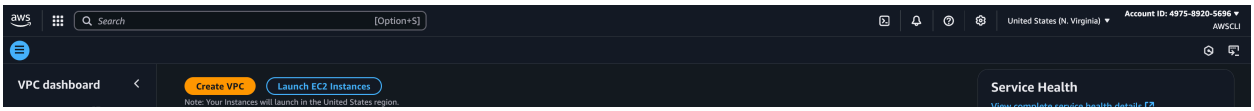
create your own Network architecture with at least 3 public Subnets and 6 private subnets.
class review: Week 4 - Class 7 ZION: Yacht Party OCT 04

Dennis' Network architecture

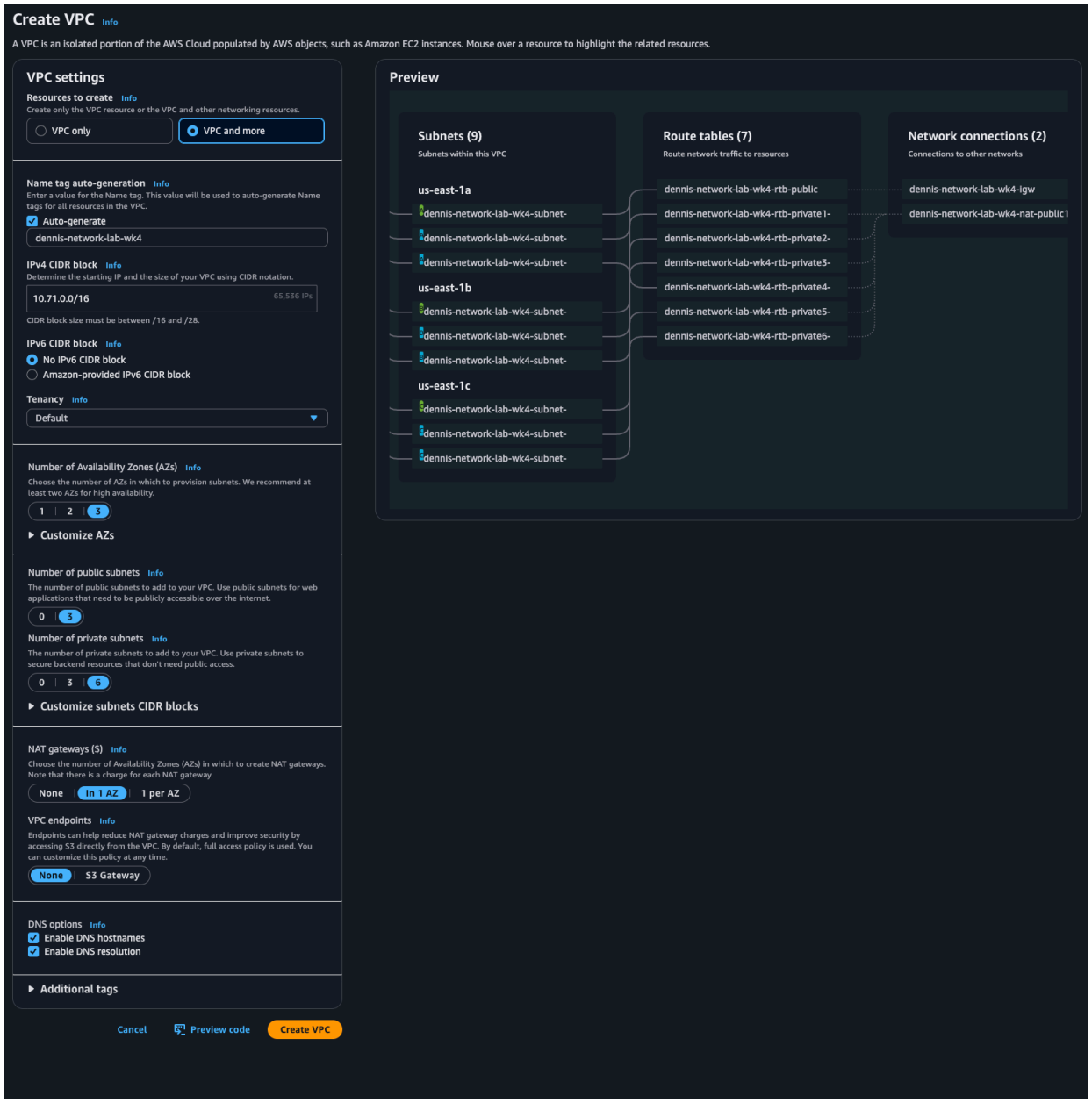
region = virginia (us-east-1)	
	10.71.0.0/16

public	10.71.1.0/24	a
public	10.71.2.0/24	b
public	10.71.3.0/24	c
public	10.71.4.0/24	d
private	10.71.11.0/24	a
private	10.71.12.0/24	b
private	10.71.13.0/24	c
private	10.71.14.0/24	d
private	10.71.21.0/24	a
private	10.71.22.0/24	b
private	10.71.23.0/24	c
private	10.71.24..0/24	d

