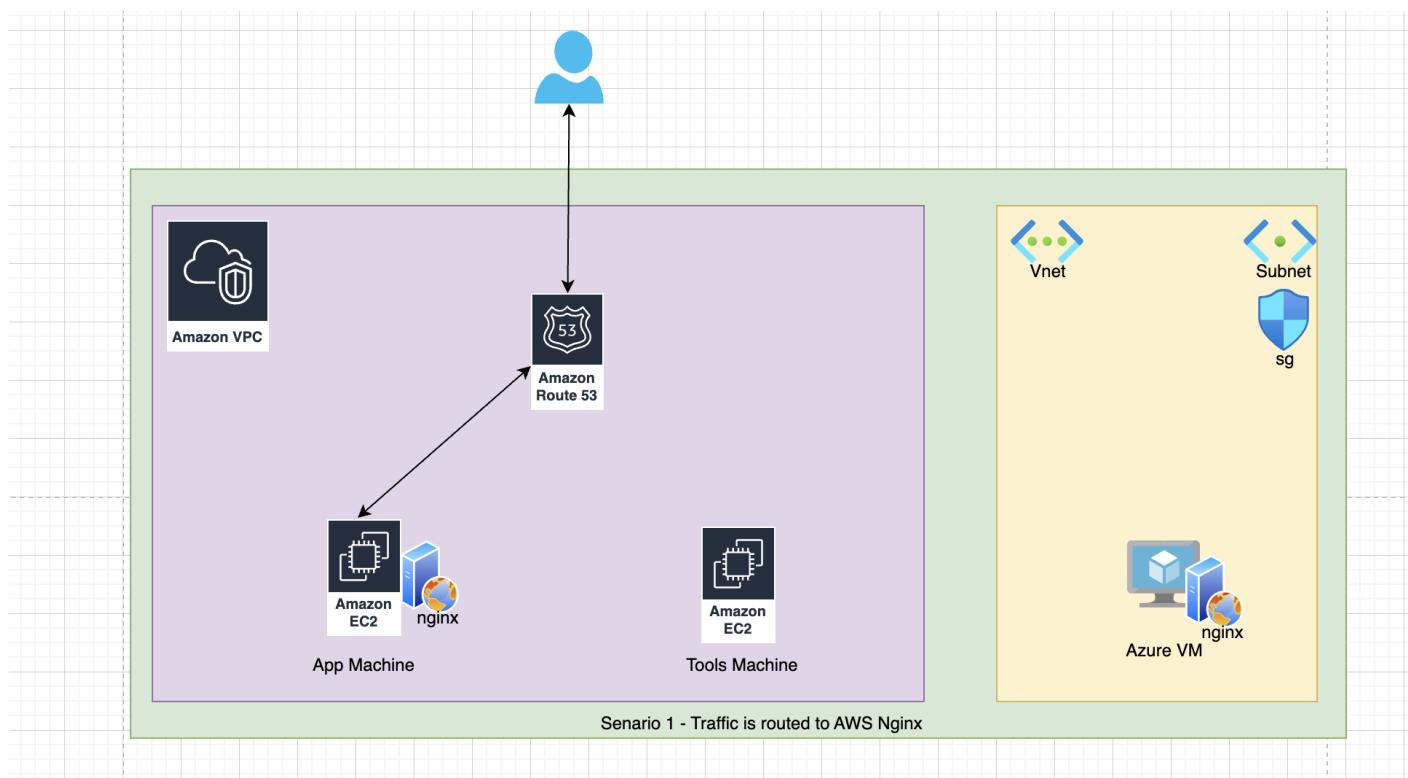


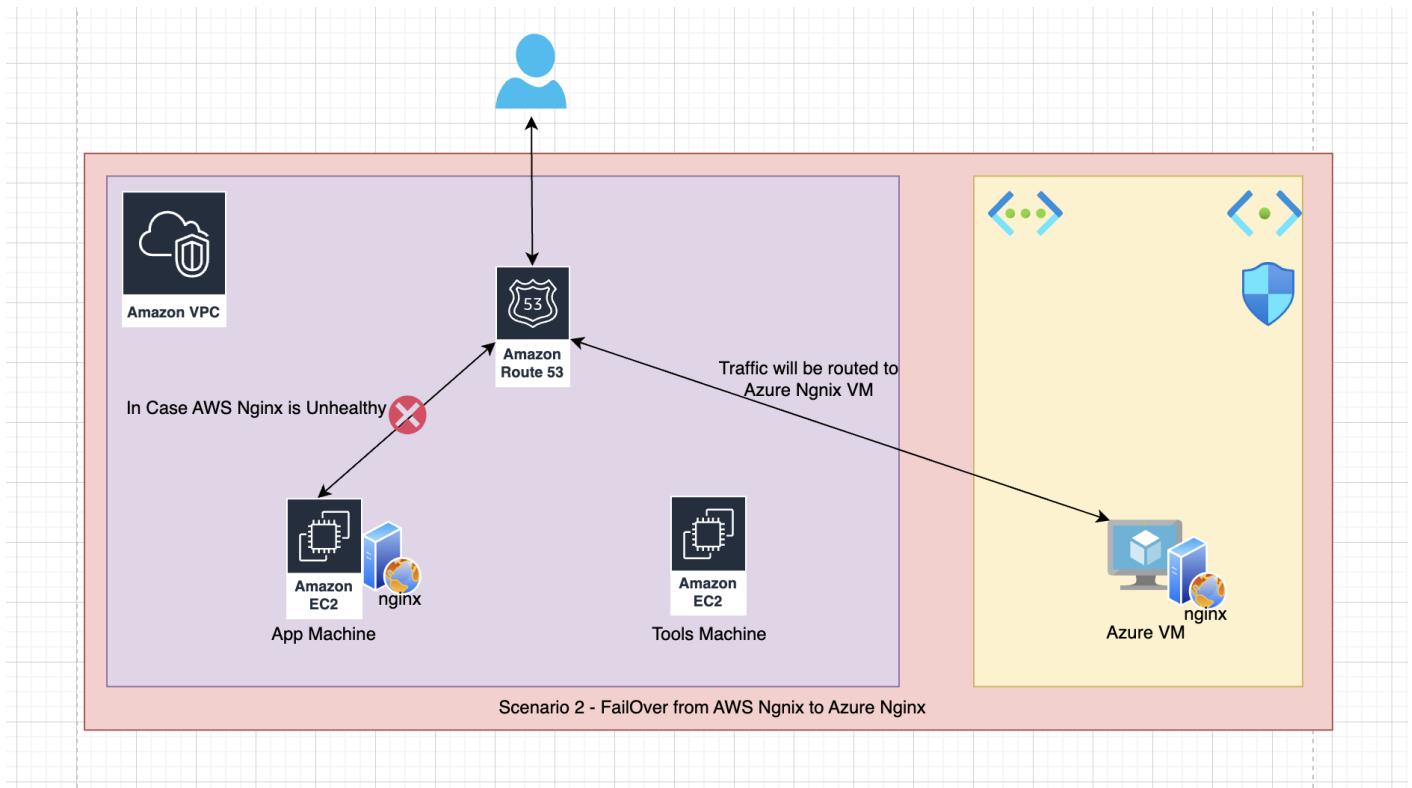
Capstone Project Solution

- By Saurav Suman

Architecture Diagram –

Below is the architecture diagram of this project which demonstrate the overall multi cloud infrastructure components involved, data flow and failover.





Google Drive Link –

This consists of the folders structure of each tasks with codebase, logs and screenshots.

https://drive.google.com/drive/folders/1NCnNKw_ta84YlzVthDWNf07rDsa0DArJ?usp=sharing

Public access has been given to the above google drive link.

Task 1: Infrastructure Provisioning

In this task, you will provision infrastructure using Terraform, automating the setup of cloud resources on AWS and Azure for a seamless, scalable, and efficient deployment. This task is divided into two subtasks: AWS and Azure.

AWS Setup

In the AWS part of this task, you will:

- Configure VPC with two public and two private subnets and Internet Gateway (IGW), NAT Gateway, and Route Tables. You may use a modular or resource-based Terraform approach.
- Create a security group with ports 22 (SSH) and 80 (HTTP) open to all and generate an EC2 key pair.
- Deploy:
 - Two EC2 instances in the public subnet using the pre-created security group and key pair.
 - Designate one EC2 instance as the App Machine and the other as the Tools Machine.
 - Use Ubuntu OS for both instances.

Azure Setup

In the Azure part of this task, you will:

- Configure Virtual Network (VNet), subnets, and Network Security Groups (NSGs).
- Deploy:
 - Virtual Machines (VMs) for application hosting, keeping 22 and 80 ports open.

Ensure SSH connectivity to both AWS and Azure instances from your local machine to verify if you have correctly set up the services.

You may also manually complete this task by provisioning AWS and Azure services without using Terraform. However, using Terraform is strongly recommended for better automation and efficiency.

Solution For Task 1 :

Terraform Script for Infrastructure deployment on AWS.

First, generate the secret key and access key of my user to use it for running terraform on my local laptop.

Access keys (2)

Use access keys to send programmatic calls to AWS from the AWS CLI, AWS Tools for PowerShell, AWS SDKs, or direct AWS API calls. You can have a maximum of two access keys (active or inactive) at a time. [Learn more](#)

AKIAVE3YXVCTL3CRL3NT	
Description	19mar2025
Last used	None
Last used region	N/A
Status	Active
Created	Now
Last used service	N/A

[Actions ▾](#)

Configure aws with the access key and secret key in the laptop to connect to aws platform.

```
(base) sauravsuman@Sauravs-MacBook-Pro awsterrcode % aws configure
AWS Access Key ID [*****X7HS]: AKIAVE3YXVCTL3CRL3NT
AWS Secret Access Key [*****gi9y]: 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
Default region name [us-east-1]:
Default output format [json]:
(base) sauravsuman@Sauravs-MacBook-Pro awsterrcode %
```

Terraform code file structure.

```
(base) sauravsuman@Sauravs-MacBook-Pro awsterrcode % ls -ltr
total 24
-rw-r--r-- 1 sauravsuman staff 389 19 Mar 14:36 provider.tf
-rw-r--r-- 1 sauravsuman staff 4394 19 Mar 14:36 deploy_aws_infra.tf
(base) sauravsuman@Sauravs-MacBook-Pro awsterrcode %
```

Create S3 bucket in aws for storing the terraform state file.

The screenshot shows the AWS S3 console. On the left, there's a sidebar with navigation links like 'Amazon S3', 'General purpose buckets', 'Directory buckets', etc. The main area has a heading 'Account snapshot - updated every 24 hours' with a link to 'All AWS Regions'. Below it, there are tabs for 'General purpose buckets' (which is selected) and 'Directory buckets'. A search bar says 'Find buckets by name'. A table lists one bucket: 'saurav-capstone-terraform-bucket' (Name), 'US East (N. Virginia) us-east-1' (AWS Region), 'View analyzer for us-east-1' (IAM Access Analyzer), and 'March 15, 2025, 12:12:25 (UTC+05:30)' (Creation date). There are buttons for 'Copy ARN', 'Empty', 'Delete', and 'Create bucket'.

Create DynamoDB table to lock the statefile.

DynamoDB > Tables > Create table

Create table

Table details info

DynamoDB is a schemaless database that requires only a table name and a primary key when you create the table.

Table name
This will be used to identify your table.

