



WRF in the Cloud

Using Amazon Web Services (AWS)

KELLY WERNER, DAVE GILL

Today's Agenda

- ▶ Presentation:
 - ▶ Introduction to cloud computing
 - ▶ Creating a cloud environment for setting up and running WRF
- ▶ Practice Session
 - ▶ Follow steps to create your own instance/environment
 - ▶ Follow steps to run WPS and WRF (including visualization)
- ▶ Questions and Follow-up Session
- ▶ **Gather feedback from you!**

What is cloud computing?

- ▶ The utilization of remote servers hosted on the internet to store, manage, and process data
 - ▶ Instead of using a local server or personal computer
 - ▶ Easy access to, and sharing of data and resources
 - ▶ May be only available to a single organization (private), may be public, or a hybrid of private/public

Pros & Cons of Cloud Computing

Pros

- ▶ Cuts cost:
 - ▶ No capital cost of hardware
 - ▶ No ongoing administrative costs
 - ▶ No idle assets
- ▶ Access to latest compute architectures and tools
- ▶ Avoid limitations of shared HPC clusters
- ▶ Elastic resource availability
- ▶ Easier collaboration/sharing

Cons

- ▶ Adds cost:
 - ▶ Computing charges
 - ▶ Data transfer charges
 - ▶ Storage charges
- ▶ Large output volumes on remote systems
- ▶ Limited or cumbersome visualization
- ▶ Steep learning curve
- ▶ Charges can accrue when activity not cancelled or suspended

Cloud Service Providers (CSPs): NCAR Collaborations

▶ **Amazon Web Services (AWS):**

<https://aws.amazon.com>

- ▶ Own/manage their own cloud server hardware
- ▶ Highly documented
- ▶ Many supported related applications



▶ **Scala Computing:**

<https://scalacomputing.com>

- ▶ Their server provides access to well-known HPC CSPs and their hardware
- ▶ Manage all processes through their local server
- ▶ Limited documentation
- ▶ Good for repeated and/or regularly-scheduled work



▶ **Penguin Computing:**

<https://www.penguincomputing.com>

- ▶ Own/manage their own cloud server
- ▶ Limited documentation



▶ **Rescale Computing:**

<https://www.rescale.com>

- ▶ Provide access to various HPC cloud services
- ▶ Manage processes through their local server
- ▶ Some documentation



AWS Computing Applications

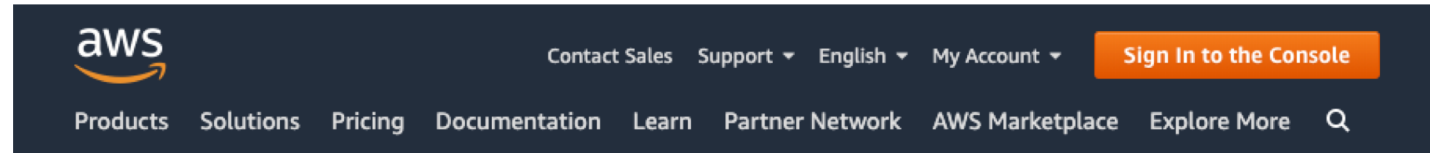
- ▶ **AWS Elastic Compute Cloud (EC2)** : <https://aws.amazon.com/ec2/>
 - ▶ Secure and resizable compute cloud platform
- ▶ **AWS Command Line Interface (CLI)** : <https://aws.amazon.com/cli/>
 - ▶ Alternative to web interface
 - ▶ Usage: Install the tool and then use a terminal window (Unix environment) to navigate and compute
- ▶ **AWS S3** : <https://aws.amazon.com/s3/>
 - ▶ Cloud storage facility
- ▶ **AWS ParallelCluster** : <https://aws.amazon.com/blogs/opensource/aws-parallelcluster/>
 - ▶ Compute server cluster management tool
 - ▶ Enables deployment & management of HPC clusters in the cloud

Steps to Run WRF in the Cloud (using AWS)

- ▶ 1. Logging-in to AWS
- ▶ 2. Machine image selection
- ▶ 3. Instance type selection
- ▶ 4. Configure instance
- ▶ 5. Key pair preparation
- ▶ 6. Launch instance
- ▶ 7. Access instance environment
- ▶ 8. Configure WRF environment

