

AWS

EC2 & VPC CRASH COURSE

WHITNEY CHAMPION

BEFORE WE START

- Prereqs
 - AWS account
 - SSH client
 - Mac - Terminal, Royal TSX, Termius
 - Windows - mRemote
 - <https://mremoteng.org/download>
 - <http://www.chiark.greenend.org.uk/~sgtatham/putty/latest.html>
 - <http://technotes.khitrenovich.com/opening-ssh-aws-hosted-linux-servers-mremoteng/>
 - Or spin up a Linux VM with VirtualBox
 - Linux - ...

AWS HAS A LOT OF STUFF

The screenshot displays the AWS Management Console's 'Services' page. At the top, there is a navigation bar with the AWS logo, 'Services', and 'Resource Groups'. Below this is a search bar labeled 'Search services' and a 'Group A-Z' button. The main content area is organized into a grid of service categories, each with an icon and a list of sub-services. On the left side, there is a 'History' sidebar with links to EC2, RDS, Connect Home, Billing, S3, and VPC.

History

- EC2
- RDS
- Connect Home
- Billing
- S3
- VPC

Search services Group A-Z

- Compute**
 - EC2
 - EC2 Container Service
 - Lightsail
 - Elastic Beanstalk
 - Lambda
 - Batch
- Storage**
 - S3
 - EFS
 - Glacier
 - Storage Gateway
- Database**
 - RDS
 - DynamoDB
 - ElastiCache
 - Redshift
- Networking & Content Delivery**
 - VPC
 - CloudFront
 - Direct Connect
 - Route 53
- Migration**
 - DMS
 - Server Migration
 - Snowball
- Developer Tools**
 - CodeCommit
 - CodeBuild
 - CodeDeploy
 - CodePipeline
- Management Tools**
 - CloudWatch
 - CloudFormation
 - CloudTrail
 - Config
 - OpsWorks
 - Service Catalog
 - Trusted Advisor
 - Managed Services
 - Application Discovery Service
- Security, Identity & Compliance**
 - IAM
 - Inspector
 - Certificate Manager
 - Directory Service
 - WAF & Shield
 - Compliance Reports
- Analytics**
 - Athena
 - EMR
 - CloudSearch
 - Elasticsearch Service
 - Kinesis
 - Data Pipeline
 - QuickSight
- Artificial Intelligence**
 - Lex
 - Polly
 - Rekognition
 - Machine Learning
- Internet of Things**
 - AWS IoT
- Game Development**
 - GameLift
- Mobile Services**
 - Mobile Hub
 - Cognito
 - Device Farm
 - Mobile Analytics
 - Pinpoint
- Application Services**
 - Step Functions
 - SWF
 - API Gateway
 - Elastic Transcoder
- Messaging**
 - SQS
 - SNS
 - SES
- Business Productivity**
 - WorkDocs
 - WorkMail
- Desktop & App Streaming**
 - WorkSpaces
 - AppStream 2.0

TODAY...

We will focus on **EC2** and **VPC**.

WHAT IS EC2?

Amazon Elastic Compute Cloud (EC2) is the Amazon Web Service you use to create and run virtual machines in the cloud.

These are called **instances**.

TODAY...

We will launch **EC2 instances** in
your very own **VPC**.

YOUR VPC

if you have an AWS account...
YOU PROBABLY ALREADY HAVE ONE!

The screenshot displays the AWS Management Console interface for the VPC Dashboard. The top navigation bar includes 'Services', 'Resource Groups', and a user profile icon. The left sidebar lists various VPC-related services such as Subnets, Route Tables, Internet Gateways, and Security Groups. The main content area features a 'Create VPC' button and a search bar. Below the search bar is a table listing VPCs. The table has columns for Name, VPC ID, State, IPv4 CIDR, IPv6 CIDR, DHCP options set, Route table, Network ACL, Tenancy, and Default VPC. One VPC is listed: 'Default' with VPC ID 'vpc-024e9066', State 'available', IPv4 CIDR '10.0.0.0/16', DHCP options set 'dopt-1ad0c578', Route table 'rtb-7e09871a', Network ACL 'acl-0ee4c25a', Tenancy 'Default', and Default VPC 'No'. Below the table, the details for the selected VPC 'vpc-024e9066 | Default' are shown, including tabs for Summary, Flow Logs, and Tags. The Summary tab displays key attributes: VPC ID, State, Network ACL, Tenancy, IPv4 CIDR, DNS resolution, IPv6 CIDR, DNS hostnames, DHCP options set, and ClassicLink DNS Support.

Name	VPC ID	State	IPv4 CIDR	IPv6 CIDR	DHCP options set	Route table	Network ACL	Tenancy	Default VPC
Default	vpc-024e9066	available	10.0.0.0/16		dopt-1ad0c578	rtb-7e09871a	acl-0ee4c25a	Default	No

vpc-024e9066 | Default

Summary | Flow Logs | Tags

VPC ID: vpc-024e9066 | Default
State: available
IPv4 CIDR: 10.0.0.0/16
IPv6 CIDR:
DHCP options set: dopt-1ad0c578
Route table: rtb-7e09871a