Between 3 and 255 characters, containing only letters, numbers, underscores (_), hyphens (-), and periods (.)

Partition key
The partition key is part of the table's primary key. It is a hash value that is used to retrieve items from your table and allocate data across hosts for scalability and availability.

1 to 255 characters and case sensitive.

Sort key - optional
You can use a sort key as the second part of a table's primary key. The sort key allows you to sort or search among all items sharing the same partition key.

1 to 255 characters and case sensitive.

DynamoDB > Tables

The terraform-dynamo-locktbl table was created successfully.

Tables (1) Info

Name	Status	Partition key	Sort key	Indexes	Replication Regions	Deletion protection	Favorite	Read capacity mode	Write capacity mode	Total size	Last modified
terraform-dynamo-locktbl	Active	LockID (\$)	-	0	0	Off	☆	On-demand	On-demand	0 bytes	Str

Execute the terraform code to deploy required infra on aws cloud platform.

terraform init

```
(base) sauravsuman@sauravsuman-MacBook-Pro ~ % terraform init

Initializing the backend...

Successfully configured the backend "s3"! Terraform will automatically
use this backend unless the backend configuration changes.

Initializing provider plugins...
- Finding hashicorp/aws versions matching "5.66.0"...
- Installing hashicorp/aws v5.66.0...
- Installed hashicorp/aws v5.66.0 (signed by HashiCorp)

Terraform has created a lock file .terraform.lock.hcl to record the provider
selections it made above. Include this file in your version control repository
so that Terraform can guarantee to make the same selections by default when
you run "terraform init" in the future.

Terraform has been successfully initialized!

You may now begin working with Terraform. Try running "terraform plan" to see
any changes that are required for your infrastructure. All Terraform commands
should now work.

If you ever set or change modules or backend configuration for Terraform,
rerun this command to reinitialize your working directory. If you forget, other
commands will detect it and remind you to do so if necessary.
(base) sauravsuman@sauravsuman-MacBook-Pro ~ %
```

terraform plan

```
(base) sauravsuman@Saurav's-MacBook-Pro awsterrcode % terraform plan
Acquiring state lock. This may take a few moments...

Terraform used the selected providers to generate the following execution plan. Resource actions are indicated with the following symbols:
+ create

Terraform will perform the following actions:

# aws_eip.saurav_aws_nat_eip will be created
+ resource "aws_eip" "saurav_aws_nat_eip" {
    + allocation_id      = (known after apply)
    + arn                = (known after apply)
    + association_id    = (known after apply)
    + carrier_ip         = (known after apply)
    + customer_owned_ip = (known after apply)
    + domain             = (known after apply)
    + id                 = (known after apply)
    + instance            = (known after apply)
    + network_border_group = (known after apply)
    + network_interface   = (known after apply)
    + private_dns        = (known after apply)
    + private_ip          = (known after apply)
    + ptr_record          = (known after apply)
    + public_dns          = (known after apply)
    + public_ip            = (known after apply)
    + public_ipv4_pool    = (known after apply)
    + tags               = {
        + "Name" = "saurav_aws_nat_eip"
    }
    + tags_all           = {
        + "Name" = "saurav_aws_nat_eip"
    }
    + vpc                = (known after apply)
}
```

```
# aws_vpc.saurav_aws_vpc will be created
+ resource "aws_vpc" "saurav_aws_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 = true
    + 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" = "saurav_aws_vpc"
    }
    + tags_all             = {
        + "Name" = "saurav_aws_vpc"
    }
}
```

Plan: 19 to add, 0 to change, 0 to destroy.

```
Note: You didn't use the -out option to save this plan, so Terraform can't guarantee to take exactly these actions if you run "terraform apply" now.
Releasing state lock. This may take a few moments...
(base) sauravsuman@Saurav's-MacBook-Pro awsterrcode %
```

terraform apply -auto-approve

```
(base) sauravsuman@Saurav-MacBook-Pro awsterrcode % terraform apply -auto-approve
Acquiring state lock. This may take a few moments...

Terraform used the selected providers to generate the following execution plan. Resource actions are indicated with the following symbols:
+ create

Terraform will perform the following actions:

# aws_eip.saurav_aws_nat_eip will be created
+ resource "aws_eip" "saurav_aws_nat_eip" {
    + allocation_id      = (known after apply)
    + arn                = (known after apply)
    + association_id    = (known after apply)
    + carrier_ip         = (known after apply)
    + customer_owned_ip = (known after apply)
    + domain             = (known after apply)
    + id                 = (known after apply)
    + instance            = (known after apply)
    + network_border_group = (known after apply)
    + network_interface   = (known after apply)
    + private_dns        = (known after apply)
    + private_ip          = (known after apply)
    + ptr_record          = (known after apply)
    + public_dns          = (known after apply)
    + public_ip            = (known after apply)
    + public_ipv4_pool    = (known after apply)
    + tags
        + "Name" = "saurav_aws_nat_eip"
    }
    + tags_all           = {
        + "Name" = "saurav_aws_nat_eip"
    }
    + vpc                = (known after apply)
}
```

```
aws_eip.saurav_aws_nat_eip: Creation complete after 3s [id=eipalloc-0d86116d23d7b1a69]
aws_vpc.saurav_aws_vpc: Still creating... [10s elapsed]
aws_vpc.saurav_aws_vpc: Creation complete after 15s [id=vpc-07245f249fc576a92]
aws_internet_gateway.saurav_aws_igw: Creating...
aws_route_table.saurav_aws_private_route_table: Creating...
aws_subnet.saurav_aws_private_subnet_1: Creating...
aws_subnet.saurav_aws_public_subnet_1: Creating...
aws_subnet.saurav_aws_private_subnet_2: Creating...
aws_subnet.saurav_aws_public_subnet_2: Creating...
aws_security_group.saurav_aws_sg: Creating...
aws_subnet.saurav_aws_private_subnet_2: Creation complete after 2s [id=subnet-0a9ff8b84788ccb64]
aws_internet_gateway.saurav_aws_igw: Creation complete after 2s [id=igw-0d796e73f99e4be84]
aws_route_table.saurav_aws_public_route_table: Creating...
aws_route_table.saurav_aws_private_route_table: Creation complete after 2s [id=rtb-0632d6d7a7d491483]
aws_route_table_association.saurav_aws_private_subnet_association_2: Creating...
aws_subnet.saurav_aws_private_subnet_1: Creation complete after 3s [id=subnet-077dc241e46fbe879]
aws_route_table_association.saurav_aws_private_subnet_association_1: Creating...
aws_route_table_association.saurav_aws_private_subnet_association_1: Creation complete after 1s [id=rtbassoc-05992cd29878b3459]
aws_route_table_association.saurav_aws_private_subnet_association_2: Creation complete after 2s [id=rtbassoc-03ca815b92e5dc55a]
aws_route_table.saurav_aws_public_route_table: Creation complete after 3s [id=rtb-0390e3f15f2b55c92]
aws_security_group.saurav_aws_sg: Creation complete after 6s [id=sg-0c9b6136af4a2bf5f]
aws_subnet.saurav_aws_public_subnet_1: Still creating... [10s elapsed]
aws_subnet.saurav_aws_public_subnet_2: Still creating... [10s elapsed]
aws_subnet.saurav_aws_public_subnet_1: Creation complete after 13s [id=subnet-050eabd20c1e8cf1c]
aws_subnet.saurav_aws_public_subnet_2: Creation complete after 17s [id=subnet-05a493c9265ce821a]
aws_route_table_association.saurav_aws_public_subnet_association_2: Creating...
aws_route_table_association.saurav_aws_public_subnet_association_1: Creating...
aws_nat_gateway.saurav_aws_nat_gateway: Creating...
aws_instance.app_machine: Creating...
aws_instance.tools_machine: Creating...
aws_route_table_association.saurav_aws_public_subnet_association_2: Creation complete after 1s [id=rtbassoc-043ff852aa48f6473]
aws_route_table_association.saurav_aws_public_subnet_association_1: Creation complete after 1s [id=rtbassoc-0a787f7499c149828b]
aws_nat_gateway.saurav_aws_nat_gateway: Still creating... [10s elapsed]
aws_instance.app_machine: Still creating... [10s elapsed]
aws_instance.tools_machine: Still creating... [10s elapsed]
aws_instance.app_machine: Creation complete after 15s [id=i-004a5039470b5fa2a]
aws_instance.tools_machine: Creation complete after 16s [id=i-02ced6c5731a6bf30]
aws_nat_gateway.saurav_aws_nat_gateway: Still creating... [20s elapsed]
aws_nat_gateway.saurav_aws_nat_gateway: Still creating... [30s elapsed]
aws_nat_gateway.saurav_aws_nat_gateway: Still creating... [40s elapsed]
aws_nat_gateway.saurav_aws_nat_gateway: Still creating... [50s elapsed]
aws_nat_gateway.saurav_aws_nat_gateway: Still creating... [1m0s elapsed]
aws_nat_gateway.saurav_aws_nat_gateway: Still creating... [1m20s elapsed]
aws_nat_gateway.saurav_aws_nat_gateway: Creation complete after 1m28s [id=nat-0b18fc6994eb6d4f2]
aws_route.saurav_aws_private_route_nat: Creating...
aws_route.saurav_aws_private_route_nat: Creation complete after 2s [id=r-rtb-0632d6d7a7d4914831080289494]
Releasing state lock. This may take a few moments...

Apply complete! Resources: 19 added, 0 changed, 0 destroyed.
(base) sauravsuman@Saurav-MacBook-Pro awsterrcode %
```

After running terraform apply, state file is created in S3 as mentioned in the below screenshot.

DynamoDB table also got the required data inserted.

Verify of the infra deployed from AWS Console.

VPC, Subnets, Routes tables, IGW and NAT.

Keypair created.

Key pairs (1) Info				
<input type="text"/> Find Key Pair by attribute or tag				Actions Create key pair
<input type="checkbox"/>	Name	Type	Created	Fingerprint
<input type="checkbox"/>	aws_upgrad_labkey	rsa	2025/03/15 12:50 GMT+5:30	46:0:cb:8:0:c:5:b:4:f:95:a:7:39:c:9:b:c:2:a:2:2:6:1:7:c:b:b

Security Group created.

The screenshot shows the AWS EC2 Security Groups page. A new security group named "sg-0c9b6136af4a2bf5f - saurav_aws_sg-20250319092125394300000002" has been created and is highlighted with a red box. The "Details" section shows the security group name, ID, owner, and various counts. Below it, the "Inbound rules" section is also highlighted with a red box, showing two entries: one for port 80 (HTTP) and one for port 22 (SSH).

2 EC2 Instances as the App Machine and the other as the Tools Machine created.

The screenshot shows the AWS EC2 Instances page. Two instances are listed: "Tools Machine" and "App Machine", both running in the "Running" state. Their details, including instance ID, type, and public IP addresses, are shown in a table. The entire table is highlighted with a red box.

SSH Test Connectivity to both the EC2 Instances.

