Amazon Web Service

Setting up Elastic Cloud 2 (EC2)

Amazon Web Services (AW ×

aws.amazon.com

Menu

amazon web services

English ▾ Sign In ▾ Sign Up

AWS re:Invent

November 11-14, 2014 | Las Vegas, NV

Register to watch the Live Stream

Sign up today »

Manage Your Resources

Sign in to the Console

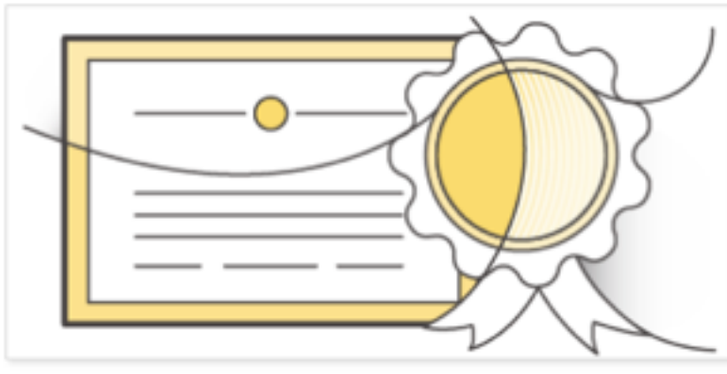
AWS Console Mobile App

View your resources on iOS and Android devices

Learn More and Download »