Go to AWS and create a VPC



name it
fill in IPv4 CIDR block
number of availability zones - 3
number of public subnets - 3
number of private subnets - 6
Customized subnets CIDR blocks - fill out using my architecture
NAT gateways - In 1 AZ
VPC endpoints - None



Create VPC

Create VPC workflow

Success

Details

- Create VPC: vpc-08f5452bdde6ac295
- Enable DNS hostnames
- Enable DNS resolution
- Verifying VPC creation: vpc-08f5452bdde6ac295
- Create subnet: subnet-06fa59f97d0a6ae0e
- Create subnet: subnet-0ffbbae0d70267554
- Create subnet: subnet-0011a6029d6343f58
- Create subnet: subnet-0e096a0212e1bcd28
- Create subnet: subnet-082f2a2d10fc66f8e
- Create subnet: subnet-0054791c7ec1adbc0
- Create subnet: subnet-07286747a83a1b411
- Create subnet: subnet-0401aadf496cbe3dd
- Create subnet: subnet-082e4416ce2ca2959
- Create internet gateway: igw-0165f1d962abe7b09
- Attach internet gateway to the VPC
- Create route table: rtb-0ae14dd8c35aa667a
- Create route
- Associate route table
- Associate route table
- Associate route table
- Allocate elastic IP: eipalloc-0d0e86da9f3661146
- Create NAT gateway: nat-0c354add33ba45ca1
- Wait for NAT Gateways to activate
- Create route table: rtb-0507274f20ace8f0c
- Create route
- Associate route table
- Create route table: rtb-06261557155fc5d24
- Create route
- Associate route table
- Create route table: rtb-02a9c01c899f7c6bf
- Create route
- Associate route table
- Create route table: rtb-0231d497b555121e8
- Create route
- Associate route table
- Create route table: rtb-0027d50a3db08339c
- Create route
- Associate route table
- Create route table: rtb-09d12a55da4f9e509
- Create route
- Associate route table
- Verifying route table creation