1st Instance – App Machine

The screenshot shows the AWS EC2 Instance summary page for the instance "i-02ced6c5731a6bf30". The "Public IPv4 address" (44.202.103.139) is highlighted with a red box. Other details like instance ID, IPv6 address, host type, and private IP DNS name are also visible.

```
(base) sauravsuman@Saurav's-MacBook-Pro key % ssh -i aws_upgrad_labkey.pem ubuntu@44.202.103.139
The authenticity of host '44.202.103.139 (44.202.103.139)' can't be established.
ED25519 key fingerprint is SHA256:Al6W13QNaUCUEBu4E0ZKNTEcycAt+ACwC+Xum0P/d0FQ.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '44.202.103.139' (ED25519) to the list of known hosts.
Welcome to Ubuntu 24.04 LTS (GNU/Linux 6.8.0-1012-aws x86_64)

 * Documentation: https://help.ubuntu.com
 * Management: https://landscape.canonical.com
 * Support: https://ubuntu.com/pro

System information as of Wed Mar 19 09:32:34 UTC 2025

System load: 0.08      Processes:          113
Usage of /: 3.2% of 47.39GB  Users logged in:    0
Memory usage: 5%           IPv4 address for enX0: 10.0.2.38
Swap usage:  0%

Expanded Security Maintenance for Applications is not enabled.

0 updates can be applied immediately.

Enable ESM Apps to receive additional future security updates.
See https://ubuntu.com/esm or run: sudo pro status

The list of available updates is more than a week old.
To check for new updates run: sudo apt update

The programs included with the Ubuntu system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*copyright.

Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by
applicable law.

To run a command as administrator (user "root"), use "sudo <command>".
See "man sudo_root" for details.

ubuntu@ip-10-0-2-38:~$ hostname
ip-10-0-2-38
ubuntu@ip-10-0-2-38:~$ df -h
Filesystem      Size  Used Avail Use% Mounted on
/dev/root       48G   1.6G   46G  4% /
tmpfs          2.0G     0    2.0G  0% /dev/shm
tmpfs          783M  876K  782M  1% /run
tmpfs          5.0M     0    5.0M  0% /run/lock
/dev/xvda16     881M   76M  744M 10% /boot
/dev/xvda15     105M   6.1M   99M  6% /boot/efi
```

2nd Instance – Tools Machine

The screenshot shows the AWS EC2 Instances page with the following details for instance i-004a5039470b5fa2a:

- Instance summary for i-004a5039470b5fa2a (App Machine)**
- Public IP4 address:** 98.81.100.47 | [open address](#)
- Private IP4 addresses:** 10.0.1.248
- Public IP4 DNS:** ec2-98-81-100-47.compute-1.amazonaws.com | [open address](#)
- Private IP DNS name (IPv4 only):** ip-10-0-1-248.ec2.internal
- Instance state:** Running
- Instance type:** t2.micro

```

(base) sauravsuman@Saurav's-MacBook-Pro key % ssh -i aws_upgrad_labkey.pem ubuntu@98.81.100.47
the authenticity of host '98.81.100.47' (98.81.100.47) can't be established.
ED25519 Key fingerprint is SHA256:6CXZUL63m8tHN/ZFVt0L57je3y/CE9Y//uUha0IKTs.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '98.81.100.47' (ED25519) to the list of known hosts.
Welcome to Ubuntu 24.04 LTS (GNU/Linux 6.8.0-1012-aws x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:     https://landscape.canonical.com
 * Support:        https://ubuntu.com/pro

System information as of Wed Mar 19 09:34:31 UTC 2025

System load: 0.0          Processes:      103
Usage of /: 22.7% of 6.71GB  Users logged in:    0
Memory usage: 20%          IPv4 address for enX0: 10.0.1.248
Swap usage:   0%

Expanded Security Maintenance for Applications is not enabled.

0 updates can be applied immediately.

Enable ESM Apps to receive additional future security updates.
See https://ubuntu.com/esm or run: sudo pro status

The list of available updates is more than a week old.
To check for new updates run: sudo apt update

The programs included with the Ubuntu system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*copyright.

Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by
applicable law.

To run a command as administrator (user "root"), use "sudo <command>".
See "man sudo_root" for details.

ubuntu@ip-10-0-1-248:~$ hostname
ip-10-0-1-248
ubuntu@ip-10-0-1-248:~$ df -h
Filesystem      Size  Used Avail Use% Mounted on
/dev/root       6.8G  1.6G  5.2G  23% /
tmpfs           479M    0  479M   0% /dev/shm
tmpfs           192M  876K  191M   1% /run
tmpfs            5.0M    0   5.0M   0% /run/lock
/dev/xvda16     881M   76M  744M  10% /boot
/dev/xvda15     105M   6.1M  99M   6% /boot/efi

```

Infra Deployment on Azure Cloud Platform

Install azure cli on the local laptop and configure azure cli.

brew update && brew install azure-cli

az login

```
(base) sauravsuman@Sauravs-MacBook-Pro azreterrcode % az login
/opt/homebrew/Cellar/azure-cli/2.70.0/libexec/lib/python3.12/site-packages/azure/batch/models/_models.py:9067: SyntaxWarning: invalid escape sequence '\\'
    """The source port ranges to match for the rule. Valid values are '\ *' (for all ports 0 - 65535),
/opt/homebrew/Cellar/azure-cli/2.70.0/libexec/lib/python3.12/site-packages/azure/batch/models/_models.py:9225: SyntaxWarning: invalid escape sequence '\\'
    """A pattern indicating which file(s) to upload. Both relative and absolute paths are supported.
A web browser has been opened at https://login.microsoftonline.com/organizations/oauth2/v2.0/authorize. Please continue the login in the web browser. If no web browser is available or if the web browser fails to open, use device code flow with `az login --use-device-code`.

Retrieving tenants and subscriptions for the selection...

[Tenant and subscription selection]

No Subscription name Subscription ID Tenant
-----
[1] * npupgradl-1695790362718 ca4cf09e-d1dd-4217-929b-dd2904fa41f2 UpGrad

The default is marked with an *; the default tenant is 'UpGrad' and subscription is 'npupgradl-1695790362718' (ca4cf09e-d1dd-4217-929b-dd2904fa41f2).

Select a subscription and tenant (Type a number or Enter for no changes):

Tenant: UpGrad
Subscription: npupgradl-1695790362718 (ca4cf09e-d1dd-4217-929b-dd2904fa41f2)

[Announcements]
With the new Azure CLI login experience, you can select the subscription you want to use more easily. Learn more about it and its configuration at https://go.microsoft.com/fwlink/?linkid=2271236

If you encounter any problem, please open an issue at https://aka.ms/azclibug

[Warning] The login output has been updated. Please be aware that it no longer displays the full list of available subscriptions by default.

/opt/homebrew/Cellar/azure-cli/2.70.0/libexec/lib/python3.12/site-packages/deviceid/deviceid.py:9: SyntaxWarning: invalid escape sequence '\\S'
"""
/opt/homebrew/Cellar/azure-cli/2.70.0/libexec/lib/python3.12/site-packages/deviceid/_store.py:6: SyntaxWarning: invalid escape sequence '\\M'
    REGISTRY_PATH = 'SOFTWARE\Microsoft\DeveloperTools'
(base) sauravsuman@Sauravs-MacBook-Pro azreterrcode %
```

Terraform file structure for azure

```
(base) sauravsuman@Sauravs-MacBook-Pro 19march25 % ls -ltr
total 24
-rw-r--r-- 1 sauravsuman staff 220 19 Mar 14:52 provider.tf
-rw-r--r-- 1 sauravsuman staff 4729 19 Mar 14:52 deploy_azure_infra.tf
(base) sauravsuman@Sauravs-MacBook-Pro 19march25 %
```

Execute the terraform code :

terraform init

```
(base) sauravsuman@Sauravs-MacBook-Pro 19march25 % terraform init
Initializing the backend...
Initializing provider plugins...
- Finding hashicorp/azurerm versions matching "4.23.0"...
- Installing hashicorp/azurerm v4.23.0...
- Installed hashicorp/azurerm v4.23.0 (signed by HashiCorp)

Terraform has created a lock file .terraform.lock.hcl to record the provider
selections it made above. Include this file in your version control repository
so that Terraform can guarantee to make the same selections by default when
you run "terraform init" in the future.
```

Terraform has been successfully initialized!

You may now begin working with Terraform. Try running "terraform plan" to see
any changes that are required for your infrastructure. All Terraform commands
should now work.

If you ever set or change modules or backend configuration for Terraform,
rerun this command to reinitialize your working directory. If you forget, other
commands will detect it and remind you to do so if necessary.

terraform plan

```
(base) sauravsuman@Saurav's-MacBook-Pro 19march25 % terraform plan
Terraform used the selected providers to generate the following execution plan. Resource actions are indicated with the following symbols:
+ create

Terraform will perform the following actions:

# azurerm_linux_virtual_machine.azure_app_machine will be created
+ resource "azurerm_linux_virtual_machine" "azure_app_machine" {
    + admin_username          = "azureadmin"
    + allow_extension_operations = true
    + bypass_platform_safety_checks_on_user_schedule_enabled = false
    + computer_name           = (known after apply)
    + disable_password_authentication = true
    + disk_controller_type     = (known after apply)
    + extensions_time_budget   = "PT1H30M"
    + id                      = (known after apply)
    + location                = "eastus"
    + max_bid_price           = -1
    + name                    = "AzureAppMachine"
    + network_interface_ids   = (known after apply)
    + patch_assessment_mode   = "ImageDefault"
    + patch_mode               = "ImageDefault"
    + platform_fault_domain    = -1
    + priority                = "Regular"
    + private_ip_address       = (known after apply)
    + private_ip_addresses     = (known after apply)
    + provision_vm_agent      = true
    + public_ip_address        = (known after apply)
    + public_ip_addresses      = (known after apply)
    + resource_group_name      = "saurav_azure_rg"
    + size                     = "Standard_B2ms"
    + tags
        + "project" = "capstone"
    }
    + virtual_machine_id      = (known after apply)
    + vm_agent_platform_updates_enabled = false
}
```

```
# azurerm_subnet_route_table_association.saurav_azure_public_routetable_association will be created
+ resource "azurerm_subnet_route_table_association" "saurav_azure_public_routetable_association" {
    + id          = (known after apply)
    + route_table_id = (known after apply)
    + subnet_id    = (known after apply)
}

# azurerm_virtual_network.saurav_azure_vnet will be created
+ resource "azurerm_virtual_network" "saurav_azure_vnet" {
    + address_space      = [
        + "10.0.0.0/16",
    ]
    + dns_servers         = (known after apply)
    + guid                = (known after apply)
    + id                  = (known after apply)
    + location             = "eastus"
    + name                = "saurav_azure_vnet"
    + private_endpoint_vnet_policies = "Disabled"
    + resource_group_name  = "saurav_azure_rg"
    + subnet               = (known after apply)
    + tags
        + "project" = "capstone"
    }
}
```

Plan: 11 to add, 0 to change, 0 to destroy.