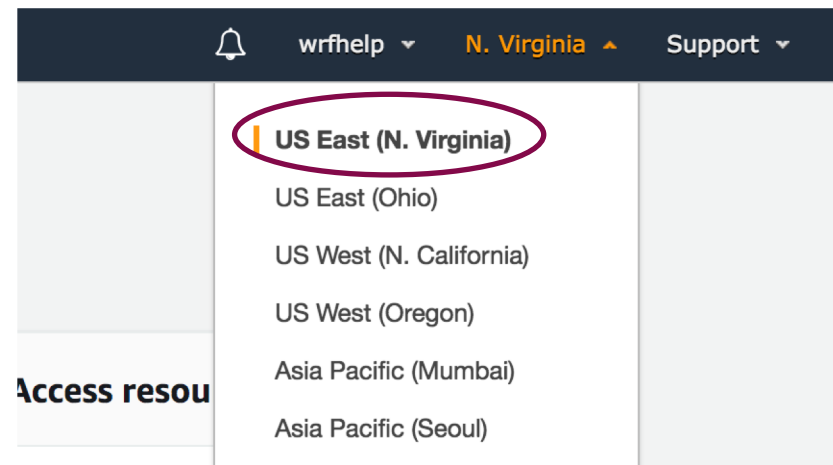
1. Logging-in to AWS: Sign-in and Choosing a Region

- ▶ <https://aws.amazon.com>



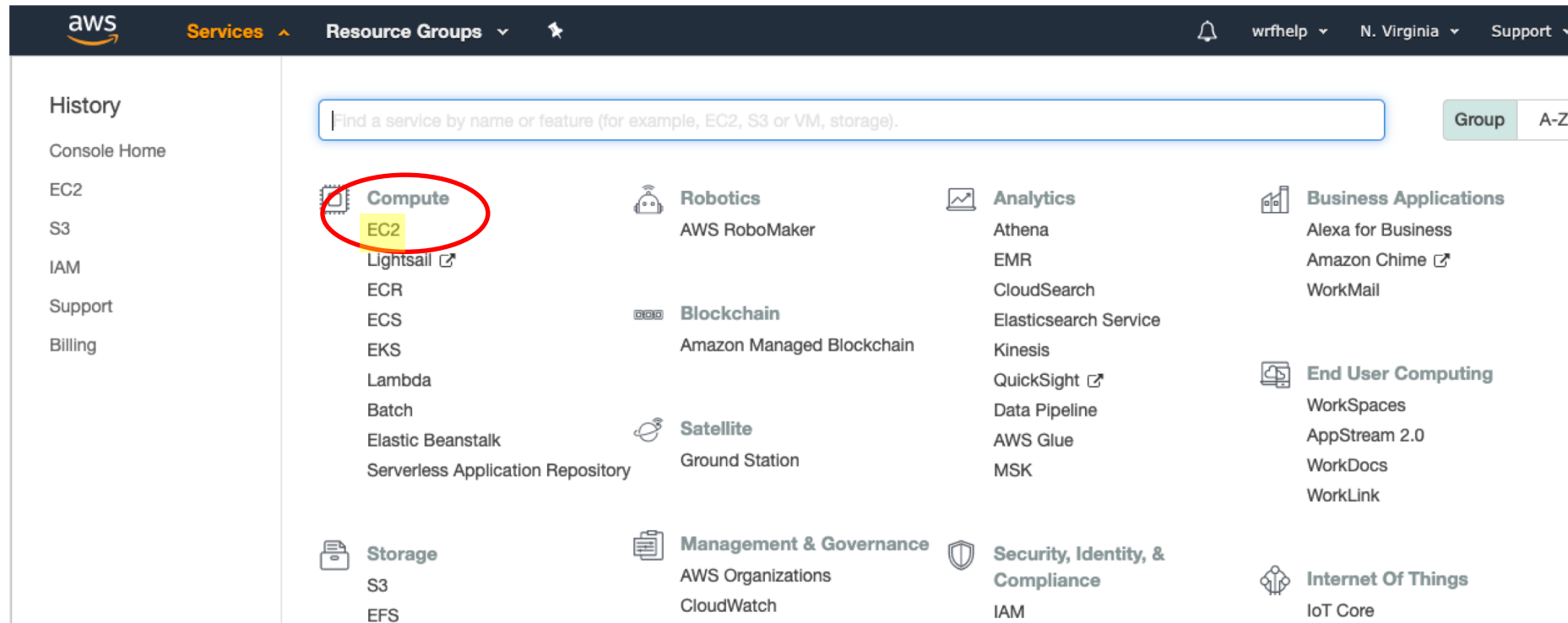
- ▶ Follow tutorial instructions and use the IAM (Identity and Access Management) User Name and Password provided.