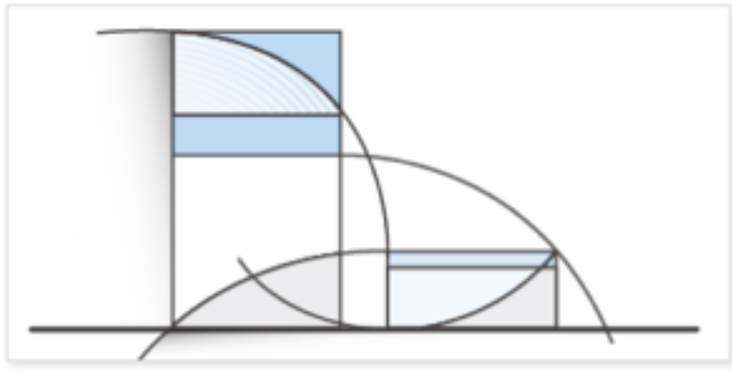
AWS STARTUP PROGRAM

Includes training, support, guidance, and more. Free to join



GET AWS CERTIFIED AT RE:INVENT

Validate your technical expertise. Registration for onsite exams is now open

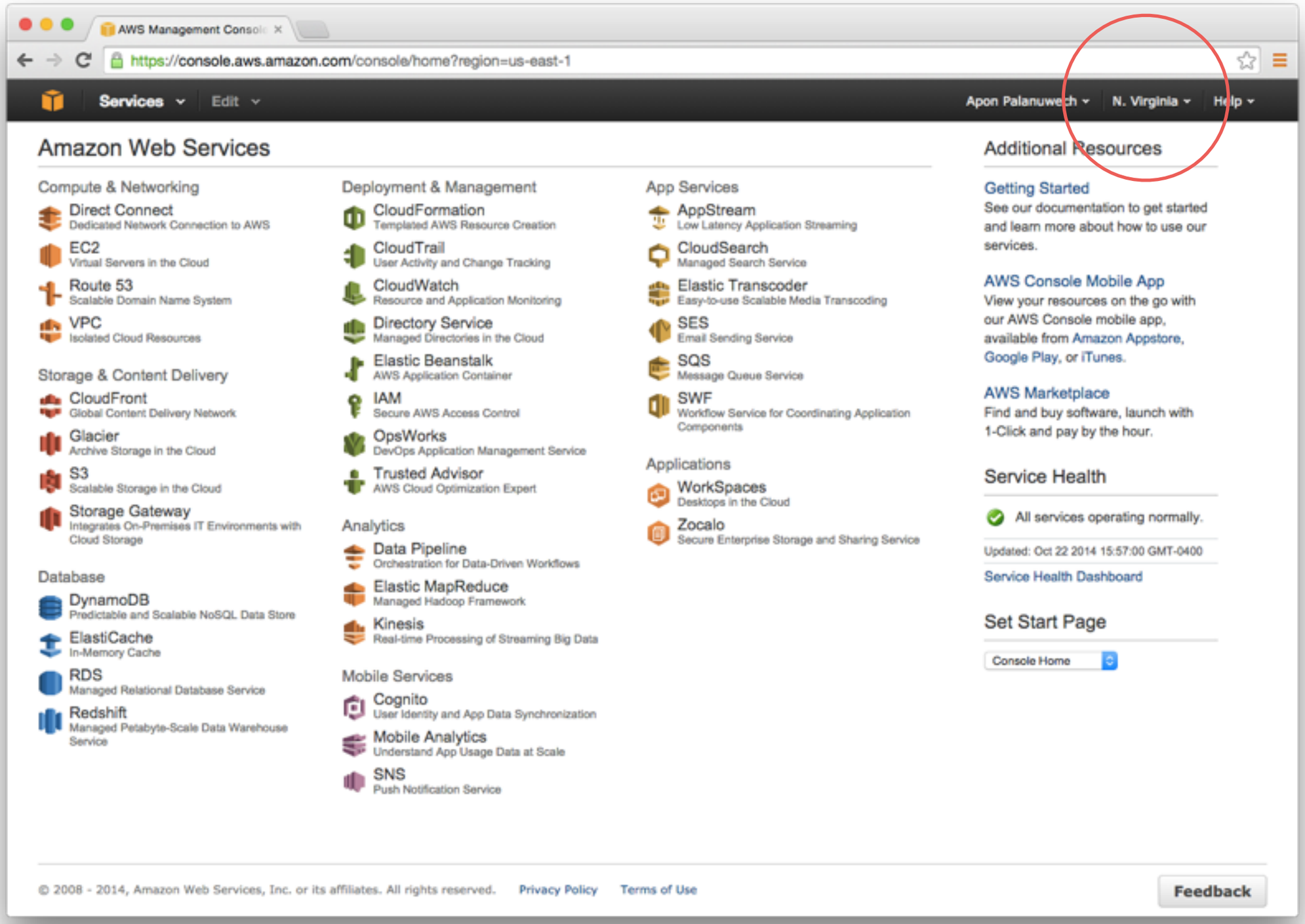


AWS TCO CALCULATOR

See how much you could save compared to traditional infrastructure

aws.amazon.com

CAUTION!!!
Without free tier, 1 EC2 unit costs me $15 monthly
If you don’t use it, stop or terminate it.



- change location to N.Virginia or any station near you
- select EC2

EC2 Management Console

https://console.aws.amazon.com/ec2/v2/home?region=us-east-1#

Services | Edit | Apon Palanuwech | N. Virginia | Help

- EC2 Dashboard
- Events
- Tags
- Reports
- Limits

INSTANCES
- Instances
- Spot Requests
- Reserved Instances

IMAGES
- AMIs
- Bundle Tasks

ELASTIC BLOCK STORE
- Volumes
- Snapshots

NETWORK & SECURITY
- Security Groups
- Elastic IPs
- Placement Groups
- Load Balancers
- Key Pairs
- Network Interfaces

AUTO SCALING
- Launch Configurations
- Auto Scaling Groups

Resources

You are using the following Amazon EC2 resources in the US East (N. Virginia) region:

- 2 Running Instances
- 2 Volumes
- 12 Key Pairs
- 0 Placement Groups
- 1 Elastic IP
- 0 Snapshots
- 0 Load Balancers
- 7 Security Groups

Easily deploy Ruby, PHP, Java, .NET, Python, Node.js & Docker applications with Elastic Beanstalk. Hide

Create Instance

To start using Amazon EC2 you will want to launch a virtual server, known as an Amazon EC2 instance.