Note: You didn't use the -out option to save this plan, so Terraform can't guarantee to take exactly these actions if you run "terraform apply" now.
(base) sauravsuman@Saurav's-MacBook-Pro 19march25 %

terraform apply -auto-approve

```
(base) sauravsuman@Saurav's-MacBook-Pro: ~ % terraform apply -auto-approve
Terraform used the selected providers to generate the following execution plan. Resource actions are indicated with the following symbols:
+ create

Terraform will perform the following actions:

# azurerm_linux_virtual_machine.azure_app_machine will be created
+ resource "azurerm_linux_virtual_machine" "azure_app_machine" {
    + admin_username                = "azureadmin"
    + allow_extension_operations     = true
    + bypass_platform_safety_checks_on_user_schedule_enabled = false
    + computer_name                 = (Known after apply)
    + disable_password_authentication = true
    + disk_controller_type          = (Known after apply)
    + extensions_time_budget        = "PT1H30M"
    + id                            = (Known after apply)
    + location                      = "eastus"
    + max_bid_price                 = -1
    + name                          = "AzureAppMachine"
    + network_interface_ids         = (Known after apply)
    + patch_assessment_mode        = "ImageDefault"
    + patch_mode                    = "ImageDefault"
    + platform_fault_domain        = -1
    + priority                      = "Regular"
    + private_ip_address            = (Known after apply)
    + private_ip_addresses          = (Known after apply)
    + provision_vm_agent           = true
    + public_ip_address             = (Known after apply)
    + public_ip_addresses           = (Known after apply)
    + resource_group_name           = "saurav_azure_rg"
    + size                          = "Standard_B2ms"
    + tags                          = {
        + "project" = "capstone"
    }
    + virtual_machine_id            = (Known after apply)
    + vm_agent_platform_updates_enabled = false
}
```

```
Plan: 11 to add, 0 to change, 0 to destroy.
azurerm_resource_group.saurav_azure_rg: Creating...
azurerm_resource_group.saurav_azure_rg: Still creating... [10s elapsed]
azurerm_resource_group.saurav_azure_rg: Creation complete after 14s [id=/subscriptions/ca4cf09e-d1dd-4217-929b-dd2904fa41f2/resourceGroups/saurav_azure_rg]
azurerm_route_table.saurav_azure_public_route_table: Creating...
azurerm_virtual_network.saurav_azure_vnet: Creating...
azurerm_public_ip.saurav_public_ip: Creating...
azurerm_network_security_group.saurav_azure_nsrg: Creating...
azurerm_network_security_group.saurav_azure_nsrg: Creation complete after 4s [id=/subscriptions/ca4cf09e-d1dd-4217-929b-dd2904fa41f2/resourceGroups/saurav_azure_rg/providers/Microsoft.Network/networkSecurityGroups/saurav_azure_nsrg]
azurerm_public_ip.saurav_public_ip: Creation complete after 6s [id=/subscriptions/ca4cf09e-d1dd-4217-929b-dd2904fa41f2/resourceGroups/saurav_azure_rg/providers/Microsoft.Network/publicIPAddresses/saurav_public_ip]
azurerm_route_table.saurav_azure_public_route_table: Creation complete after 8s [id=/subscriptions/ca4cf09e-d1dd-4217-929b-dd2904fa41f2/resourceGroups/saurav_azure_rg/providers/Microsoft.Network/routeTables/saurav_azure_public_route_table]
azurerm_virtual_network.saurav_azure_vnet: Creation complete after 9s [id=/subscriptions/ca4cf09e-d1dd-4217-929b-dd2904fa41f2/resourceGroups/saurav_azure_rg/providers/Microsoft.Network/virtualNetworks/saurav_azure_vnet]
azurerm_subnet.saurav_private_subnet: Creating...
azurerm_subnet.saurav_public_subnet: Creating...
azurerm_subnet.saurav_public_subnet: Creation complete after 7s [id=/subscriptions/ca4cf09e-d1dd-4217-929b-dd2904fa41f2/resourceGroups/saurav_azure_rg/providers/Microsoft.Network/virtualNetworks/saurav_azure_vnet/subnets/saurav_public_subnet]
azurerm_subnet_route_table_association.saurav_azure_public_routetable_association: Creating...
azurerm_subnet_network_security_group_association.saurav_subnet_nsrg_assoc: Creating...
azurerm_network_interface.saurav_azure_vm_nic: Creating...
azurerm_subnet.saurav_private_subnet: Still creating... [10s elapsed]
azurerm_subnet.saurav_private_subnet: Creation complete after 14s [id=/subscriptions/ca4cf09e-d1dd-4217-929b-dd2904fa41f2/resourceGroups/saurav_azure_rg/providers/Microsoft.Network/virtualNetworks/saurav_azure_vnet/subnets/saurav_private_subnet]
azurerm_subnet_network_security_group_association.saurav_subnet_nsrg_assoc: Still creating... [10s elapsed]
azurerm_subnet_route_table_association.saurav_azure_public_routetable_association: Still creating... [10s elapsed]
azurerm_network_interface.saurav_azure_vm_nic: Still creating... [10s elapsed]
azurerm_subnet_route_table_association.saurav_azure_public_routetable_association: Creation complete after 14s [id=/subscriptions/ca4cf09e-d1dd-4217-929b-dd2904fa41f2/resourceGroups/saurav_azure_rg/providers/Microsoft.Network/routeTables/saurav_azure_public_routetable_association]
azurerm_network_interface.saurav_azure_vm_nic: Still creating... [20s elapsed]
azurerm_subnet_network_security_group_association.saurav_subnet_nsrg_assoc: Still creating... [20s elapsed]
azurerm_subnet_network_security_group_association.saurav_subnet_nsrg_assoc: Creation complete after 21s [id=/subscriptions/ca4cf09e-d1dd-4217-929b-dd2904fa41f2/resourceGroups/saurav_azure_rg/providers/Microsoft.Network/virtualNetworks/saurav_azure_vnet/subnets/saurav_public_subnet]
azurerm_network_interface.saurav_azure_vm_nic: Creation complete after 26s [id=/subscriptions/ca4cf09e-d1dd-4217-929b-dd2904fa41f2/resourceGroups/saurav_azure_rg/providers/Microsoft.Network/networkInterfaces/saurav_azure_vm_nic]
azurerm_linux_virtual_machine.azure_app_machine: Creating...
azurerm_linux_virtual_machine.azure_app_machine: Still creating... [10s elapsed]
azurerm_linux_virtual_machine.azure_app_machine: Creation complete after 17s [id=/subscriptions/ca4cf09e-d1dd-4217-929b-dd2904fa41f2/resourceGroups/saurav_azure_rg/providers/Microsoft.Compute/virtualMachines/AzureAppMachine]
```

```
Apply complete! Resources: 11 added, 0 changed, 0 destroyed.
```

Verify the required infra deployed on azure cloud using azure portal.

Vnet created.

The screenshot shows the 'Virtual networks' page in the Azure portal. A red box highlights the search bar and the results table. The table has columns: Name, Resource group, Location, and Subscription. One row is selected, showing 'saurav_azure_vnet' under 'Name', 'saurav_azure_rg' under 'Resource group', 'East US' under 'Location', and 'npupgradl-1695790362718' under 'Subscription'. The entire table row is also highlighted with a red box.

Subnet created.

The screenshot shows the 'saurav_azure_vnet | Subnets' page. A red box highlights the title bar. On the left is a navigation menu with items like Overview, Activity log, Access control (IAM), Tags, Diagnose and solve problems, Resource visualizer, Settings, Address space, and Connected devices. The main area shows a table of subnets. The table has columns: Name, IPv4, IPv6, Available IPs, Delegated to, Security group, and Route table. Two subnets are listed: 'saurav_private_subnet' (IPv4: 10.0.2.0/24) and 'saurav_public_subnet' (IPv4: 10.0.1.0/24). Both subnets have 251 and 250 available IPs respectively. The security group for both is 'saurav_azure_n...'. The route table for both is 'saurav_azu...'. The entire table is highlighted with a red box.

Virtual machine created.

The screenshot shows the 'Compute infrastructure | Virtual machines' page. A red box highlights the title bar. On the left is a navigation menu with items like Virtual machines, Create, Reservations, Manage view, Refresh, Export to CSV, Open query, Assign tags, Start, Stop, Delete, Services, and The main area shows a table of virtual machines. The table has columns: Name, Subscription, Resource group, Location, Status, Operating system, Size, Public IP address, and Disks. One VM is listed: 'AzureAppMachine' (Subscription: npupgradl-169579036..., Resource group: saurav_azure_rg, Location: East US, Status: Running, OS: Linux, Size: Standard_B2ms, Public IP address: 52.179.126.20, Disks: 1). The entire table is highlighted with a red box.

Virtual machine details with public IP.

The screenshot shows the 'Compute infrastructure | Virtual machines | AzureAppMachine | Network settings' page. A red box highlights the title bar. On the left is a navigation menu with items like Virtual machines, Create, Reservations, Manage view, Refresh, Open query, Assign tags, Start, Stop, Delete, Services, and The main area shows network settings for 'AzureAppMachine'. Under 'Network settings', the 'Network interface / IP configuration' section is expanded, showing 'saurav.azure_vm_nic (primary) / internal (primary)'. Within this section, the 'Public IP address' field is highlighted with a red box and contains the value '52.179.126.20'. Other network details shown include 'Private IP address' (10.0.1.4), 'Load balancers' (0), 'Application security groups' (0), 'Network security group' (none), 'Accelerated networking' (Disabled), and 'Effective security rules' (0).

ssh testing from the local laptop to the azure vm

```
(base) sauravsuman@Sauravs-MacBook-Pro .ssh % ls
known hosts      known hosts.old      saurav mac key      saurav mac key.pub
(base) sauravsuman@Sauravs-MacBook-Pro .ssh % ssh -i saurav_mac_key azureadmin@52.179.126.20
The authenticity of host '52.179.126.20 (52.179.126.20)' can't be established.
ED25519 key fingerprint is SHA256:KxUjTao3jiL6oNaZHE0y+Tt0LfK9ZZLxisH11IQQtKU.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '52.179.126.20' (ED25519) to the list of known hosts.
Welcome to Ubuntu 22.04.5 LTS (GNU/Linux 6.8.0-1021-azure x86_64)

 * Documentation: https://help.ubuntu.com
 * Management:   https://landscape.canonical.com
 * Support:       https://ubuntu.com/pro

System information as of Wed Mar 19 09:46:47 UTC 2025

System load: 0.1          Processes:           113
Usage of /: 5.2% of 28.89GB Users logged in: 0
Memory usage: 3%          IPv4 address for eth0: 10.0.1.4
Swap usage:  0%

Expanded Security Maintenance for Applications is not enabled.

0 updates can be applied immediately.

Enable ESM Apps to receive additional future security updates.
See https://ubuntu.com/esm or run: sudo pro status

The list of available updates is more than a week old.
To check for new updates run: sudo apt update

The programs included with the Ubuntu system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*copyright.

Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by
applicable law.

To run a command as administrator (user "root"), use "sudo <command>".
See "man sudo_root" for details.
```

```
azureadmin@AzureAppMachine:~$ hostname
AzureAppMachine
azureadmin@AzureAppMachine:~$ df -h
Filesystem      Size  Used Avail Use% Mounted on
/dev/root      29G   2G  28G  6% /
tmpfs          3.9G    0  3.9G  0% /dev/shm
tmpfs          1.6G  960K  1.6G  1% /run
tmpfs          5.0M    0  5.0M  0% /run/lock
```

Task 2: Configuration Management

In this task, you will:

- Install Ansible on the Tools Machine (AWS).
- Create an Ansible inventory file listing the details of AWS (App Machine) and Azure (VM).
- Develop an Ansible playbook to:
 - Install and configure Nginx web server on both AWS and Azure machines.
 - Ensure the Nginx process starts automatically and verify its running status in the playbook file.

To verify if Nginx is actively running, access the application via HTTP (IP:80) and confirm if both instances show the text, Welcome to Nginx.