View VPC

vpc-08f5452bdde6ac295 / dennis-network-lab-wk4-vpc

Actions

Details Info

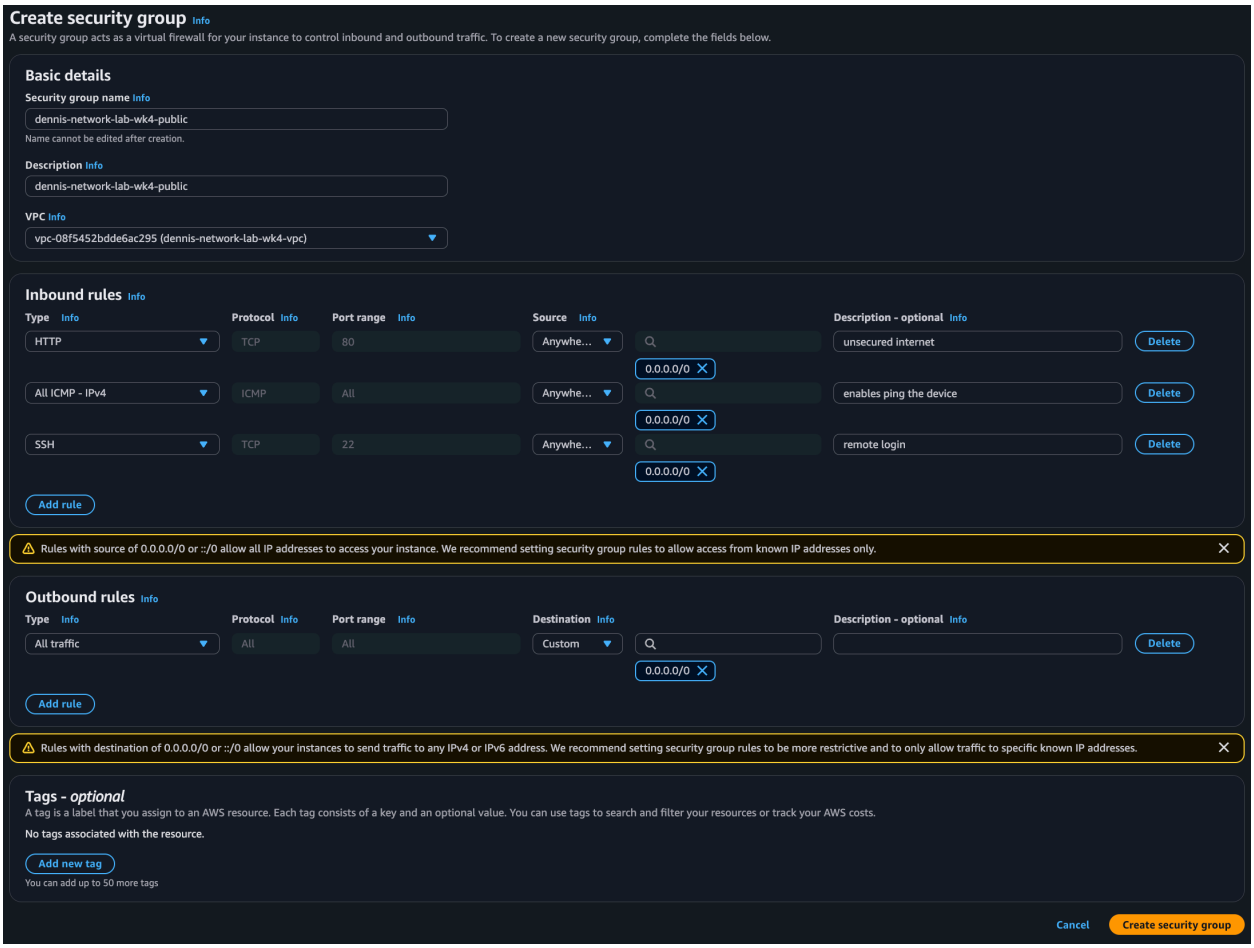
VPC ID vpc-08f5452bdde6ac295	**State** Available	**Block Public Access** Off	**DNS hostnames** Enabled
DNS resolution Enabled	**Tenancy** default	**DHCP option set** dopt-0f21e38a07d7727ba	**Main route table** rtb-0fd1fbf0180ff77cf
Main network ACL acl-0b69ed1d5a78f5e78	**Default VPC** No	**IPv4 CIDR** 10.71.0.0/16	**IPv6 pool** –
IPv6 CIDR (Network border group) –	**Network Address Usage metrics** Disabled	**Route 53 Resolver DNS Firewall rule groups** –	**Owner ID** 497589205696

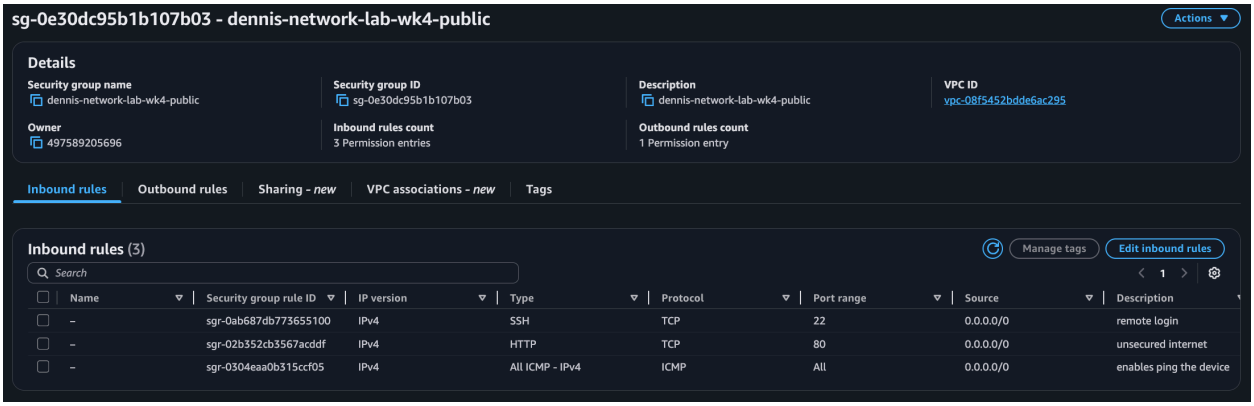
Resource map | CIDRs | Flow logs | Tags | Integrations