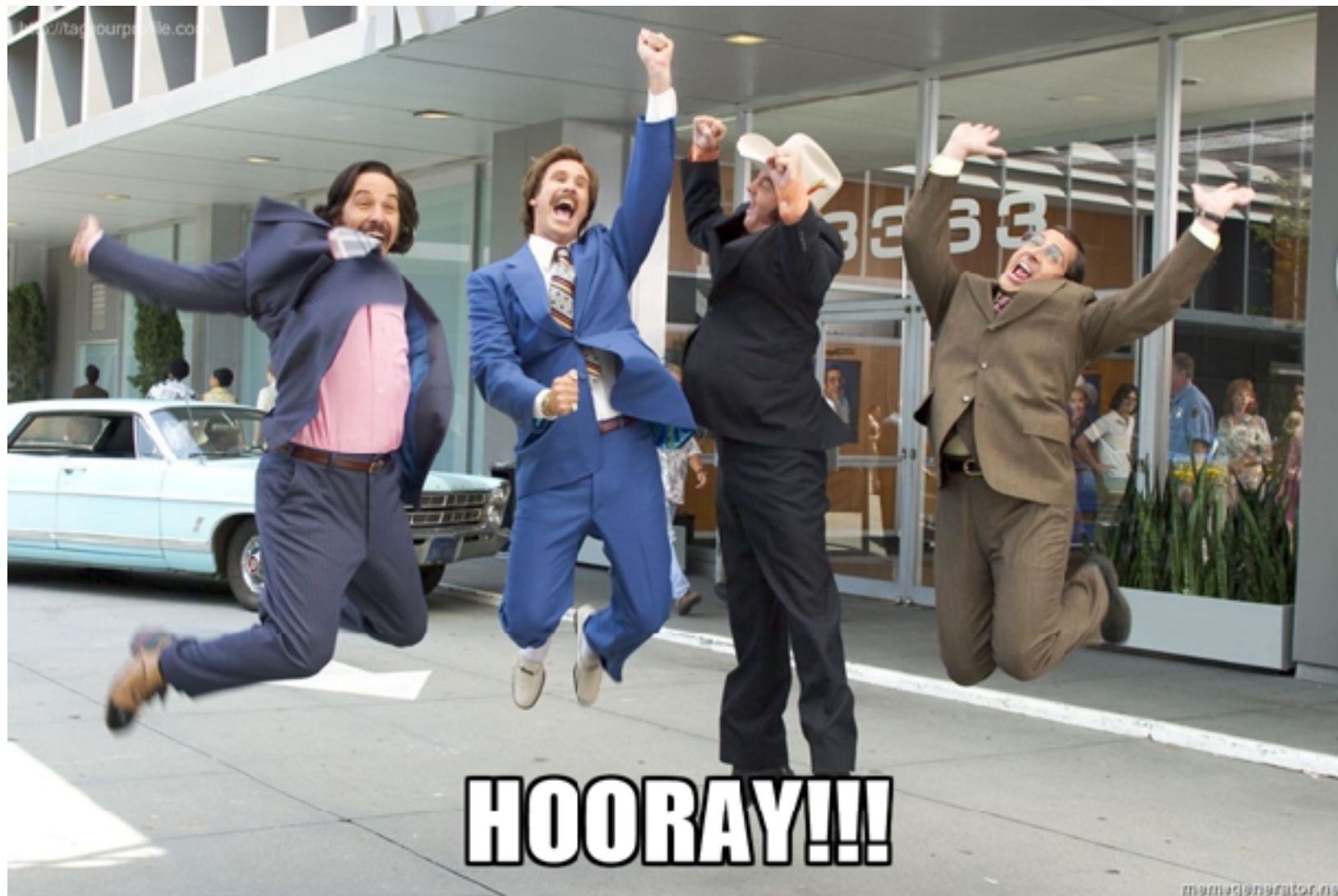
Network ACL: acl-0ee4c25a
Tenancy: Default
DNS resolution: yes
DNS hostnames: yes
ClassicLink DNS Support: no

BENEFITS OF A VPC

- Multiple network interfaces / IP addresses per instance
- You pick your IP address ranges
- Internal load balancers
- DHCP option sets
- Option to use single-tenant hardware
- Probably a dozen more benefits
- **Proper network segmentation**
- **Security**

LET'S MAKE ONE

- Even if you already have one
- Because it is that much fun




I PUT ON MY ROBE AND WIZARD HAT

The screenshot shows the AWS VPC Dashboard. At the top, there is a navigation bar with the AWS logo, 'AWS', 'Services', and 'Edit' dropdown menus. On the left, the 'VPC Dashboard' sidebar includes a 'Filter by VPC' dropdown set to 'None' and a list of VPC resources: Virtual Private Cloud, Your VPCs, Subnets, Route Tables, Internet Gateways, DHCP Options Sets, and Elastic IPs. The main content area is titled 'Resources' and features two buttons: 'Start VPC Wizard' (highlighted in blue) and 'Launch EC2 Instances'. A red arrow points from the 'Start VPC Wizard' button to the word 'Instances' in the note below. The note states: 'Note: Your Instances will launch in the US East (N. Virginia) region.' Below this, it lists the following Amazon VPC resources in the US East (N. Virginia) region:

5 VPCs	5 Internet Gateways
6 Subnets	10 Route Tables
5 Network ACLs	2 Elastic IPs
0 VPC Peering Connections	0 Endpoints
1 Nat Gateway	15 Security Groups
1 Running Instance	0 VPN Connections
0 Virtual Private Gateways	0 Customer Gateways

THIS IS NOT WHAT WE WANT

 **AWS** ▾ **Services** ▾ **Edit** ▾

Step 1: Select a VPC Configuration

VPC with a Single Public Subnet

Your instances run in a private, isolated section of the AWS cloud with direct access to the Internet. Network access control lists and security groups can be used to provide strict control over inbound and outbound network traffic to your instances.

Creates:
A /16 network with a /24 subnet. Public subnet instances use Elastic IPs or Public IPs to access the Internet.

Select

VPC with Public and Private Subnets


VPC with Public and Private Subnets and Hardware VPN Access

VPC with a Private Subnet Only and Hardware VPN Access

Internet, S3, DynamoDB, SNS, SQS, etc.

Public Subnet

Amazon Virtual Private Cloud



THIS IS WHAT WE WANT

Step 1: Select a VPC Configuration

VPC with a Single Public Subnet

VPC with Public and Private Subnets

VPC with Public and Private Subnets and Hardware VPN Access

VPC with a Private Subnet Only and Hardware VPN Access

In addition to containing a public subnet, this configuration adds a private subnet whose instances are not addressable from the Internet. Instances in the private subnet can establish outbound connections to the Internet via the public subnet using Network Address Translation (NAT).

Creates:

A /16 network with two /24 subnets. Public subnet instances use Elastic IPs to access the Internet. Private subnet instances access the Internet via Network Address Translation (NAT). (Hourly charges for NAT devices apply.)

Select

The diagram illustrates the VPC configuration. It shows an Amazon Virtual Private Cloud containing two subnets: a Public Subnet and a Private Subnet. The Public Subnet contains several server icons and is connected to a NAT gateway. The Private Subnet also contains several server icons. A cloud icon representing the Internet, with services like S3, SNS, and SQS, is connected to the Public Subnet. A line connects the NAT gateway in the Public Subnet to the Private Subnet, indicating that traffic from the private subnet is routed through the NAT gateway to the Internet.