Solution for Task 2 :

First, create public and private key using ssh-keygen command on the aws tools machine. The public key will be added to authorized_keys file on the both the aws and azure app machine remote servers. This will create passwordless connectivity from aws tools machine to both the app machine.

```
root@ip-10-0-2-38:~# ssh-keygen
Generating public/private ed25519 key pair.
Enter file in which to save the key (/root/.ssh/id_ed25519):
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /root/.ssh/id_ed25519
Your public key has been saved in /root/.ssh/id_ed25519.pub
The key fingerprint is:
SHA256:pF6acwBTv/jGpjwCvQLC0RAUV3dDLgCHL9/u0VmNmM root@ip-10-0-2-38
The key's randomart image is:
++-[ED25519 256] ++
|000o== ..+
|+=+=o o o .
|*o.o. ...
|+o+. . o.
|...oo o S o
| .o o + o .
| .+ +
| ...+.E
| .o..o.o
+---[SHA256]-----
root@ip-10-0-2-38:~# ls -ltr
total 4
drwx----- 3 root root 4096 Mar 19 09:22 snap
root@ip-10-0-2-38:~# cd .ssh/
root@ip-10-0-2-38:~/ssh# ls -ltr
total 12
-rw----- 1 root root 563 Mar 19 09:22 authorized_keys
-rw-r--r-- 1 root root 99 Mar 19 09:48 id_ed25519.pub
-rw----- 1 root root 411 Mar 19 09:48 id_ed25519
root@ip-10-0-2-38:~/ssh#
```

Install Ansible on the aws tools machine.

```
root@ip-10-0-2-29:~# apt-get update
```

```
root@ip-10-0-2-38:~/capstone-ansible# apt update
Hit:1 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble InRelease
Get:2 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates InRelease [126 kB]
Get:3 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-backports InRelease [126 kB]
Get:4 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/universe amd64 Packages [15.0 MB]
Get:5 http://security.ubuntu.com/ubuntu noble-security InRelease [126 kB]
Get:6 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/universe Translation-en [5982 kB]
Get:7 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/universe amd64 Components [3871 kB]
Get:8 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/universe amd64 c-n-f Metadata [301 kB]
Get:9 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/multiverse amd64 Packages [269 kB]
Get:10 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/multiverse Translation-en [118 kB]
Get:11 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/multiverse amd64 Components [35.0 kB]
Get:12 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/multiverse amd64 c-n-f Metadata [8328 B]
Get:13 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates/main amd64 Packages [921 kB]
Get:14 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates/main Translation-en [209 kB]
```

```
root@ip-10-0-2-29:~# apt-get install ansible
```

```
root@ip-10-0-2-38:~/capstone-ansible# apt install ansible
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following additional packages will be installed:
  ansible-core libselinux1 python3-argcomplete python3-dnspython python3-jmespath python3-kerberos python3-libcloud python3-lockfile python3-ntlm-auth python3-packaging python3-passlib
  python3-requests-ntlm python3-resolve1ib python3-selinux python3-simplejson python3-winrm python3-xm1todict
Suggested packages:
  cowsay sshpass python3-trio python3-aioquic python3-h2 python3-httplib python3-httplibcore python-lockfile-doc
The following NEW packages will be installed:
  ansible ansible-core python3-argcomplete python3-dnspython python3-jmespath python3-kerberos python3-libcloud python3-lockfile python3-ntlm-auth python3-packaging python3-passlib python3-requests-ntlm
  python3-resolve1ib python3-selinux python3-simplejson python3-winrm python3-xm1todict
The following packages will be upgraded:
  libselinux1
1 upgraded, 17 newly installed, 0 to remove and 239 not upgraded.
Need to get 19.6 MB of archives.
After this operation, 315 MB of additional disk space will be used.
Do you want to continue? [Y/n] Y
Get:1 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates/main amd64 libselinux1 amd64 3.5-2ubuntu2.1 [79.7 kB]
Get:2 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/main amd64 python3-packaging all 24.0-1 [41.1 kB]
root@ip-10-0-2-38:~/capstone-ansible# ansible --version
ansible [core 2.16.3]
  config file = None
  configured module search path = ['/root/.ansible/plugins/modules', '/usr/share/ansible/plugins/modules']
  ansible python module location = /usr/lib/python3/dist-packages/ansible
  ansible collection location = /root/.ansible/collections:/usr/share/ansible/collections
  executable location = /usr/bin/ansible
  python version = 3.12.3 (main, Apr 10 2024, 05:33:47) [GCC 13.2.0] (/usr/bin/python3)
  jinja version = 3.1.2
  libyaml = True
root@ip-10-0-2-38:~/capstone-ansible# █
```

Create inventory.ini file with 2 groups, one for aws nginx vm and other for azure nginx vm

```
root@ip-10-0-2-38:~/capstone-ansible# cat inventory.ini
[aws_nginx_app_vm]
98.81.100.47

[azure_nginx_app_vm]
52.179.126.20
root@ip-10-0-2-38:~/capstone-ansible#
```

Ping both the vms using playbook and test the connectivity from tools vm.

```
root@ip-10-0-2-38:~/capstone-ansible# cat ping_remote_vms.yml
---
- name: Ping both the remote vms and get response
  hosts: all
  tasks:
    - name: Ping the remote VM
      ping:

    - name: Respond with Pong
      debug:
        msg: "Pong"
```

```
root@ip-10-0-2-38:~/capstone-ansible#
```

Log out of execution of the above ansible playbook

```
root@ip-10-0-2-38:~/capstone-ansible# ansible-playbook -i inventory.ini ping_remote_vms.yml
PLAY [Ping both the remote vms and get response] ****
TASK [Gathering Facts] ****
ok: [52.179.126.20]
ok: [98.81.100.47]

TASK [Ping the remote VM] ****
ok: [98.81.100.47]
ok: [52.179.126.20]

TASK [Respond with Pong] ****
ok: [98.81.100.47] => {
    "msg": "Pong"
}
ok: [52.179.126.20] => {
    "msg": "Pong"
}

PLAY RECAP ****
52.179.126.20      : ok=3      changed=0      unreachable=0      failed=0      skipped=0      rescued=0      ignored=0
98.81.100.47      : ok=3      changed=0      unreachable=0      failed=0      skipped=0      rescued=0      ignored=0
root@ip-10-0-2-38:~/capstone-ansible#
```

Install Nginx on the remote aws azure app machine using ansible playbook.

```
root@ip-10-0-2-38:~/capstone-ansible# cat install-nginx.yml
---
- name: Install,enable and verify Nginx service
  hosts: all
  become: yes
  tasks:
    - name: Update apt cache
      apt:
        update_cache: yes

    - name: Install Nginx on the remote servers
      apt:
        name: nginx
        state: present

    - name: Ensure Nginx service is enabled to start at boot
      service:
        name: nginx
        state: started
        enabled: yes

    - name: Check Nginx service status
      command: systemctl is-active nginx
      register: nginx_status
      ignore_errors: yes

    - name: Display Nginx service status
      debug:
        msg: "Nginx service is {{ nginx_status.stdout }}"

root@ip-10-0-2-38:~/capstone-ansible#
```

Log output of the execution of the above playbook.

```
root@ip-10-0-2-38:~/capstone-ansible# ansible-playbook -i inventory.ini install-nginx.yml
PLAY [Install,enable and verify Nginx service] ****
TASK [Gathering Facts] ****
ok: [98.81.100.47]
ok: [52.179.126.20]

TASK [Update apt cache] ****
changed: [98.81.100.47]
changed: [52.179.126.20]

TASK [Install Nginx on the remote servers] ****
changed: [98.81.100.47]
changed: [52.179.126.20]

TASK [Ensure Nginx service is enabled to start at boot] ****
ok: [52.179.126.20]
ok: [98.81.100.47]

TASK [Check Nginx service status] ****
changed: [52.179.126.20]
changed: [98.81.100.47]

TASK [Display Nginx service status] ****
ok: [98.81.100.47] => {
    "msg": "Nginx service is active"
}
ok: [52.179.126.20] => {
    "msg": "Nginx service is active"
}

PLAY RECAP ****
52.179.126.20 : ok=6    changed=3    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0
98.81.100.47 : ok=6    changed=3    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0

root@ip-10-0-2-38:~/capstone-ansible#
```

Verify the nginx service running on aws app machine.

```
ubuntu@ip-10-0-1-248:~$ sudo su -
root@ip-10-0-1-248:~# systemctl status nginx
● nginx.service - A high performance web server and a reverse proxy server
  Loaded: loaded (/usr/lib/systemd/system/nginx.service; enabled; preset: enabled)
  Active: active (running) since Wed 2025-03-19 10:02:25 UTC; 2min 40s ago
    Docs: man:nginx(8)
 Process: 2304 ExecStartPre=/usr/sbin/nginx -t -q -g daemon on; master_process on; (code=exited, status=0/SUCCESS)
 Process: 2306 ExecStart=/usr/sbin/nginx -g daemon on; master_process on; (code=exited, status=0/SUCCESS)
Main PID: 2307 (nginx)
  Tasks: 2 (limit: 1130)
 Memory: 1.7M (peak: 1.9M)
    CPU: 11ms
   CGroup: /system.slice/nginx.service
           ├─2307 "nginx: master process /usr/sbin/nginx -g daemon on; master_process on;"
           └─2309 "nginx: worker process"

Mar 19 10:02:25 ip-10-0-1-248 systemd[1]: Starting nginx.service - A high performance web server and a reverse proxy server...
Mar 19 10:02:25 ip-10-0-1-248 systemd[1]: Started nginx.service - A high performance web server and a reverse proxy server.
root@ip-10-0-1-248:~# hostname
ip-10-0-1-248
root@ip-10-0-1-248:~#
```

Verify the nginx service running on azure app machine.

```
root@AzureAppMachine:~# systemctl status nginx
● nginx.service - A high performance web server and a reverse proxy server
  Loaded: loaded (/lib/systemd/system/nginx.service; enabled; vendor preset: enabled)
  Active: active (running) since Wed 2025-03-19 10:02:33 UTC; 2min 49s ago
    Docs: man:nginx(8)
 Process: 3040 ExecStartPre=/usr/sbin/nginx -t -q -g daemon on; master_process on; (code=exited, status=0/SUCCESS)
 Process: 3041 ExecStart=/usr/sbin/nginx -g daemon on; master_process on; (code=exited, status=0/SUCCESS)
Main PID: 3136 (nginx)
  Tasks: 3 (limit: 9519)
 Memory: 5.4M
    CPU: 32ms
   CGroup: /system.slice/nginx.service
           ├─3136 "nginx: master process /usr/sbin/nginx -g daemon on; master_process on;"
           ├─3138 "nginx: worker process"
           └─3139 "nginx: worker process"

Mar 19 10:02:33 AzureAppMachine systemd[1]: Starting A high performance web server and a reverse proxy server...
Mar 19 10:02:33 AzureAppMachine systemd[1]: Started A high performance web server and a reverse proxy server.
root@AzureAppMachine:~# hostname
AzureAppMachine
```

Verify the location of default nginx html file on aws app machine.

```
root@ip-10-0-1-248:~# cd /var/www/html/
root@ip-10-0-1-248:/var/www/html# ls -ltr
total 4
-rw-r--r-- 1 root root 615 Mar 19 10:02 index.nginx-debian.html
root@ip-10-0-1-248:/var/www/html#
```

Verify the location of default nginx html file on azure app machine.

```
root@AzureAppMachine:~# cd /var/www/html/
root@AzureAppMachine:/var/www/html# ls -ltr
total 4
-rw-r--r-- 1 root root 612 Mar 19 10:02 index.nginx-debian.html
root@AzureAppMachine:/var/www/html#
```

Verify the nginx web ui in the browser using aws public ip.