Resource map Info

Show all details

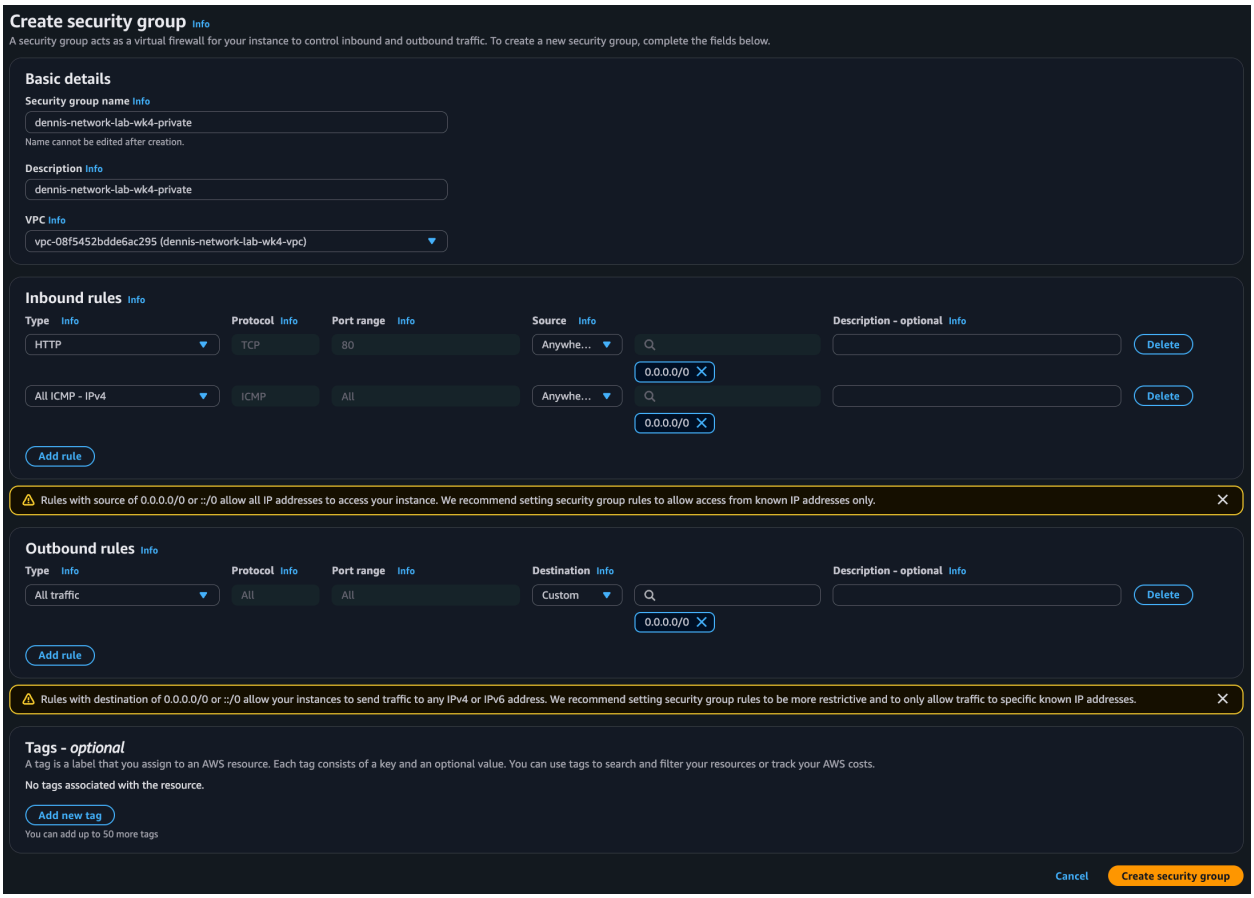
View VPC
Make an EC2 instance
first create public security group
name it and *put public in the title
add a description
choose VPC created
Inbound rules: HTTP, SSH, and All ICMP - IPv4 *so we can ping the device later