PRIVATE VS PUBLIC SUBNETS

- Public subnet == internet facing
 - External subnet
 - The hax0rs can get your datas
- Private subnet != internet facing
 - Internal subnet
 - The hax0rs will have a much harder time getting your datas



PRIVATE VS PUBLIC SUBNETS

- Public subnet requires an IGW (internet gateway) to talk to the internet
- Private subnet requires an NGW (NAT gateway) to talk to the internet

WTF ARE ALL THESE GATEWAYS

- IGW (internet gateway)
 - Your public subnet traffic talks directly to the internets through the IGW
- NGW (NAT (Network Address Translation) gateway)
 - Acronyms FTMFW
 - Your private subnet traffic must tunnel into the public subnet, through the NAT gateway, and *then* through the IGW

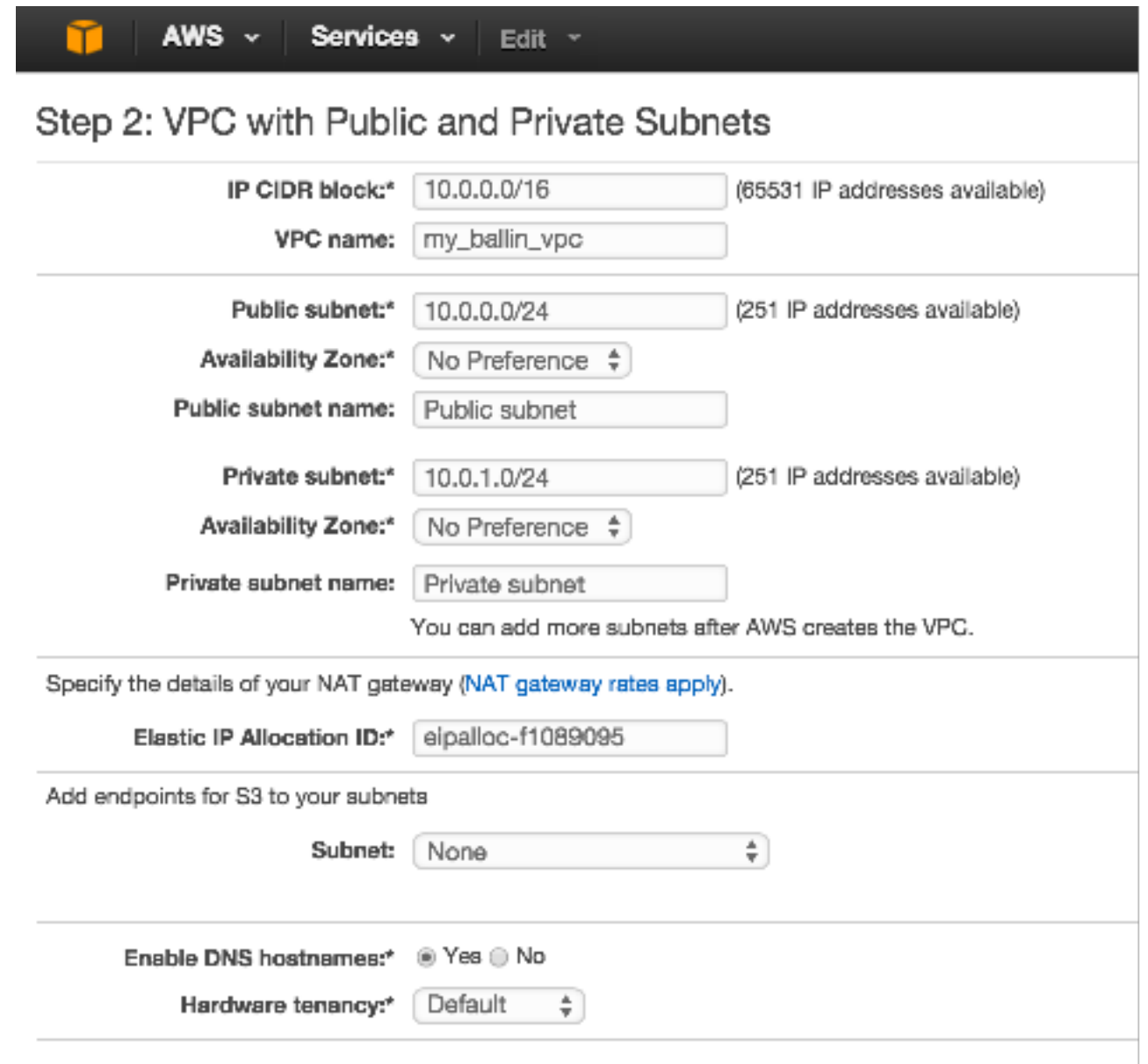
MOST IMPORTANTLY

- NAT Gateways cost money
- Internet Gateways do not
- So delete your NAT when you're done



BELLS AND WHISTLES

- Choose your IP ranges, default is fine
- Name your VPC and subnets
- Place them in the same availability zone
- Aaand... now we need an elastic IP for our NAT gateway
- Open EC2 dashboard in a new tab



The screenshot shows the AWS Management Console interface for configuring a VPC. The navigation bar at the top includes the AWS logo, 'AWS', 'Services', and 'Edit' dropdown menus. The main heading is 'Step 2: VPC with Public and Private Subnets'. The configuration is organized into several sections:

- VPC Configuration:** 'IP CIDR block:' is set to '10.0.0.0/16' (65531 IP addresses available). 'VPC name:' is 'my_ballin_vpc'.
- Public Subnet Configuration:** 'Public subnet:' is '10.0.0.0/24' (251 IP addresses available). 'Availability Zone:' is 'No Preference'. 'Public subnet name:' is 'Public subnet'.
- Private Subnet Configuration:** 'Private subnet:' is '10.0.1.0/24' (251 IP addresses available). 'Availability Zone:' is 'No Preference'. 'Private subnet name:' is 'Private subnet'.