← → ⌂ Not Secure 98.81.100.47

Welcome to nginx!

If you see this page, the nginx web server is successfully installed and working. Further configuration is required.

For online documentation and support please refer to [nginx.org](#).
Commercial support is available at [nginx.com](#).

Thank you for using nginx.

Verify the nginx web ui in the browser using azure public ip.

← → ⌂ Not Secure 52.179.126.20

Welcome to nginx!

If you see this page, the nginx web server is successfully installed and working. Further configuration is required.

For online documentation and support please refer to [nginx.org](#).
Commercial support is available at [nginx.com](#).

Thank you for using nginx.

Task 3: Application Deployment

In this task, you will:

- Create two customised index.html files:
 - Create two index html files, index-aws.html and index-azure.html as a copy of the default nginx welcome html page.
 - index-aws.html: Change the content from "Welcome to Nginx" to "Welcome to AWS".
 - index-azure.html: Change the content from "Welcome to Nginx" to "Welcome to Azure".
- Modify the Ansible playbook to:
 - Copy the respective index-\${cloud}.html file to /var/www/html/ on the corresponding machine.
 - Update Nginx configuration to serve the new HTML page.

To verify the changes are applied, access the application via HTTP (IP:80) and confirm if:

- AWS instance displays: "Welcome to AWS".
- Azure instance displays: "Welcome to Azure".

Solution of Task 3 :

Create a custom nginx html file on the aws tools machine as per the below content.

```
root@ip-10-0-2-38:~/capstone-ansible# cat index-aws.html
<h1>Welcome to AWS</h1>
root@ip-10-0-2-38:~/capstone-ansible# cat index-azure.html
<h1>Welcome to Azure</h1>
root@ip-10-0-2-38:~/capstone-ansible#
```

Create the ansible playbook to copy the custom nginx html file to respective app machines and restart nginx.

```
- name: Update with custom index file on remote servers
  hosts: aws_nginx_app_vm
  become: yes
  tasks:
    - name: Copy the aws index-aws.html file to /var/www/html/
      copy:
        src: "/root/capstone-ansible/index-aws.html"
        dest: "/var/www/html/index-aws.html"
        owner: "root"
        group: "root"
        mode: "0644"
    tags:
      - aws_vm
    - name: Add index-aws.html in the /etc/nginx/sites-available/default under index section
      lineinfile:
        path: /etc/nginx/sites-available/default
        line: "index index-aws.html;"
        insertafter: "^.root /var/www/html.*$"

- name: Update with custom index file on remote servers
  hosts: azure_nginx_app_vm
  become: yes
  tasks:
    - name: Copy the azure index-azure.html file to /var/www/html/
      copy:
        src: "/root/capstone-ansible/index-azure.html"
        dest: "/var/www/html/index-azure.html"
        owner: "root"
        group: "root"
        mode: "0644"
    tags:
      - azure_vm

    - name: Add index-azure.html in the /etc/nginx/sites-available/default under index section
      lineinfile:
        path: /etc/nginx/sites-available/default
        line: "index index-azure.html;"
        insertafter: "^.root /var/www/html.*$"

- name: Restart Nginx service of remote servers
  hosts: all
  become: yes
  tasks:
    - name: Delete the existing default nginx file from remote server under /var/www/html/
      file:
        path: /var/www/html/index.nginx-debian.html
        state: absent

    - name: Restart Nginx service
      service:
        name: nginx
```

Ansible playbook execution log output screenshot.

```
root@ip-10-0-2-38:~/capstone-ansible# ansible-playbook -i inventory.ini update-nginx-webpage.yml
PLAY [Update with custom index file on remote servers] ****
TASK [Gathering Facts] ****
ok: [98.81.100.47]
TASK [Copy the aws index-aws.html file to /var/www/html/] ****
changed: [98.81.100.47]
TASK [Add index-aws.html in the /etc/nginx/sites-available/default under index section] ****
changed: [98.81.100.47]
PLAY [Update with custom index file on remote servers] ****
TASK [Gathering Facts] ****
ok: [52.179.126.20]
TASK [Copy the azure index-azure.html file to /var/www/html/] ****
changed: [52.179.126.20]
TASK [Add index-azure.html in the /etc/nginx/sites-available/default under index section] ****
changed: [52.179.126.20]
PLAY [Restart Nginx service of remote servers] ****
TASK [Gathering Facts] ****
ok: [98.81.100.47]
ok: [52.179.126.20]
TASK [Delete the existing default nginx file from remote server under /var/www/html/] ****
changed: [98.81.100.47]
changed: [52.179.126.20]
TASK [Restart Nginx service] ****
changed: [52.179.126.20]
changed: [98.81.100.47]
PLAY RECAP ****
52.179.126.20 : ok=6    changed=4    unreachable=0   failed=0    skipped=0   rescued=0   ignored=0   
98.81.100.47  : ok=6    changed=4    unreachable=0   failed=0    skipped=0   rescued=0   ignored=0
```

Verify the location of html file to confirm the custom nginx html is copied on aws app machine.

```
root@ip-10-0-1-248:/var/www/html# ls -ltr
total 4
-rw-r--r-- 1 root root 24 Mar 19 11:08 index-aws.html
root@ip-10-0-1-248:/var/www/html# hostname
ip-10-0-1-248
root@ip-10-0-1-248:/var/www/html#
```

Verify the location of html file to confirm the custom nginx html is copied on aws app machine.

```
root@AzureAppMachine:/var/www/html# ls -ltr
total 4
-rw-r--r-- 1 root root 26 Mar 19 11:08 index-azure.html
root@AzureAppMachine:/var/www/html# hostname
AzureAppMachine
root@AzureAppMachine:/var/www/html#
```

Verify the nginx config file to make sure the new index-aws.html entry is added on aws app machine.

```
root@ip-10-0-1-248:/var/www/html# cat /etc/nginx/sites-available/default
##
# You should look at the following URL's in order to grasp a solid understanding
# of Nginx configuration files in order to fully unleash the power of Nginx.
# https://www.nginx.com/resources/wiki/start/
# https://www.nginx.com/resources/wiki/start/topics/tutorials/config_pitfalls/
# https://wiki.debian.org/Nginx/DirectoryStructure
#
# In most cases, administrators will remove this file from sites-enabled/ and
# leave it as reference inside of sites-available where it will continue to be
# updated by the nginx packaging team.
#
# This file will automatically load configuration files provided by other
# applications, such as Drupal or Wordpress. These applications will be made
# available underneath a path with that package name, such as /drupal8.
#
# Please see /usr/share/doc/nginx-doc/examples/ for more detailed examples.
##


# Default server configuration
#
server {
    listen 80 default_server;
    listen [::]:80 default_server;

    # SSL configuration
    #
    # listen 443 ssl default_server;
    # listen [::]:443 ssl default_server;
    #
    # Note: You should disable gzip for SSL traffic.
    # See: https://bugs.debian.org/77332
    #
    # Read up on ssl_ciphers to ensure a secure configuration.
    # See: https://bugs.debian.org/765782
    #
    # Self signed certs generated by the ssl-cert package
    # Don't use them in a production server!
    #
    # include snippets/snakeoil.conf;

    root /var/www/html;
    index index-aws.html;

    # Add index.php to the list if you are using PHP
    index index.php index.html index.htm index.nginx-debian.html;

    server_name _;
```

Verify the nginx config file to make sure the new index-azure.html entry is added on azure app machine.

```

root@AzureAppMachine:/var/www/html# cat /etc/nginx/sites-available/default
##
# You should look at the following URL's in order to grasp a solid understanding
# of Nginx configuration files in order to fully unleash the power of Nginx.
# https://www.nginx.com/resources/wiki/start/
# https://www.nginx.com/resources/wiki/start/topics/tutorials/config_pitfalls/
# https://wiki.debian.org/Nginx/DirectoryStructure
#
# In most cases, administrators will remove this file from sites-enabled/ and
# leave it as reference inside of sites-available where it will continue to be
# updated by the nginx packaging team.
#
# This file will automatically load configuration files provided by other
# applications, such as Drupal or Wordpress. These applications will be made
# available underneath a path with that package name, such as /drupal8.
#
# Please see /usr/share/doc/nginx-doc/examples/ for more detailed examples.
##

# Default server configuration
#
server {
    listen 80 default_server;
    listen [::]:80 default_server;

    # SSL configuration
    #
    # listen 443 ssl default_server;
    # listen [::]:443 ssl default_server;
    #
    # Note: You should disable gzip for SSL traffic.
    # See: https://bugs.debian.org/773332
    #
    # Read up on ssl_ciphers to ensure a secure configuration.
    # See: https://bugs.debian.org/765782
    #
    # Self signed certs generated by the ssl-cert package
    # Don't use them in a production server!
    #
    # include snippets/snakeoil.conf;

    root /var/www/html;
    index index-azure.html;
    # Add index.php to the list if you are using PHP
    index index.html index.htm index.nginx-debian.html;
}

```

Verify the new nginx webpage in the browser using aws public IP



Welcome to AWS

Verify the new nginx webpage in the browser using azure public IP



Welcome to Azure

Task 4: Jenkins Setup for Continuous Deployment

In this task, you will:

- Install Jenkins on the Tools Machine (AWS).
- Set up a Jenkins declarative pipeline to:
 - Pull index-aws.html and index-azure.html from GitHub
 - Deploy the files to their respective locations (/var/www/html/)
 - Restart the Nginx service

To verify if the Jenkins pipeline runs correctly, modify web content in GitHub, trigger the Jenkins job, and check if the updates reflect correctly on AWS and Azure instances.