Launch Instance

Note: Your instances will launch in the US East (N. Virginia) region

Service Health

Service Status:

US East (N. Virginia):
This service is operating normally

Availability Zone Status:

us-east-1a:
Availability zone is operating normally

us-east-1c:
Availability zone is operating normally

us-east-1d:
Availability zone is operating normally

Service Health Dashboard

Scheduled Events

US East (N. Virginia):

No events

Account Attributes

Supported Platforms

Additional Information

Getting Started Guide
Documentation
All EC2 Resources
Forums
Pricing
Contact Us

AWS Marketplace

Find **free software trial** products in the AWS Marketplace from the EC2 Launch Wizard.
Or try these popular AMIs:

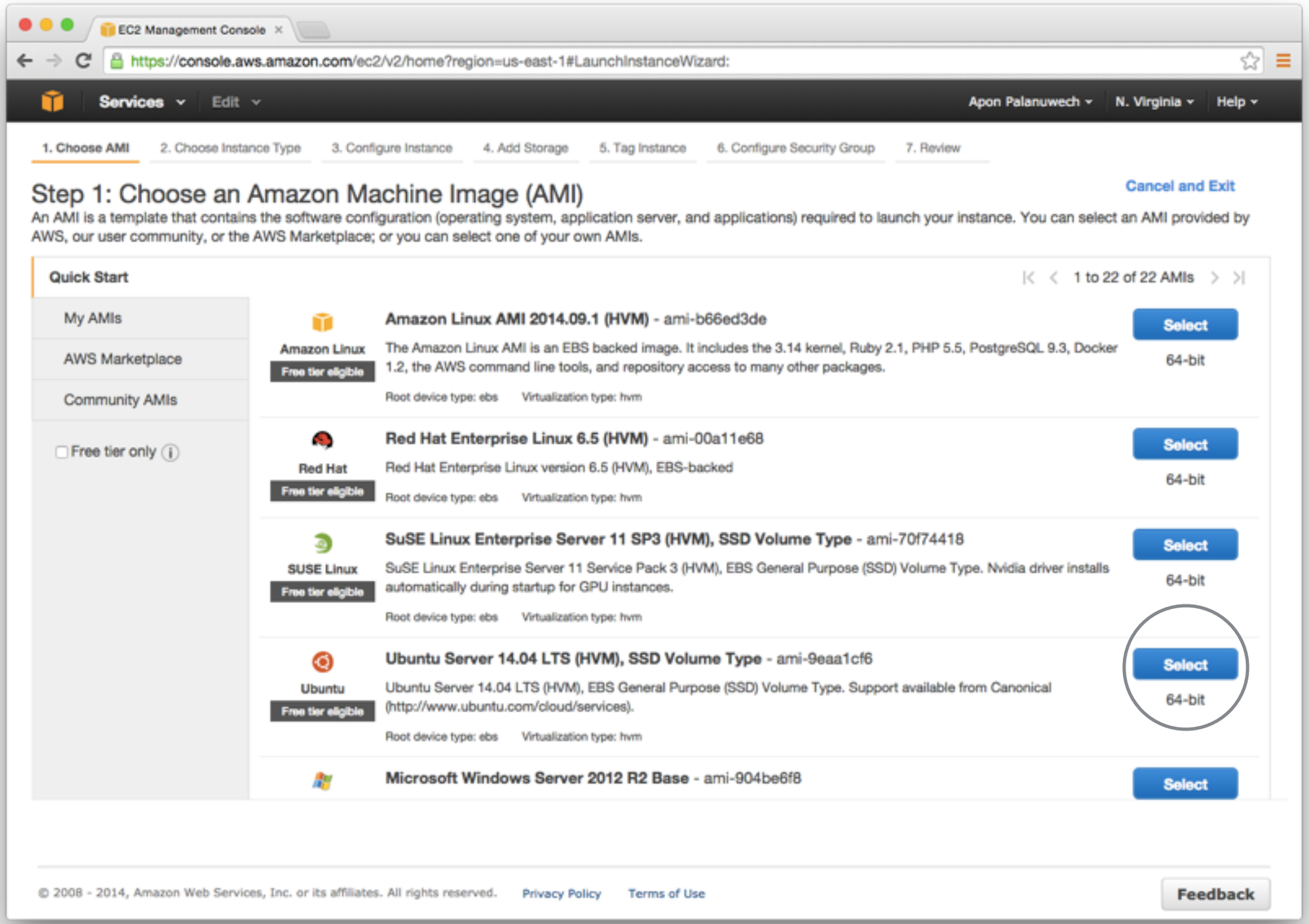
Vyatta Virtual Router/Firewall/VPN
Provided by Vyatta, Inc.
Rating ★★★★☆
Pay by the hour for software and AWS usage
View all Networking