Below the subnet configurations, there is a note: 'You can add more subnets after AWS creates the VPC.' The next section is 'Specify the details of your NAT gateway (NAT gateway rates apply)', with 'Elastic IP Allocation ID:' set to 'eipalloc-f1089095'. The 'Add endpoints for S3 to your subnets' section has 'Subnet:' set to 'None'. At the bottom, 'Enable DNS hostnames:' is set to 'Yes' (radio button selected) and 'Hardware tenancy:' is set to 'Default'.

GET AN ELASTIC IP


The screenshot displays the AWS Management Console interface for Elastic IP addresses. At the top, there is a navigation bar with the AWS logo, 'AWS', 'Services', and 'Edit' dropdown menus. On the left side, there is a navigation menu with options: EC2 Dashboard, Events, Tags, Reports, Limits, INSTANCES (expanded), Instances, Spot Requests, Reserved Instances, Scheduled Instances, Commands, and Dedicated Hosts.

The main content area shows the 'Elastic IP' management page. At the top, there is a blue 'Allocate New Address' button and an 'Actions' dropdown menu. Below this is a search bar with the text 'search : eipalloc-e0325184'. A table lists the Elastic IP addresses:

<input type="checkbox"/>	Elastic IP	Instance	Private IP Address
<input checked="" type="checkbox"/>	52.6.108.168	eipalloc-e0325184	10.0.0.183

An 'Actions' dropdown menu is open over the table, showing the following options: 'Allocate New Address' (highlighted in orange), 'Release Address', 'Associate Address', and 'Disassociate Address'.

ASSIGN THE ELASTIC IP

 Services ▾ Resource Groups ▾ ↻

Step 2: VPC with Public and Private Subnets

IPv4 CIDR block:* (65531 IP addresses available)

IPv6 CIDR block: No IPv6 CIDR Block
 Amazon provided IPv6 CIDR block

VPC name:

Public subnet's IPv4 CIDR:* (251 IP addresses available)

Availability Zone:*

Public subnet name:

Private subnet's IPv4 CIDR:* (251 IP addresses available)

Availability Zone:*

Private subnet name:

You can add more subnets after AWS creates the VPC.

Specify the details of your NAT gateway ([NAT gateway rates apply](#)).

Elastic IP Allocation ID:*

Service endpoints

Enable DNS hostnames:* Yes No

Hardware tenancy:*

FAIL



AWS

Services

Edit

Whitney Champion - N. Virginia -

VPC Dashboard

Filter by VPC:

None

Virtual Private Cloud

Your VPCs

Subnets

VPC Creation Failed

There was an error creating your VPC: The maximum number of VPCs has been reached. (Service: AmazonEC2; Status Code: 400; Error Code: VpcLimitExceeded; Request ID: ead692fd-609f-4af1-b156-75abc01b2794)

SUCCESS



AWS ▾

Services ▾

Edit ▾

VPC Dashboard

Filter by VPC:

None ▾

Virtual Private Cloud

Your VPCs

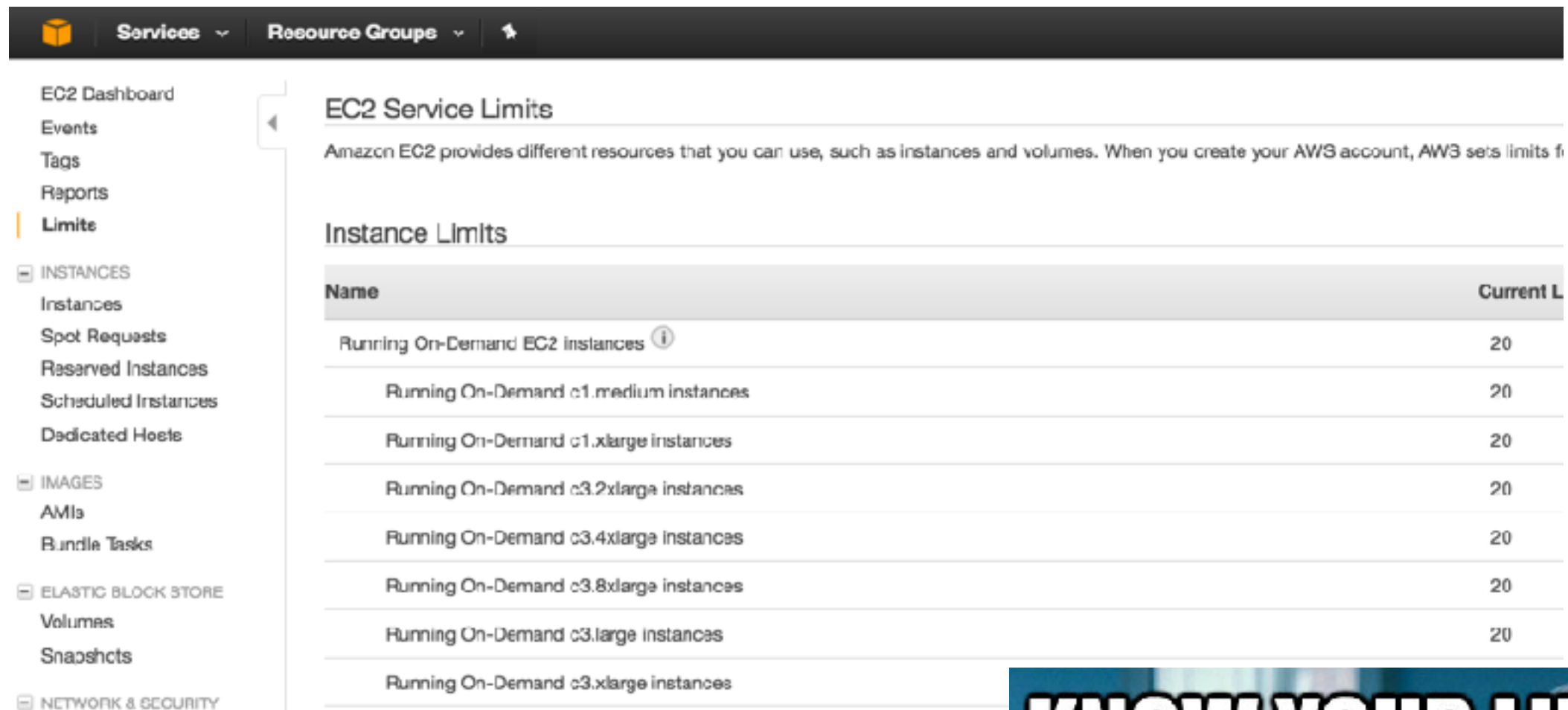
VPC Successfully Created

Your VPC has been successfully created.