Solution of Task 4 :

Install Jenkins on the aws tools machine. Below are the steps for Jenkins installation.

```
#updating the system
sudo apt-get update -y && sudo apt-get upgrade -y

#Install Java
apt-get install -y wget unzip openjdk-17-jdk

#For the Debian package repository of Jenkins
sudo wget -O /usr/share/keyrings/jenkins-keyring.asc \
https://pkg.jenkins.io/debian-stable/jenkins.io-2023.key

#Add the Jenkins apt repository entry
echo deb [signed-by=/usr/share/keyrings/jenkins-keyring.asc] \
https://pkg.jenkins.io/debian-stable binary/ | sudo tee \
/etc/apt/sources.list.d/jenkins.list > /dev/null

#Update your local package index, then finally install Jenkins:
sudo apt-get update
sudo apt-get install jenkins
systemctl status jenkins

#Note down the initial password
cat /var/lib/jenkins/secrets/initialAdminPassword
```

Verify the Jenkins service running successfully or not.

```
root@ip-10-0-2-38:~# systemctl status jenkins
● jenkins.service - Jenkins Continuous Integration Server
   Loaded: loaded (/usr/lib/systemd/system/jenkins.service; enabled; preset: enabled)
     Active: active (running) since Wed 2025-03-19 11:19:41 UTC; 7s ago
       Main PID: 25782 (java)
          Tasks: 50 (limit: 4676)
         Memory: 579.6M (peak: 596.3M)
            CPU: 16.014s
           Group: /system.slice/jenkins.service
              └─25782 /usr/bin/java -Djava.awt.headless=true -jar /usr/share/java/jenkins.war --webroot=/var/cache/jenkins/war --httpPort=8080

Mar 19 11:19:38 ip-10-0-2-38 jenkins[25782]: 14d14f8c135047e8810d797d0c44d993
Mar 19 11:19:38 ip-10-0-2-38 jenkins[25782]: This may also be found at: /var/lib/jenkins/secrets/initialAdminPassword
Mar 19 11:19:38 ip-10-0-2-38 jenkins[25782]: ****
Mar 19 11:19:38 ip-10-0-2-38 jenkins[25782]: ****
Mar 19 11:19:38 ip-10-0-2-38 jenkins[25782]: ****
Mar 19 11:19:41 ip-10-0-2-38 jenkins[25782]: 2025-03-19 11:19:41.494+0000 [id=30]      INFO      jenkins.InitReactorRunner$1#onAttained: Completed initialization
Mar 19 11:19:41 ip-10-0-2-38 jenkins[25782]: 2025-03-19 11:19:41.517+0000 [id=23]      INFO      hudson.lifecycle.Lifecycle$OnReady: Jenkins is fully up and running
Mar 19 11:19:41 ip-10-0-2-38 systemd[1]: Started jenkins.service - Jenkins Continuous Integration Server.
Mar 19 11:19:41 ip-10-0-2-38 jenkins[25782]: 2025-03-19 11:19:41.803+0000 [id=48]      INFO      h.m.DownloadService$Downloadable#load: Obtained the updated data file for hudson.tasks.Maven.MavenInst...
Mar 19 11:19:41 ip-10-0-2-38 jenkins[25782]: 2025-03-19 11:19:41.803+0000 [id=48]      INFO      hudson.util.Retrier#start: Performed the action check updates server successfully at the attempt #1
root@ip-10-0-2-38:~#
root@ip-10-0-2-38:~#
```

Add port 8080 in the inbound rule of aws created security group for Jenkins UI.

Name	Security group rule ID	Type	Protocol	Port range	Source	Description
sgr-07cf7f6fb078ad3f4a	IPv4	Custom TCP	TCP	8080	0.0.0.0/0	Jenkins
-	IPv4	HTTP	TCP	80	0.0.0.0/0	-
-	IPv4	SSH	TCP	22	0.0.0.0/0	-

Create a github repo and create 2 custom nginx html files for aws and azure nginx.

Name	Last commit message	Last commit date
..		
index-aws.html	Added custom nginx webpages	3 days ago
index-azure.html	Added custom nginx webpages	3 days ago

Copy the aws tools machine private key from the /root/.ssh folder to Jenkins folder - /var/lib/Jenkins/.ssh
This will help Jenkins job to run from aws tools machine connecting to aws and azure app machine with passwordless mechanism.

```
root@ip-10-0-2-38:/var/lib/jenkins/.ssh# cp /root/.ssh/id_ed25519 .
root@ip-10-0-2-38:/var/lib/jenkins/.ssh# ls -ltr
total 4
-rw----- 1 root root 411 Mar 19 11:29 id_ed25519
root@ip-10-0-2-38:/var/lib/jenkins/.ssh#
root@ip-10-0-2-38:/var/lib/jenkins/.ssh# chown jenkins:jenkins id_ed25519
root@ip-10-0-2-38:/var/lib/jenkins/.ssh# ls -ltr
total 4
-rw----- 1 jenkins jenkins 411 Mar 19 11:29 id_ed25519
root@ip-10-0-2-38:/var/lib/jenkins/.ssh#
```

Jenkins job configuration –

The screenshot shows the Jenkins job configuration interface. At the top, there's a navigation bar with the Jenkins logo, a search icon, a shield icon with a red dot, an 'admin' dropdown, and a 'log out' link. Below the navigation, the breadcrumb path is 'Dashboard > nginx-custom-webpage-deployment > Configuration'. On the left, a sidebar titled 'Configure' lists 'General', 'Triggers', 'Pipeline', and 'Advanced', with 'General' currently selected. The main content area is titled 'General' and shows the job is 'Enabled' (indicated by a blue toggle switch). A 'Description' field contains the text: 'This job will deploy a custom webpage of nginx'. Below the description, there are several checkboxes: 'Discard old builds', 'Do not allow concurrent builds', 'Do not allow the pipeline to resume if the controller restarts', 'GitHub project', and 'Pipeline speed/durability override'.

Configure

Triggers

Set up automated actions that start your build based on specific events, like code changes or scheduled times.

 General Triggers Pipeline Advanced Build after other projects are built ? Build periodically ? GitHub hook trigger for GITScm polling ? Poll SCM ?

Schedule ?

*/1 * * * *

⚠️ Do you really mean "every minute" when you say "*/1 * * * *"? Perhaps you meant "H * * * *" to poll once per hour

Would last have run at Wednesday, March 19, 2025 at 12:24:34 PM Coordinated Universal Time; would next run at Wednesday, March 19, 2025 at 12:24:34 PM Coordinated Universal Time.

 Ignore post-commit hooks ? Trigger builds remotely (e.g., from scripts) ? Triggers Pipeline Advanced

Pipeline

Define your Pipeline using Groovy directly or pull it from source control.

Definition

Pipeline script

```
1 pipeline {
2     agent any // Runs on any available Jenkins agent
3
4     environment {
5         GITHUB_REPO = 'https://github.com/sauravsuman689/multicloud-nginx-capstone-project.git' // Replace with your G
6         BRANCH = 'main' // The branch to pull from, e.g., 'main' or 'master'
7         AWS_REMOTE_SERVER = '98.81.100.47' // The remote server hostname or IP address
8         AZURE_REMOTE_SERVER = '52.179.126.20'
9         SSH_USER = 'root' // SSH user with necessary permissions
10    }
11
12    stages {
13        stage('Checkout Code') {
14            steps {
15                script {
```

try sample Pipeline...

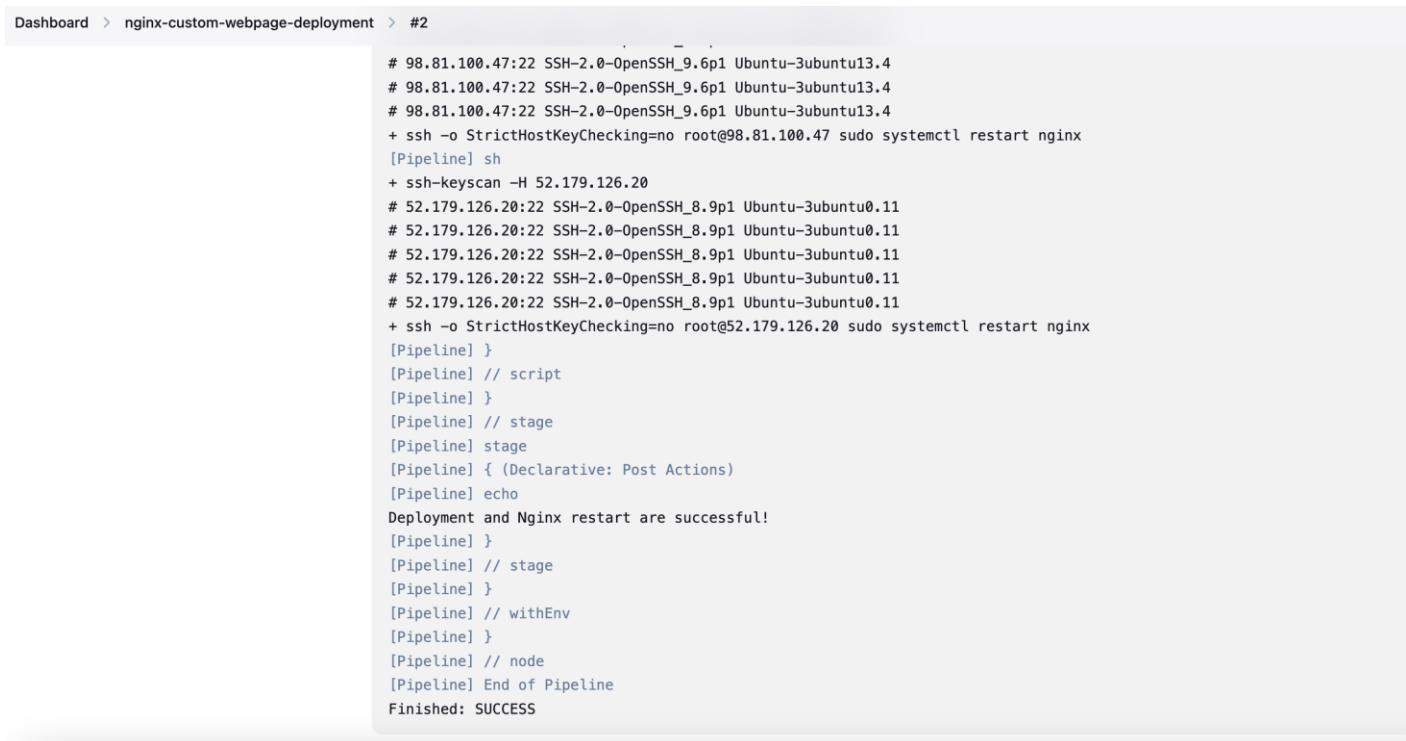
SaveApply

Build the Jenkins job.

The screenshot shows the Jenkins Pipeline configuration page for the job 'nginx-custom-webpage-deployment'. The left sidebar contains navigation links: Status, Changes, Build Now, Configure, Delete Pipeline, Stages, Rename, Pipeline Syntax, and GitHub Hook Log. The main content area displays the pipeline configuration with a green checkmark icon and the title 'nginx-custom-webpage-deployment'. A note states: 'This job will deploy a custom webpage of nginx'. Below this is a section titled 'Permalinks' with a bulleted list of build history: Last build (#2), Last stable build (#2), Last successful build (#2), Last failed build (#1), Last unsuccessful build (#1), and Last completed build (#2). The bottom part of the page shows the 'Builds' section with a table for build #2, showing it was started by user 'admin' at 11:38 on March 19, 2025, and took 4.8 seconds.

The screenshot shows the Jenkins build details page for build #2. The top navigation bar includes the Jenkins logo, search, notifications, user 'admin', and log out. The main content area shows the build summary: '#2 (19 Mar 2025, 11:38:41)'. It includes links to Add description and Keep this build forever. The left sidebar lists build-related actions: Status, Changes, Console Output, Edit Build Information, Delete build '#2', Timings, Git Build Data, Pipeline Overview, Pipeline Console, and Restart from Stage. The right side provides details about the build: Started by user 'admin' at 11:38 on March 19, 2025, and took 4.8 seconds. It also shows the git revision and repository information: Revision: bf032ddf59a8950b8bf406726457c1537056632f, Repository: <https://github.com/sauravsuman689/multicloud-nginx-capstone-project.git>, and a note that there were no changes.