Alert Logic Threat Manager for AWS
Provided by Alert Logic
Rating ★★★★½
Pay by the hour for software and AWS usage
View all Security Software

ColdFusion 11

 Privacy Policy Terms of Use

Feedback



We'll be using Ubuntu distribution
(Free for a year!)

EC2 Management Console

https://console.aws.amazon.com/ec2/v2/home?region=us-east-1#LaunchInstanceWizard:

Services | Edit | Apon Palanuwech | N. Virginia | Help

1. Choose AMI | 2. Choose Instance Type | 3. Configure Instance | 4. Add Storage | 5. Tag Instance | 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 have varying combinations of CPU, memory, storage, and networking capacity, and give you the flexibility to choose the appropriate mix of 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

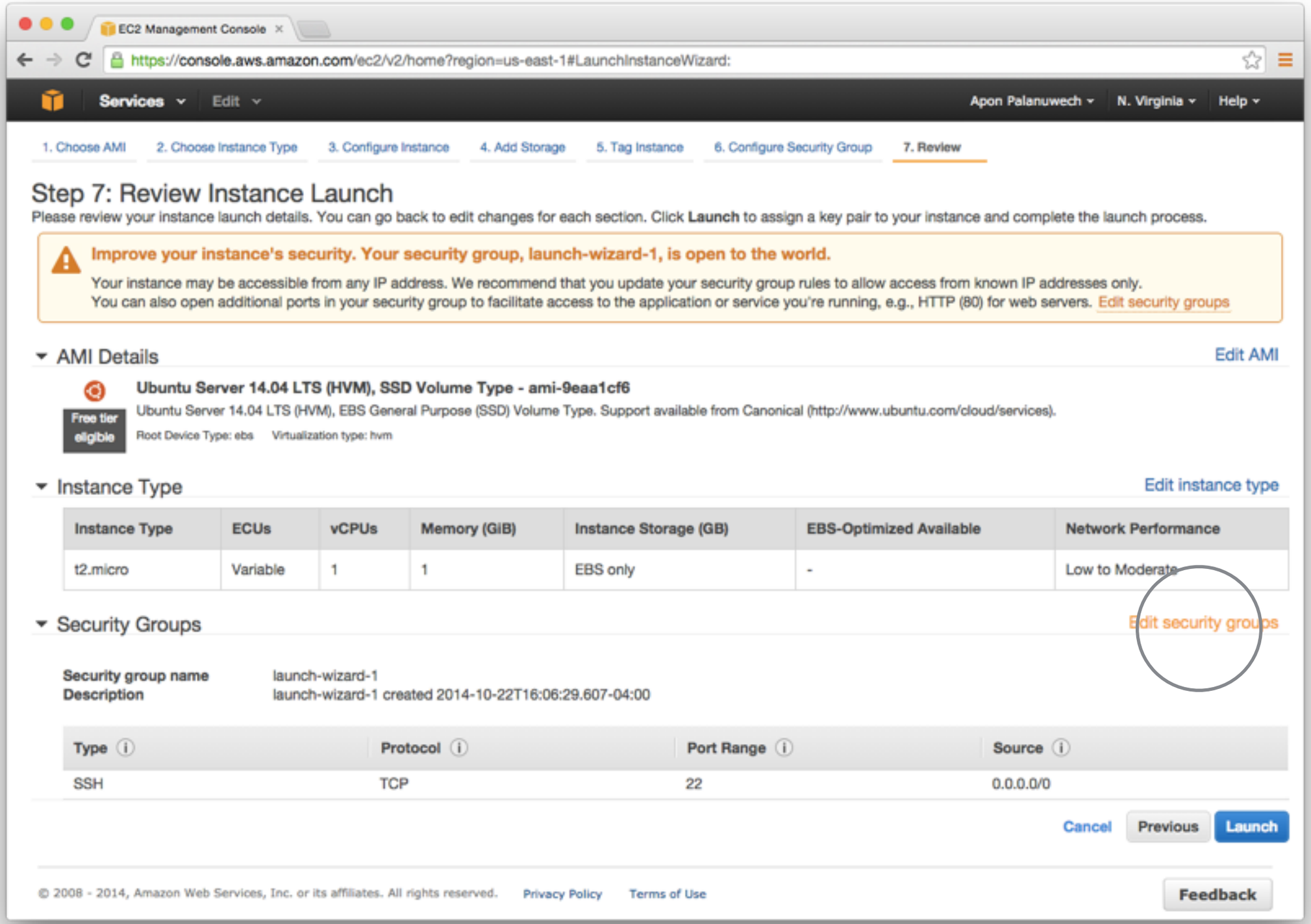
Currently selected: t2.micro (Variable ECUs, 1 vCPUs, 2.5 GHz, Intel Xeon Family, 1 GiB memory, EBS only)