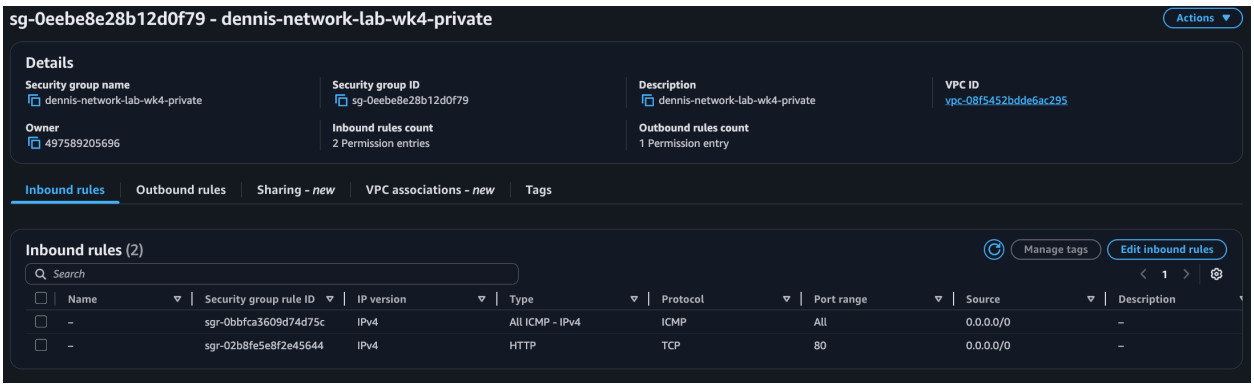




Outbound rules : DON'T TOUCH

create private security group

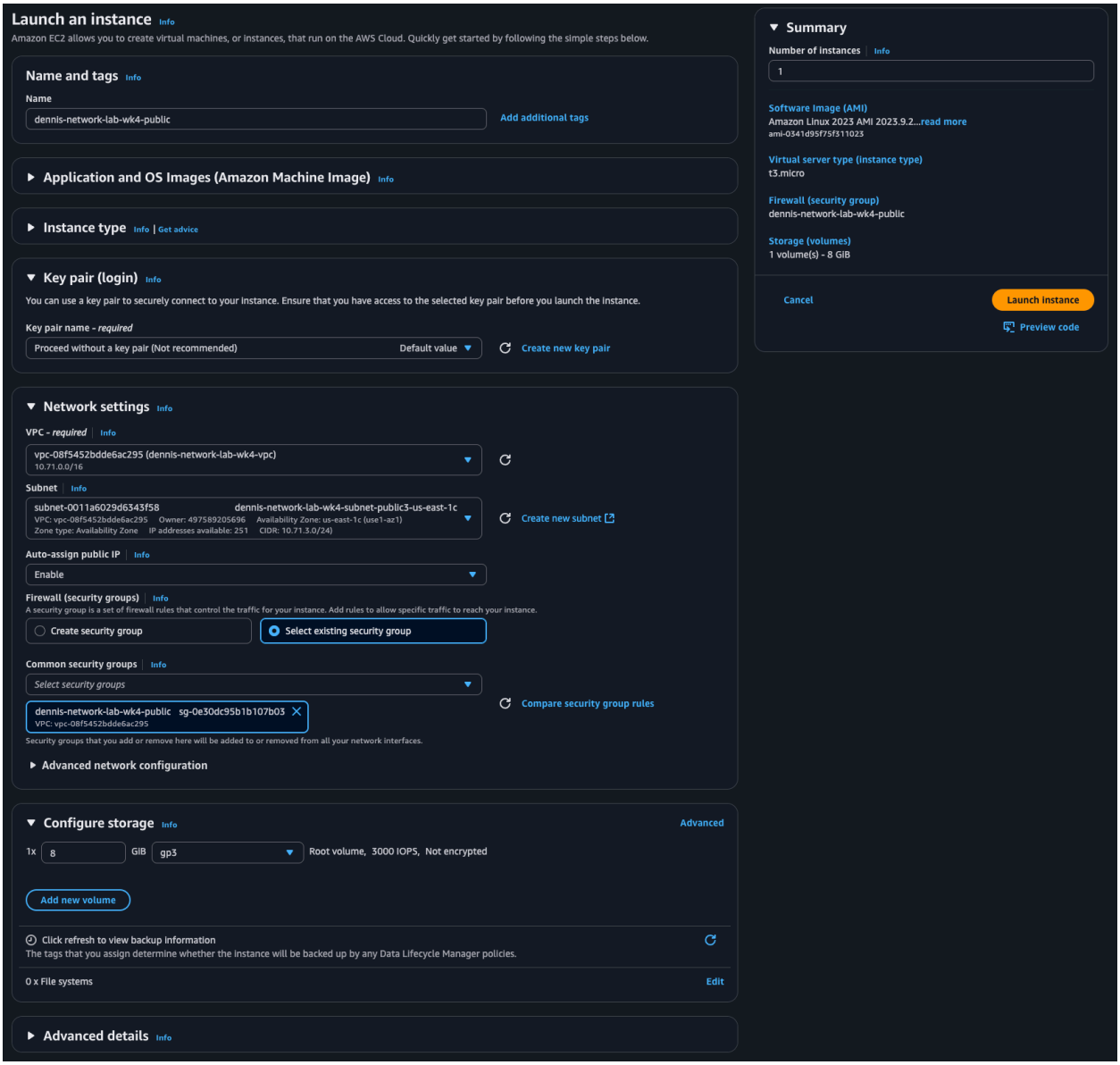




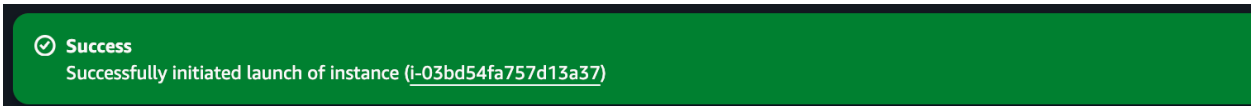
Add inbound rules: HTTP and All ICMP - IPv4

create instance
go to EC2 - instance
launch instance

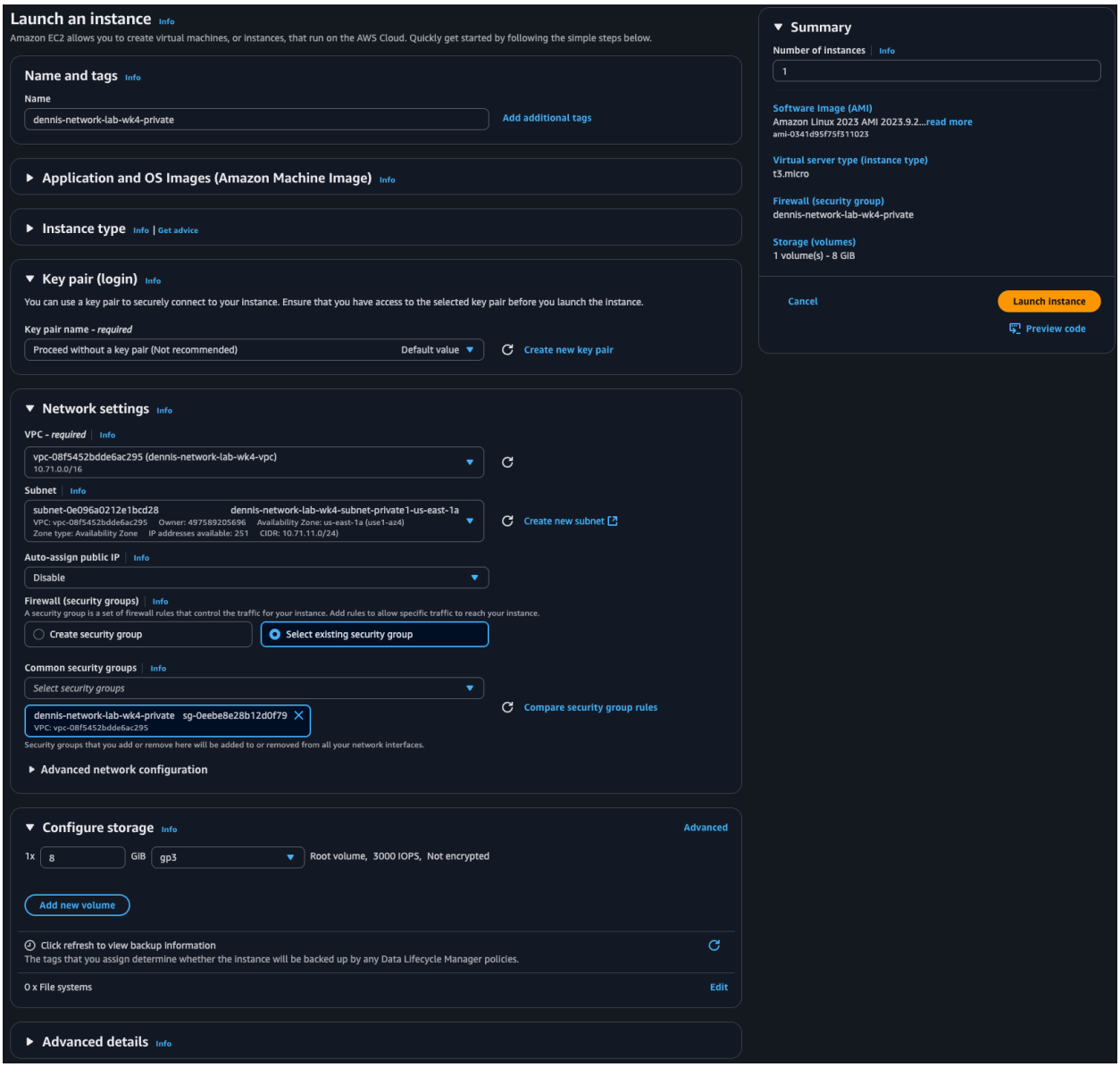
name - put public in the name of
Key pair - use the default (Proceed without a key pair...)
Network settings - Edit
VPC - choose the one created
Auto-assign public IP - enable
Firewall - select existing security group
common security groups - select the public one created
Advance details - scroll to the bottom and add script in the user data