Verify the console output of the Jenkins pipeline job.



```
Dashboard > nginx-custom-webpage-deployment > #2

# 98.81.100.47:22 SSH-2.0-OpenSSH_9.6p1 Ubuntu-3ubuntu13.4
# 98.81.100.47:22 SSH-2.0-OpenSSH_9.6p1 Ubuntu-3ubuntu13.4
# 98.81.100.47:22 SSH-2.0-OpenSSH_9.6p1 Ubuntu-3ubuntu13.4
+ ssh -o StrictHostKeyChecking=no root@98.81.100.47 sudo systemctl restart nginx
[Pipeline] sh
+ ssh-keyscan -H 52.179.126.20
# 52.179.126.20:22 SSH-2.0-OpenSSH_8.9p1 Ubuntu-3ubuntu0.11
+ ssh -o StrictHostKeyChecking=no root@52.179.126.20 sudo systemctl restart nginx
[Pipeline] }
[Pipeline] // script
[Pipeline] }
[Pipeline] // stage
[Pipeline] stage
[Pipeline] { (Declarative: Post Actions)
[Pipeline] echo
Deployment and Nginx restart are successful!
[Pipeline] }
[Pipeline] // stage
[Pipeline] }
[Pipeline] // withEnv
[Pipeline] }
[Pipeline] // node
[Pipeline] End of Pipeline
Finished: SUCCESS
```

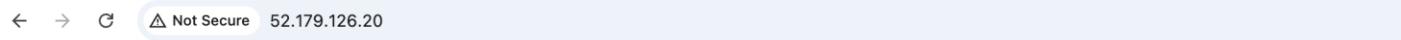
Verify the new nginx webpage deployed by Jenkins cicd job in the browser using aws public IP.



Welcome to AWS

This webpage is deployed using Jenkins CICD

Verify the new nginx webpage deployed by Jenkins cicd job in the browser using azure public IP.



Welcome to Azure

This webpage is deployed using Jenkins CICD

Update the github repo html files with some code change and verify the job triggered to update the webpages.

The screenshot shows the Jenkins interface for a pipeline named "nginx-custom-webpage-deployment". The build number is "#3". On the left, there's a sidebar with various options like Status, Changes, Console Output (which is selected), Edit Build Information, Delete build '#3', Timings, Git Build Data, Pipeline Overview, Pipeline Console, Restart from Stage, Replay, Pipeline Steps, and Workspaces. The main area is titled "Console Output" and contains the following log output:

```
Started by user admin
[Pipeline] Start of Pipeline
[Pipeline] node
Running on Jenkins in /var/lib/jenkins/workspace/nginx-custom-webpage-deployment
[Pipeline] {
[Pipeline] withEnv
[Pipeline] {
[Pipeline] stage
[Pipeline] {
  (Checkout Code)
[Pipeline] script
[Pipeline] {
[Pipeline] git
The recommended git tool is: NONE
No credentials specified
> git rev-parse --resolve-git-dir /var/lib/jenkins/workspace/nginx-custom-webpage-deployment/.git # timeout=10
Fetching changes from the remote Git repository
> git config remote.origin.url https://github.com/sauravsuman689/multicloud-nginx-capstone-project.git # timeout=10
Fetching upstream changes from https://github.com/sauravsuman689/multicloud-nginx-capstone-project.git
> git --version # timeout=10
```

Again, verify the updated nginx webpage deployed by Jenkins cicd job in the browser using aws public IP.

← → ⌂ △ Not Secure 98.81.100.47

Welcome to AWS

This webpage is deployed using Jenkins CICD and This is the 2nd code change.

Again, verify the updated nginx webpage deployed by Jenkins cicd job in the browser using azure public IP.

← → ⌂ △ Not Secure 52.179.126.20

Welcome to Azure

This webpage is deployed using Jenkins CICD and This is the 2nd code change.

Task 5: Traffic Management Using AWS Route 53

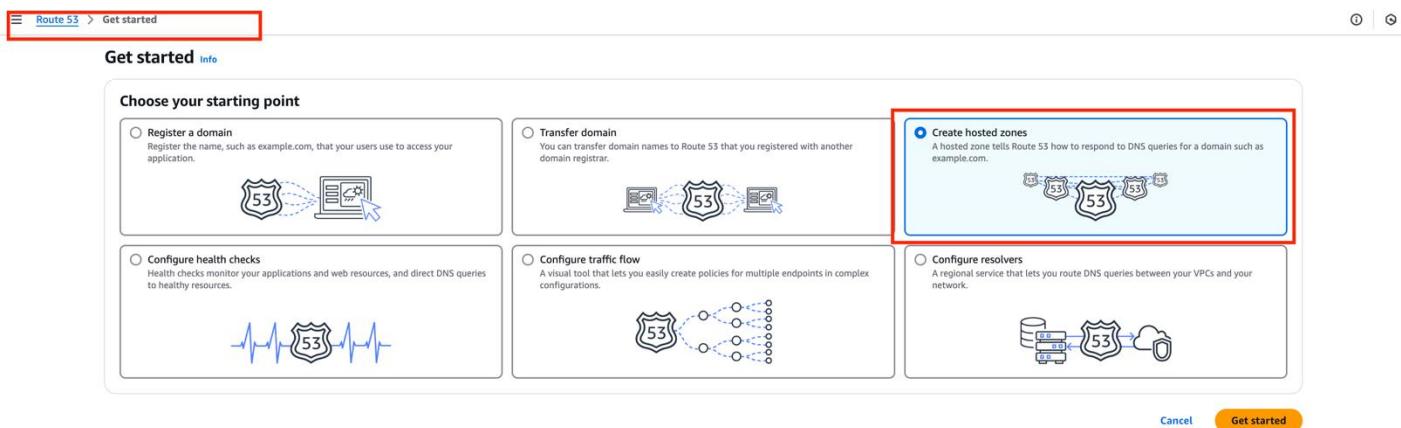
In this task, you will:

- Create a hosted zone upgrad.com in AWS Route 53.
- Add a DNS entry for app.upgrad.com pointing to the AWS App Machine's public IP.
- Implement a failover policy:
 - Route 53 directs traffic to the AWS App Machine if the AWS instance is healthy.
 - If the AWS instance fails, Route 53 redirects traffic to the Azure VM's public IP.

To verify if you have done this task correctly, simulate a failure by stopping the AWS instance and confirm that traffic is redirected to Azure. Restore the AWS instance and confirm traffic returns to it.

Solution of Task 5 :

Create a hosted zone in aws route53 as per the below screenshot.



Route 53 < Public upgrad.com Info

Hosted zone details

Records (2) DNSSEC signing Hosted zone tags (0)

Records (2) Info

Record name	Type	Routing policy	Alias	Value/Route traffic to	TTL (s)	Health check ID	Evaluation time	Record ID
upgrad.com	NS	Simple	-	No ns-1272.awsdns-31.org, ns-494.awsdns-61.com, ns-2015.awsdns-59.co.uk, ns-802.awsdns-36.net.	172800	-	-	-
upgrad.com	SOA	Simple	-	No ns-1272.awsdns-31.org, aws... 900	900	-	-	-

Define a health check for both the aws and azure nginx app endpoints.

Route 53 < Health checks

Health checks (2) Info

ID	Name	Details	Status in last 24 hours	Current state	Alarm	State	Actions
1ea27ca9-bba9-4f76-8cc0-b...	aws-app-health-check	http://98.81.100.47:80/	Healthy	Green Healthy	None, Create alarm	Enabled	⋮
6953f93d-30dc-4371-8385-...	azure-app-health-check	http://52.179.126.20:80/	Healthy	Green Healthy	None, Create alarm	Enabled	⋮

Create a failover policy in the hosted zone for nginx deployed on both aws and azure app machines where aws being the primary dns and azure as secondary dns.

Route 53 < Public upgrad.com Info

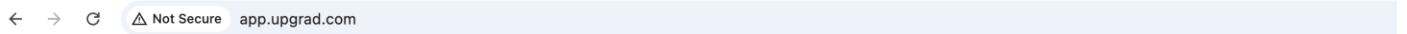
Hosted zone details

Records (4) DNSSEC signing Hosted zone tags (0)

Records (4) Info

Record name	Type	Routing policy	Primary	Secondary	Value/Route traffic to	TTL (s)	Health check ID	Evaluation time	Record ID
upgrad.com	NS	Simple	-	No	ns-1272.awsdns-31.org, ns-494.awsdns-61.com, ns-2015.awsdns-59.co.uk, ns-802.awsdns-36.net.	172800	-	-	
upgrad.com	SOA	Simple	-	No	ns-1272.awsdns-31.org, aws... 900	900	-	-	
app.upgrad.com	A	Failover	Primary	No	98.81.100.47 300 1ea27ca9-bba9-4f76-8cc0-b...	300	1ea27ca9-bba9-4f76-8cc0-b...	1	
app.upgrad.com	A	Failover	Secondary	No	52.179.126.20 300 6953f93d-30dc-4371-8385-...	300	6953f93d-30dc-4371-8385-...	2	

Verify the nginx webpage using the subdomain – **app.upgrad.com** in the browser which points to primary aws nginx webpage.



Welcome to AWS

This webpage is deployed using Jenkins CI/CD and This is the 2nd code change.

Stopping aws instance –

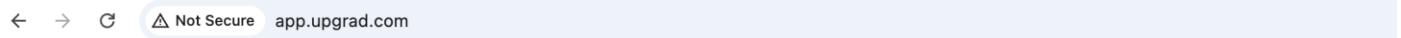
Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IPv4 DNS	Public IPv4 ...
Tools Machine	i-02ced6c5731a6bf30	Running	t2.medium	2/2 checks passed	View alarms +	us-east-1b	ec2-44-202-103-139.co...	44.202.103.139
App Machine	i-004a5039470b5fa2a	Stopping	t2.micro	-	View alarms +	us-east-1a	ec2-98-81-100-47.com...	98.81.100.47

Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IPv4 DNS	Public IPv4 ...
Tools Machine	i-02ced6c5731a6bf30	Running	t2.medium	2/2 checks passed	View alarms +	us-east-1b	ec2-44-202-103-139.co...	44.202.103.139
App Machine	i-004a5039470b5fa2a	Stopped	t2.micro	-	View alarms +	us-east-1a	-	-

Verify the health check after stopping aws instance.

ID	Name	Details	Status in last 24 hours	Current s...	Alarm	State	Actions
1ea27ca9-bba9-4f76-8cc0-b...	aws-app-health-check	http://98.81.100.47:80/	Unhealthy	None, Create alarm	Enabled	Unhealthy	Edit
6953f93d-30dc-4371-8385-...	azure-app-health-check	http://52.179.126.20:80/	Healthy	None, Create alarm	Enabled	Healthy	Edit

Again, Verify the nginx webpage using the subdomain – app.upgrad.com in the browser which points to secondary dns which is azure nginx webpage confirming that traffic is redirected to azure as primary aws instance is down.



Welcome to Azure

This webpage is deployed using Jenkins CICD and This is the 2nd code change.

Once aws vm comes back healthy, traffic is diverted back to AWS.

A screenshot of the AWS EC2 Instances page. The left sidebar shows 'EC2 > Instances'. The main table lists two instances: 'Tools Machine' (Instance ID: i-02ced6c5731a6bf30, Status: Running, Type: t2.medium) and 'App Machine' (Instance ID: i-004a5039470b5fa2a, Status: Running, Type: t2.micro). Both instances have 2/2 checks passed. The table includes columns for Name, Instance ID, Instance state, Instance type, Status check, Alarm status, Availability Zone, and Public IPv4 DNS. A search bar at the top left says 'Find Instance by attribute or tag (case-sensitive)' and an 'All states' dropdown is shown. Action buttons like 'Connect', 'Actions', and 'Launch instances' are at the top right.

Welcome to AWS

This webpage is deployed using Jenkins CICD and This is the 2nd code change.

NOTE - In the above scenario, we have to make sure aws elasticIP is assigned to the both the app vms so that after reboot the ip addresses of the vms don't change.

Thank You..!