You can launch instances into the subnets of your VPC. For more information, see [Launching an Instance into Your Subnet](#).

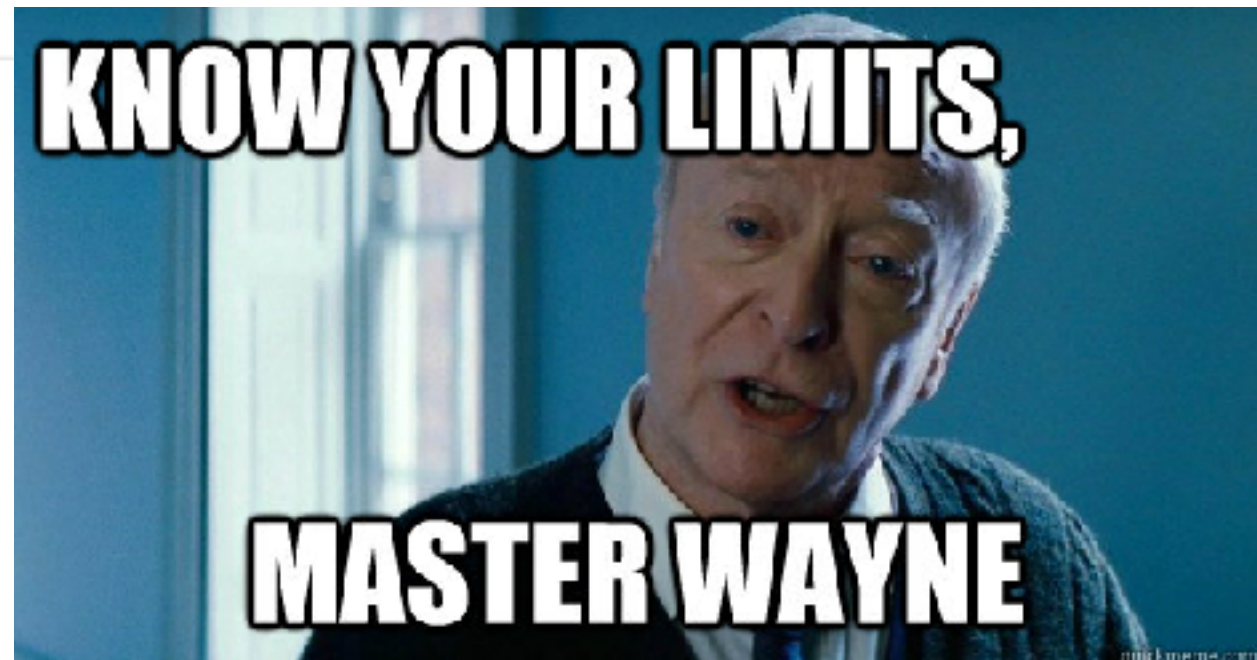
Tangent!

ALL AWS SERVICES HAVE LIMITS



The screenshot shows the AWS Management Console interface. At the top, there are navigation tabs for 'Services' and 'Resource Groups'. On the left, a sidebar menu lists various AWS services, with 'Limits' highlighted. The main content area is titled 'EC2 Service Limits' and includes a brief introduction: 'Amazon EC2 provides different resources that you can use, such as instances and volumes. When you create your AWS account, AWS sets limits for you.' Below this, a section titled 'Instance Limits' contains a table with two columns: 'Name' and 'Current Limit'. The table lists several instance types, all with a current limit of 20.

Name	Current Limit
Running On-Demand EC2 instances ⓘ	20
Running On-Demand c1.medium instances	20
Running On-Demand c1.xlarge instances	20
Running On-Demand c3.2xlarge instances	20
Running On-Demand c3.4xlarge instances	20
Running On-Demand c3.8xlarge instances	20
Running On-Demand c3.large instances	20
Running On-Demand c3.xlarge instances	20



Back to the fun stuff...

ALL THE SUBNETS



The screenshot shows the AWS VPC console interface. At the top, there is a navigation bar with the AWS logo, 'AWS', 'Services', and 'Edit' menus. The user's name 'Whitney Champion' is visible in the top right corner. On the left side, there is a sidebar with the 'VPC Dashboard' and a list of navigation options: 'Your VPCs', 'Subnets' (highlighted with an orange bar), 'Route Tables', and 'Internet Gateways'. The main content area has a 'Filter by VPC:' dropdown set to 'None'. Below this, there are two buttons: 'Create Subnet' and 'Subnet Actions'. A search bar contains the text 'vpc-60f18904'. The main area displays a table of subnets with the following columns: Name, Subnet ID, State, VPC, CIDR, Available IPs, Availability Zone, Route Table, and Network ACL. Two subnets are listed: a 'Public subnet' and a 'Private subnet', both in an 'available' state.

<input type="checkbox"/>	Name	Subnet ID	State	VPC	CIDR	Available IPs	Availability Zone	Route Table	Network ACL
<input type="checkbox"/>	Public subnet	subnet-d0822ca3	available	vpc-60f18904 (10.0.0.0/16) my...	10.0.0.0/24	250	us-east-1c	rtb-7196ae15	acl-9f94dcfb
<input type="checkbox"/>	Private subnet	subnet-d2822ca4	available	vpc-60f18904 (10.0.0.0/16) my...	10.0.1.0/24	251	us-east-1c	rtb-7096ae14	acl-9f94dcfb

BUT WAIT! THERE'S MORE!

- Route tables
- Network Access Control Lists (NACLs)
- Gateways
- All the things



STEP 1: CHOOSE A FLAVOR

Services Resource Groups

Welcome to AWS - N. Virginia - Support

1. Choose AMI 2. Choose Instance Type 3. Configure Instance 4. Add Storage 5. Add Tags 6. Configure Security Group 7. Review

Step 1: Choose an Amazon Machine Image (AMI)

An AMI is a template that contains the software configuration (operating system, application server, and applications) required to launch your instances. You can select an AMI provided by AWS, our user community, or the AWS Marketplace; or you can select one of your own AMIs.