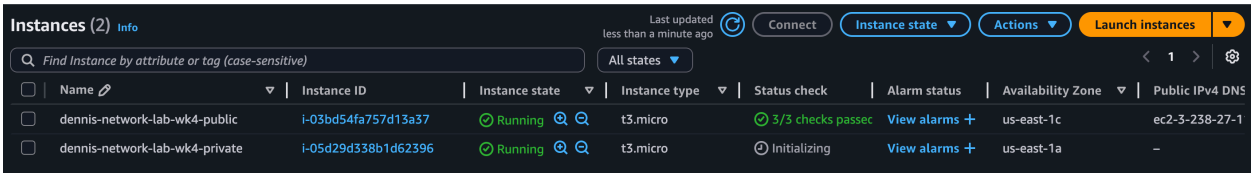
Launch Instance



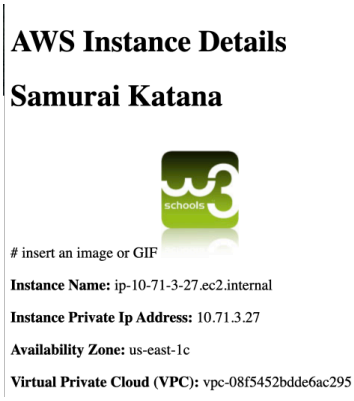
launch a private instance
name - private in the name
Network settings/subnet choose private subnet
auto-assign public IP - disable
common security groups - choose the created private group



launch instance

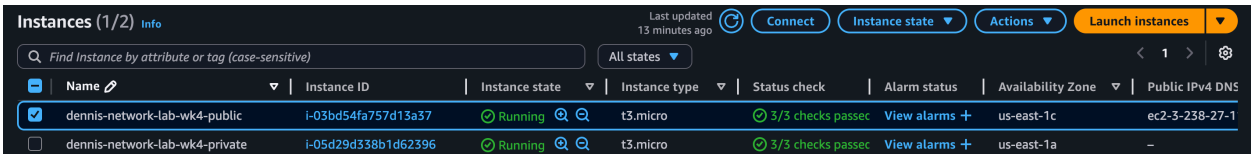


go to public instance, copy public DNC and past with http://

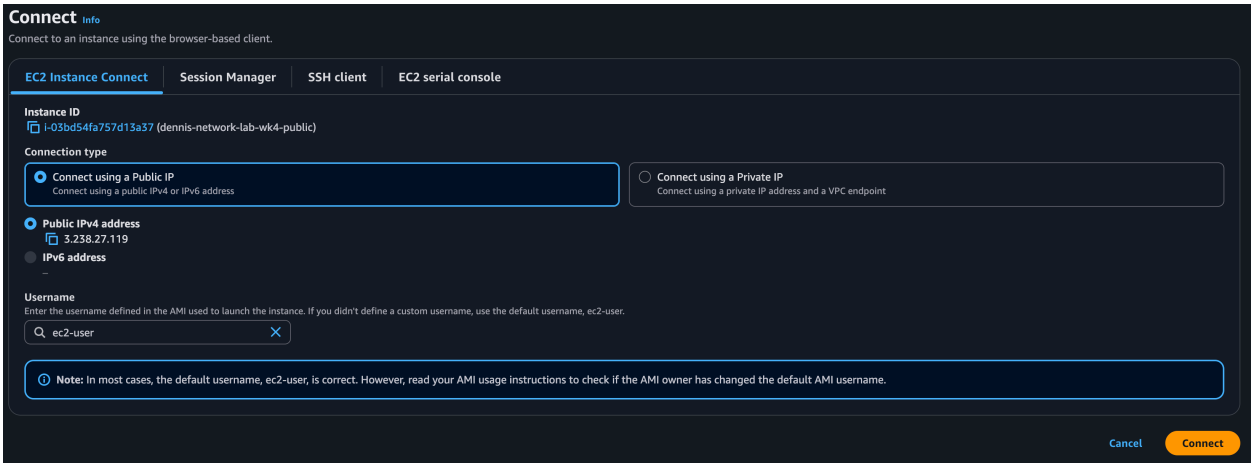


to check the private instance go to private instance and copy the private IPv4 address