T2 instances are VPC-only. Your T2 instance will launch into your VPC. Learn more about T2 and VPC.

	Family	Type	vCPUs	Memory (GiB)	Instance Storage (GB)	EBS-Optimized Available	Network Performance
☑	General purpose	t2.micro (Free tier eligible)	1	1	EBS only	-	Low to Moderate
☐	General purpose	t2.small	1	2	EBS only	-	Low to Moderate
☐	General purpose	t2.medium	2	4	EBS only	-	Low to Moderate
☐	General purpose	m3.medium	1	3.75	1 x 4 (SSD)	-	Moderate
☐	General purpose	m3.large	2	7.5	1 x 32 (SSD)	-	Moderate
☐	General purpose	m3.xlarge	4	15	2 x 40 (SSD)	Yes	High
☐	General purpose	m3.2xlarge	8	30	2 x 80 (SSD)	Yes	High

Cancel | Previous | Review and Launch | Next: Configure Instance Details

 Privacy Policy | Terms of Use | Feedback



Edit security groups allows full control of all ports remember 0-65535

EC2 Management Console

https://console.aws.amazon.com/ec2/v2/home?region=us-east-1#LaunchInstanceWizard:

Services | Edit | Apon Palanuwech | N. Virginia | Help

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

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: my_triple_security

Description: something something

Type	Protocol	Port Range	Source	
SSH	TCP	22	Anywhere 0.0.0.0/0	⊗
Custom TCP Rule	TCP	80	Anywhere 0.0.0.0/0	⊗
Custom TCP Rule	TCP	445	Anywhere 0.0.0.0/0	⊗
Custom TCP Rule	TCP	8080	Anywhere 0.0.0.0/0	⊗
Custom TCP Rule	TCP	9999	Anywhere 0.0.0.0/0	⊗
Custom TCP Rule	TCP	3000-4000	Anywhere 0.0.0.0/0	⊗

Add Rule

Warning

Rules with source of 0.0.0.0/0 allow all IP addresses to access your instance. We recommend setting security group rules to allow access from known IP addresses only.

Cancel | Previous | Review and Launch

 Privacy Policy | Terms of Use | Feedback

EC2 Management Console

https://console.aws.amazon.com/ec2/v2/home?region=us-east-1#LaunchInstanceWizard:

Services | Edit | Apon Palanuwech | N. Virginia | Help

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

Step 7: Review Instance Launch

Please review your instance launch details. You can go back to edit changes for each section. Click **Launch** to assign a key pair to your instance and complete the launch process.

Improve your instance's security. Your security group, my_triple_security, is open to the world.