Cancel and Edit

Quick Start

- My AMIs
- AWS Marketplace
- Community AMIs
- Free tier only

1 to 31 of 31 AMIs

Amazon Linux Free tier eligible	Amazon Linux AMI 2016.09.1 (HVM), SSD Volume Type - ami-0b33d91d The Amazon Linux AMI is an EBS-backed, AWS-supported image. The default image includes AWS command line tools, Python, Ruby, Perl, and Java. The repositories include Docker, PHP, MySQL, Postgres, and other packages. Root device type: ebs Virtualization type: hvm	Select 64-bit
Red Hat Free tier eligible	Red Hat Enterprise Linux 7.3 (HVM), SSD Volume Type - ami-b83789a1 Red Hat Enterprise Linux version 7.3 (HVM), EBS General Purpose (SSD) Volume Type Root device type: ebs Virtualization type: hvm	Select 64-bit
SUSE Linux Free tier eligible	SUSE Linux Enterprise Server 12 SP2 (HVM), SSD Volume Type - ami-fde4e6ea SUSE Linux Enterprise Server 12 Service Pack 2 (HVM), EBS General Purpose (SSD) Volume Type. Public Cloud, Advanced Systems Management, Web and Scripting, and Legacy modules enabled. Root device type: ebs Virtualization type: hvm	Select 64-bit
Ubuntu Free tier eligible	Ubuntu Server 16.04 LTS (HVM), SSD Volume Type - ami-e13738f8 Ubuntu Server 16.04 LTS (HVM), EBS General Purpose (SSD) Volume Type. Support available from Canonical: http://www.ubuntu.com/cloud/services . Root device type: ebs Virtualization type: hvm	Select 64-bit
Windows Free tier eligible	Microsoft Windows Server 2016 Base - ami-188d8e0e Microsoft Windows 2016 Datacenter edition. [English] Root device type: ebs Virtualization type: hvm	Select 64-bit

Are you launching a database instance? Try Amazon RDS. Hide

Amazon Relational Database Service (RDS) makes it easy to set up, operate, and scale your database on AWS by automating time-consuming database management tasks. With RDS, you can easily deploy Amazon Aurora, MariaDB, MySQL, Oracle, PostgreSQL, and SQL Server databases on AWS. Aurora is a MySQL-compatible, enterprise-class database at 1/10th the cost of commercial databases. [Learn more about RDS](#)

[Launch a database using RDS](#)

STEP 2: CHOOSE SPECS

1. Choose AMI

2. Choose Instance Type

3. Configure Instance

4. Add Storage

5. Add Tags

6. Configure Security Group

7. Review

Step 2: Choose an Instance Type

Amazon EC2 provides a wide selection of instance types optimized to fit different use cases. Instances are virtual servers that can run applications. They provide the computing resources for your applications. [Learn more](#) about instance types and how they can meet your computing needs.

Filter by:

All instance types

Current generation

Show/Hide Columns

Currently selected: t2.micro (Variable ECUs, 1 vCPUs, 2.5 GHz, Intel Xeon Family, 1 GiB memory, EBS only)

	Family	Type	vCPUs	Memory (GiB)
<input type="checkbox"/>	General purpose	t2.nano	1	0.5
<input checked="" type="checkbox"/>	General purpose	t2.micro Free tier eligible	1	1
<input type="checkbox"/>	General purpose	t2.small	1	2
<input type="checkbox"/>	General purpose	t2.medium	2	4
<input type="checkbox"/>	General purpose	t2.large	2	8

STEP 3: CONFIGURE DETAILS

1. Choose AMI 2. Choose Instance Type **3. Configure Instance** 4. Add Storage 5. Add Tags 6. Configure Security Group 7. Review

Step 3: Configure Instance Details

Configure the instance to suit your requirements. You can launch multiple instances from the same AMI, request Spot instances to take advantage of the lower pricing, a

Number of instances ⓘ	1	Launch into Auto Scaling Group ⓘ
Purchasing option ⓘ	<input type="checkbox"/> Request Spot instances	
Network ⓘ	vpc-47c14d20 SG DEPLOY VPC (default) ⌵	Create new VPC
Subnet ⓘ	No preference (default subnet in any Availability Zone) ⌵	Create new subnet
Auto-assign Public IP ⓘ	Use subnet setting (Enable) ⌵	
IAM role ⓘ	None ⌵	Create new IAM role
Shutdown behavior ⓘ	Stop ⌵	
Enable termination protection ⓘ	<input type="checkbox"/> Protect against accidental termination	
Monitoring ⓘ	<input type="checkbox"/> Enable CloudWatch detailed monitoring Additional charges apply.	
Tenancy ⓘ	Shared - Run a shared hardware instance ⌵ Additional charges will apply for dedicated tenancy.	

▶ [Advanced Details](#)

STEP 4: STORAGE

- 1. Choose AMI
- 2. Choose Instance Type
- 3. Configure Instance
- 4. Add Storage**
- 5. Add Tags
- 6. Configure Security Group
- 7. Review

Step 4: Add Storage

Your instance will be launched with the following storage device settings. You can attach additional EBS volumes and instance store volumes to you edit the settings of the root volume. You can also attach additional EBS volumes after launching an instance, but not instance store volumes. [Learn more](#) about storage options in Amazon EC2.

Volume Type <small>i</small>	Device <small>i</small>	Snapshot <small>i</small>	Size (GiB) <small>i</small>	Volume Type <small>i</small>	IOPS <small>i</small>	Throughput (MB/s) <small>i</small>
Root	/dev/xvda	snap-037f1f9e6c8ea4d65	<input type="text" value="8"/>	General Purpose SSD (GP2) <small>i</small>	100 / 3000	N/A

[Add New Volume](#)

Free tier eligible customers can get up to 30 GB of EBS General Purpose (SSD) or Magnetic storage. [Learn more](#) about free usage tier eligibility usage restrictions.