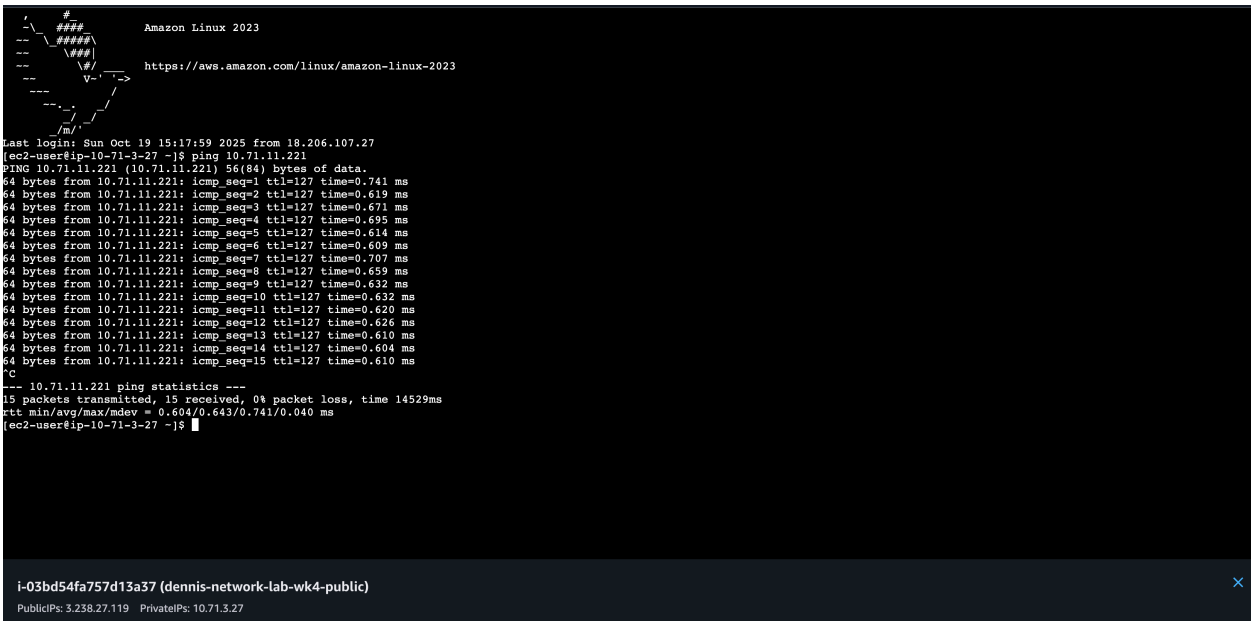
go back to instance choose public instance and connect



taken to the connect page, leave all the defaults and hit connect



type my private IPv4 address that I copied and type it with ping in the front (ping 10.71.11.221) *if you don't want your pings to be infinite you can click "control+c" to stop or choose how the amount of pings ie. "ping -c 10 10.71.11.221" I've chosen 10 pings



Everything looks good!

Now tear down *in reverse to how it was created

Terminate EC2 Instances

Go to EC2 Dashboard → Instances.
Select and terminate all instances in your VPC.
Delete NAT Gateway

Go to VPC Dashboard → NAT Gateways.
Delete the NAT Gateway.
Release the associated Elastic IP (in EC2 Dashboard → Elastic IPs).
Delete Subnets

Go to VPC Dashboard → Subnets.
Delete all subnets in your VPC (public and private).
Detach and Delete Internet Gateway

Go to VPC Dashboard → Internet Gateways.
Detach the Internet Gateway from your VPC.
Delete the Internet Gateway.
Delete Route Tables

Go to VPC Dashboard → Route Tables.
Delete any custom route tables (not the main route table).
Delete Security Groups

Go to VPC Dashboard → Security Groups.
Delete any custom security groups (default cannot be deleted until VPC is deleted).
Delete the VPC

Go to VPC Dashboard → Your VPCs.
Delete the VPC.