Your instance may be access... IP addresses only.
You can also open additional ...eb servers. Edit security groups

AMI Details

Edit AMI

Ubuntu Server 14.04

Ubuntu Server 14.04 LTS...

Free tier eligible

Root Device Type: ebs Vir...

...ices).

Instance Type

Edit instance type

Instance Type	ECUs		Network Performance
t2.micro	Variable		Low to Moderate

Security Groups

Edit security groups

Security group name m...
Description sc...

Type	...rce
SSH	...0.0/0

Cancel | Previous | Launch

© 2008 - 2014, Amazon Web Services, Inc. or its affiliates. All rights reserved. Privacy Policy Terms of Use

Feedback

Select an existing key pair or create a new key pair

A key pair consists of a **public key** that AWS stores, and a **private key file** that you store. Together, they allow you to connect to your instance securely. For Windows AMIs, the private key file is required to obtain the password used to log into your instance. For Linux AMIs, the private key file allows you to securely SSH into your instance.

Note: The selected key pair will be added to the set of keys authorized for this instance. Learn more about removing existing key pairs from a public AMI.

Create a new key pair

Key pair name

aws_key

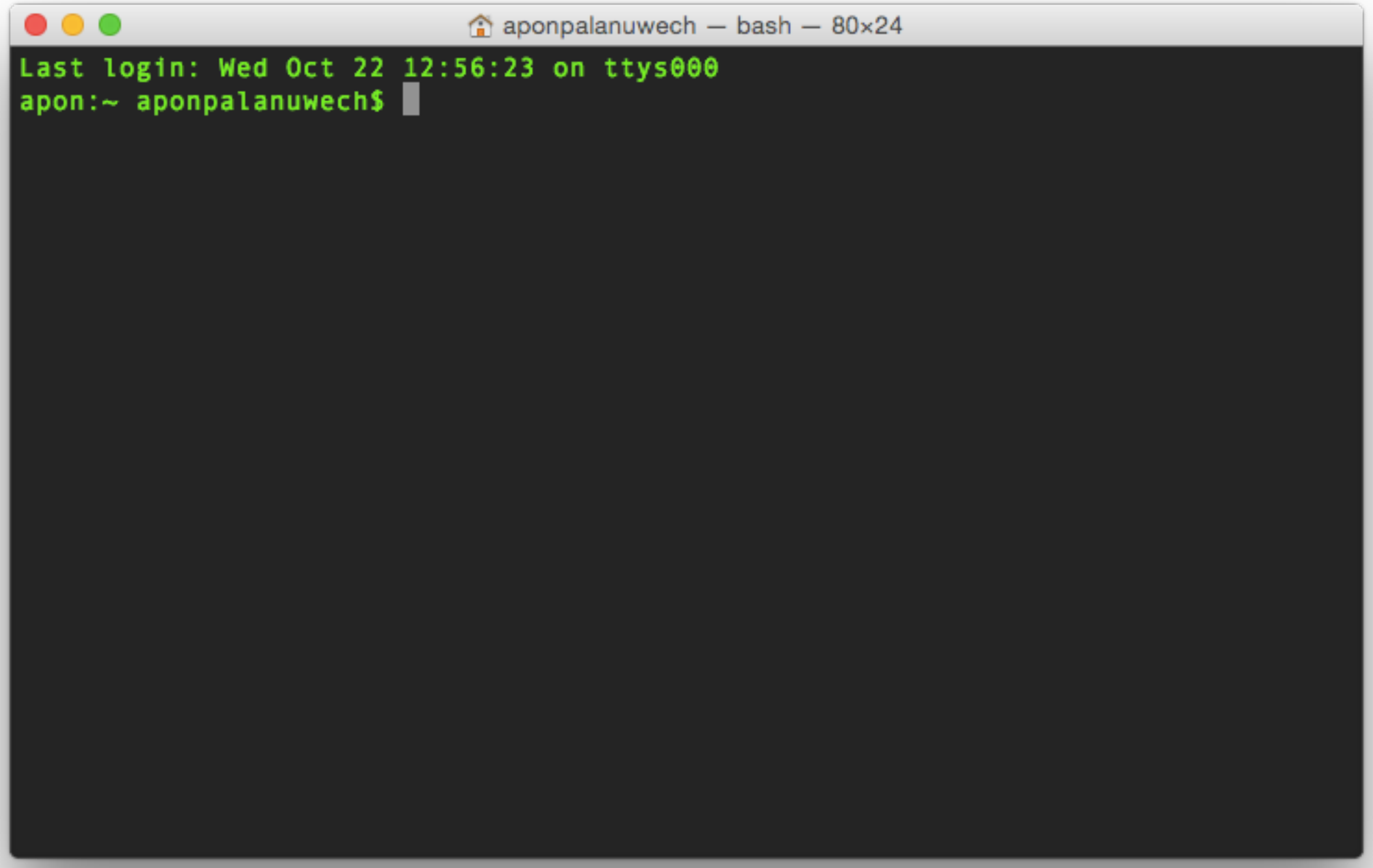
Download Key Pair

You have to download the **private key file** (*.pem file) before you can continue. **Store it in a secure and accessible location.** You will not be able to download the file again after it's created.

Cancel | Launch Instances

Now wait like 5 minutes…

In the meantime, let’s configure our .pem
we just downloaded


aponpalanuwech — bash — 80×24
Last login: Wed Oct 22 12:56:23 on ttys000
apon:~ aponpalanuwech$

Ubuntu requires the .pem key to have 0600 permission
http://www.thinkplexx.com/learn/article/unix/command/chmod-permissions-flags-explained-600-0600-700-777-100-etc

$ sudo chmod 0600 ~/Downloads/aws_key.pem

Now assuming that our server is ready to go. Check instances tab —> if 2/2 checks pass then it's good.

- Grab Public IP, for example, 54.173.20.63
- $ ssh ubuntu@ 54.173.20.63 -i ~/Download/aws_key.pem
- say yes
- and boom! you're now in a Linux machine somewhere in Amazon computer center in North Virginia

Using Nano text editor

- to create new file: $ sudo nano banana.txt
- to save and exit: ctrl + x then return

Now we will:

– Add a user (ourselves)
– Append this user as ROOT so he/she can do root commands (sudo)
– Edit Secure-shell (ssh) configuration
– Restart sshd

- $ sudo adduser apon
- set your password
- $ sudo visudo

http://askubuntu.com/questions/59458/error-message-when-i-run-sudo-unable-to-resolve-host-none