STEP 5: TAGS

- Assign tags (optional)

1. Choose AMI 2. Choose Instance Type 3. Configure Instance 4. Add Storage 5. Add Tags 6. Configure Security Group 7. Review

Step 5: Add Tags

A tag consists of a case-sensitive key-value pair. For example, you could define a tag with key = Name and value = Webservers.

A copy of a tag can be applied to volumes, instances or both.

Tags will be applied to all instances and volumes. [Learn more](#) about tagging your Amazon EC2 resources.

Key (127 characters maximum)	Value (255 characters maximum)	Instances ⓘ	Volumes ⓘ	
Name	bastion	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
Env	dev	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>

Add another tag

(Up to 50 tags maximum)

STEP 6: SECURITY GROUPS

- Create a security group
 - By default: deny all inbound
 - Goal: Allow only what is necessary

Step 6: Configure Security Group

A security group is a set of firewall rules that control the traffic for your instance. On this page, you can add rules to allow specific traffic to reach your instance. For example, if you want to set up a web server and allow Internet traffic to reach your instance, add rules that allow unrestricted access to the HTTP and HTTPS ports. You can create a new security group or select from an existing one below. [Learn more](#) about Amazon EC2 security groups.

Assign a security group: Create a new security group
 Select an existing security group

Security group name:

Description:

Type <small>i</small>	Protocol <small>i</small>	Port Range <small>i</small>	Source <small>i</small>	
SSH <small>⌵</small>	TCP	22	My IP <small>⌵</small> 74.118.240.108/32	<small>✕</small>

STEP 7: LAUNCH OUR INSTANCE

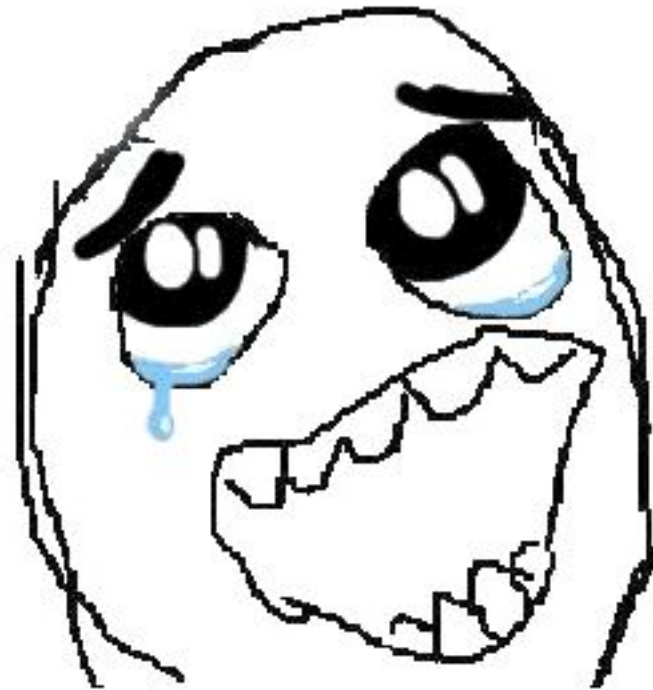
- **REVIEW AND LAUNCH!**
- It will ask you to choose or create a key
- If you create a new one, save it in a safe place
 - At the very least, *remember where you put it*
- You will need it to SSH

STEP 8: ASSIGN ELASTIC IP

- Optional
- Just like we did for the NAT gateway, go allocate a new elastic IP address
 - EC2 Dashboard —> Elastic IPs —> Allocate New
- Associate it with your new instance

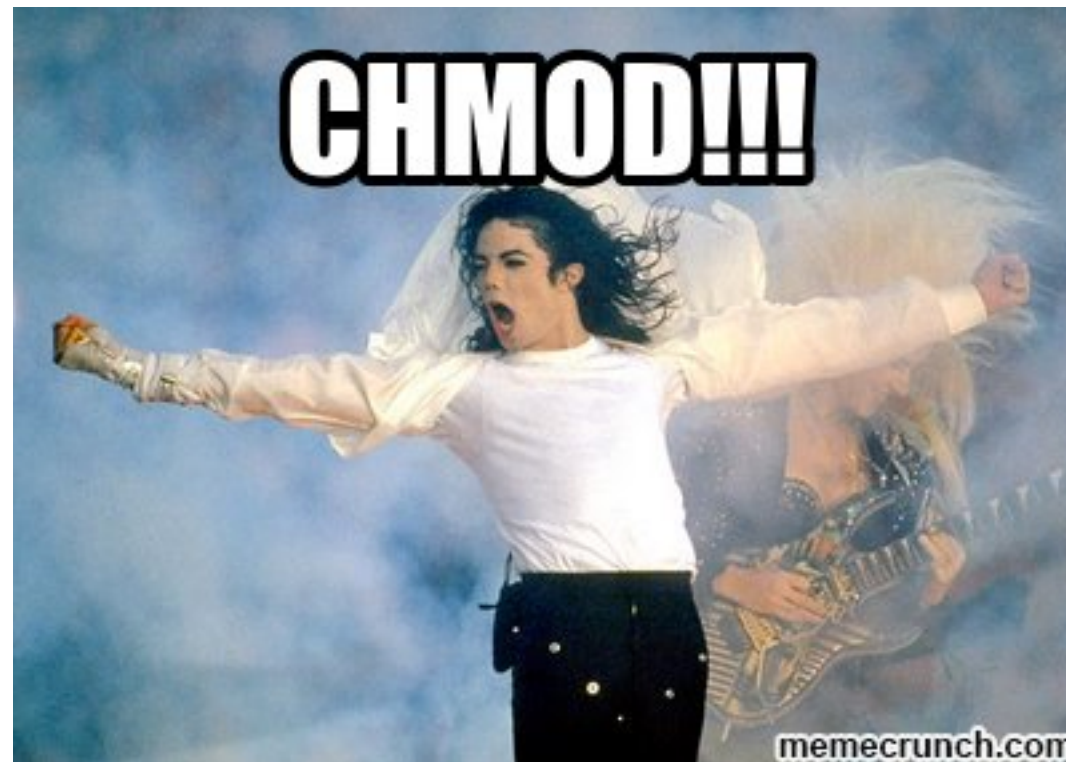
IT'S ALIVE

```
ssh -i /path/to/your_key.pem ec2-user@$ELASTIC_IP
```