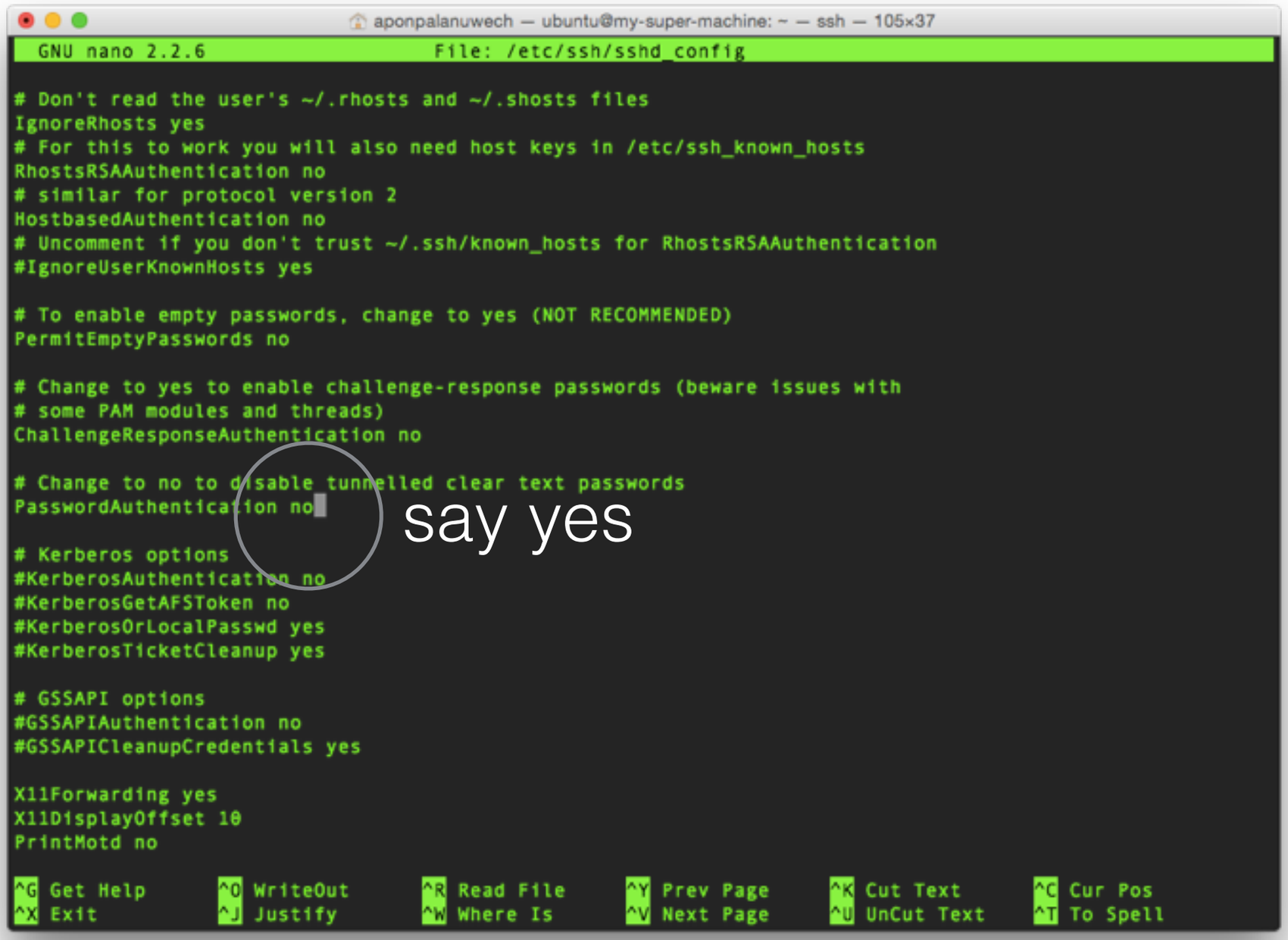
104

Two things to check (assuming your machine is called `my-machine`, you can change this as appropriate):

1. That the `/etc/hostname` file contains just the name of the machine.
2. That `/etc/hosts` has an entry for `localhost`. It should have something like:

```
127.0.0.1    localhost.localdomain localhost
127.0.1.1    my-machine
```

- $ sudo nano /etc/ssh/sshd_config



- $ sudo service ssh restart

Let’s login using our username!

On your mac,

$ ssh apon@ 54.173.20.63

enter password

// Let's update our machine and install some useful stuff:

- $ sudo add-apt-repository ppa:chris-lea/node.js

// For mongodb reference: http://docs.mongodb.org/manual/tutorial/install-mongodb-on-ubuntu/

- $ sudo apt-key adv --keyserver hkp://keyserver.ubuntu.com:80 --recv 7F0CEB10
- $ echo 'deb http://downloads-distro.mongodb.org/repo/ubuntu-upstart dist 10gen' | sudo tee /etc/apt/sources.list.d/mongodb.list
- $ sudo apt-get update
- $ sudo apt-get install nodejs samba build-essential

// mongodb takes a lot of space and time to install, do it only when needed

- $ sudo apt-get install mongodb-org

Now we will:

– Configure Samba

On Ubuntu,

$ mkdir share
$ sudo chmod 777 share/

$ sudo nano /etc/samba/smb.conf

At the bottom of the file

[SHARE]
path = /home/apon/share
read only = no
browsable = yes
public = yes
writable = yes
create mask = 0644
directory mask = 0755

$ sudo restart smbd

On your mac,

Finder > Go > Connect to server
smb://your-ip-of-the-server

login as guest… and you're good

Samba


samba