PERMISSION DENIED?

```
chmod 600 your_key.pem
```



NOW YOU HAVE A BASTION HOST

- It is sitting in your public subnet (so you can get to it)
 - You will use it to SSH tunnel to the instances you deploy in your private subnet (aka the stuff you care about)
- Now repeat steps 1-7 on your own to deploy a second instance in the private subnet
 - Choose the **private** subnet in step 3
 - Make note of the **private IP address** assigned to the second instance

Step 6: Configure Security Group

A security group is a set of firewall rules that control the traffic for your instance. On this page, you can add rules to allow specific traffic to reach your instance. For example, if you want to set up a web server and allow Internet traffic to reach your instance, add rules that allow unrestricted access to the HTTP and HTTPS ports. You can create a new security group or select from an existing one below. [Learn more](#) about Amazon EC2 security groups.

Assign a security group: Create a new security group

Select an existing security group

Security group name:

Description:

Type ⓘ	Protocol ⓘ	Port Range ⓘ	Source ⓘ	
SSH ▾	TCP	22	Custom ▾ 10.0.0.0/16	✕
HTTP ▾	TCP	80	Custom ▾ 0.0.0.0/0, ::/0	✕
HTTPS ▾	TCP	443	Custom ▾ 0.0.0.0/0, ::/0	✕

Add Rule

OTHER OPTIONS

- Rather than using a bastion host, you can set up a VPN server in your public subnet
 - OpenVPN is a great choice
 - Easy to set up
 - Free for 2 concurrent users
 - 2 factor authentication option
- 2 or more VPCs connected with VPC peering, containing only private subnets, and a designated VPC with a public subnet for a bastion or VPN servers
- Or whatever you want

CONFIGURE SSH TUNNEL

- Mac
 - `vim ~/.ssh/config`

```
Host $PRIVATE_IP
    ProxyCommand ssh bastion -W %h:%p
Host bastion
    Hostname $ELASTIC_IP
    User ec2-user
    IdentityFile /path/to/your_key.pem
```

- Windows
 - Generate ppk with PuTTYgen
 - Save connection settings in PuTTY with your ppk
 - Connect via mRemote
- Or use an SSH manager (Royal TSX, Termius, Shuttle, etc)

TUNNEL ALL THE THINGS

```
ssh -i /path/to/your_key.pem ec2-user@$PRIVATE_IP
```

INSTALL A SIMPLE WEB SERVER

```
sudo yum -y install httpd  
sudo service httpd start  
sudo chkconfig httpd on
```

LOAD BALANCING

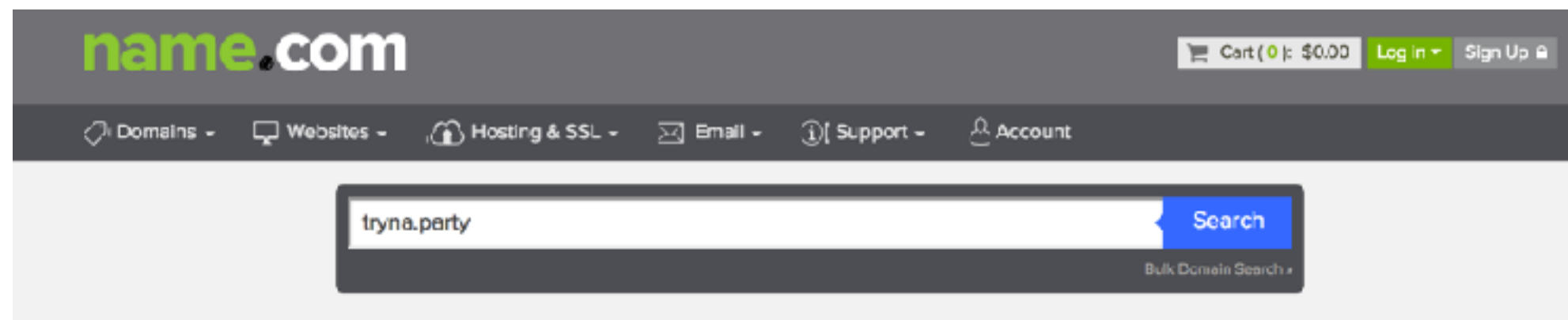
- Now you have a web server, but no way for anyone to get to it
- Create a classic load balancer with the following listener configuration

Load Balancer Protocol	Load Balancer Port	Instance Protocol	Instance Port	
HTTP	80	HTTP	80	X
<input type="button" value="Add"/>				

- Choose your VPC, the **public** subnet, and server security group
- Health check on TCP port 80, Healthy threshold 2
- Assign your web server instance to it

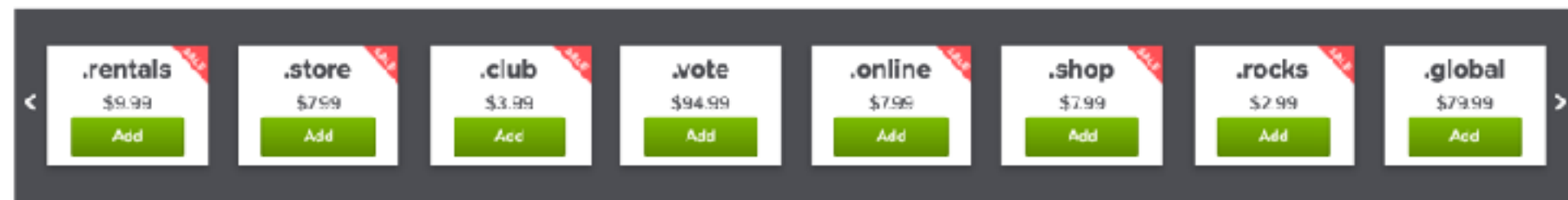
YOUR LOAD BALANCER IS LIVE

- Buy a domain
- Add a CNAME for your load balancer
- Put something weird on your web server
- ...
- Profit



Your domain is available!
tryna.party

\$34.99 Add to Cart
Renewal: \$34.99





I EXPECT GREAT THINGS.

DO NOT DISSAPOINT ME.

memegenerator.net