- ▶ Choosing Regions
 - ▶ Regional endpoint (URL entry point for a web service)
 - ▶ Purpose of reducing data latency
 - ▶ US East (N. Virginia)



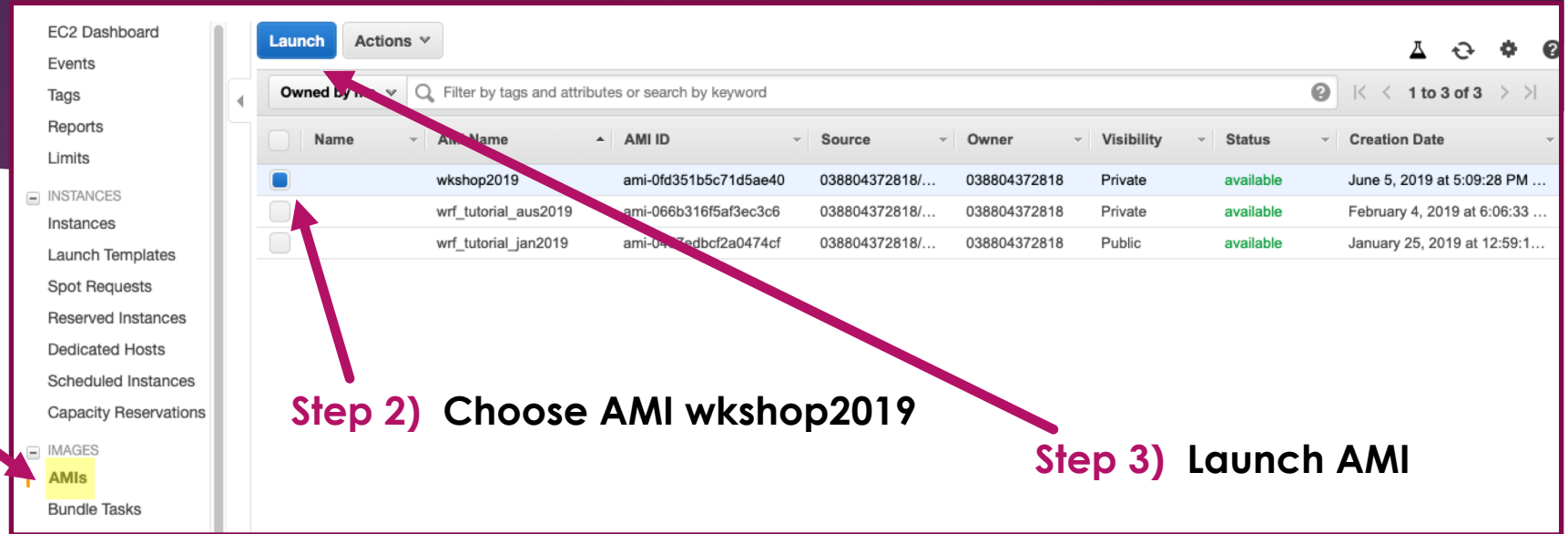
1. Logging-in to AWS: Navigate to EC2

- Navigate to Elastic Compute Cloud (EC2 – under "Compute")



2. Machine image selection

Step 1) Navigate to AMI's



The screenshot shows the AWS Management Console's EC2 Dashboard. On the left sidebar, under the 'IMAGES' section, 'AMIs' is highlighted. In the main content area, a table lists available AMIs. The first row, 'wkshop2019', is selected. A 'Launch' button is visible at the top left of the table. Three red arrows indicate the steps: one from 'Step 1) Navigate to AMI's' to the 'AMIs' link, one from 'Step 2) Choose AMI wkshop2019' to the selected row, and one from 'Step 3) Launch AMI' to the 'Launch' button.

Name	AMI Name	AMI ID	Source	Owner	Visibility	Status	Creation Date
<input checked="" type="checkbox"/>	wkshop2019	ami-0fd351b5c71d5ae40	038804372818/...	038804372818	Private	available	June 5, 2019 at 5:09:28 PM ...
<input type="checkbox"/>	wrf_tutorial_au2019	ami-066b316f5af3ec3c6	038804372818/...	038804372818	Private	available	February 4, 2019 at 6:06:33 ...
<input type="checkbox"/>	wrf_tutorial_jan2019	ami-0403edbcf2a0474cf	038804372818/...	038804372818	Public	available	January 25, 2019 at 12:59:1...

Step 2) Choose AMI wkshop2019

Step 3) Launch AMI

► Amazon Machine Image (AMI)

- Snapshot of a pre-configured environment
- Useful for saving work and sharing with others
- AMI Name: **wkshop2019**
 - Includes OS – Amazon Linux AMI (HVM), SSD Volume Type, 64-bit
 - Includes libraries, compiled WPS/WRF code & NCL

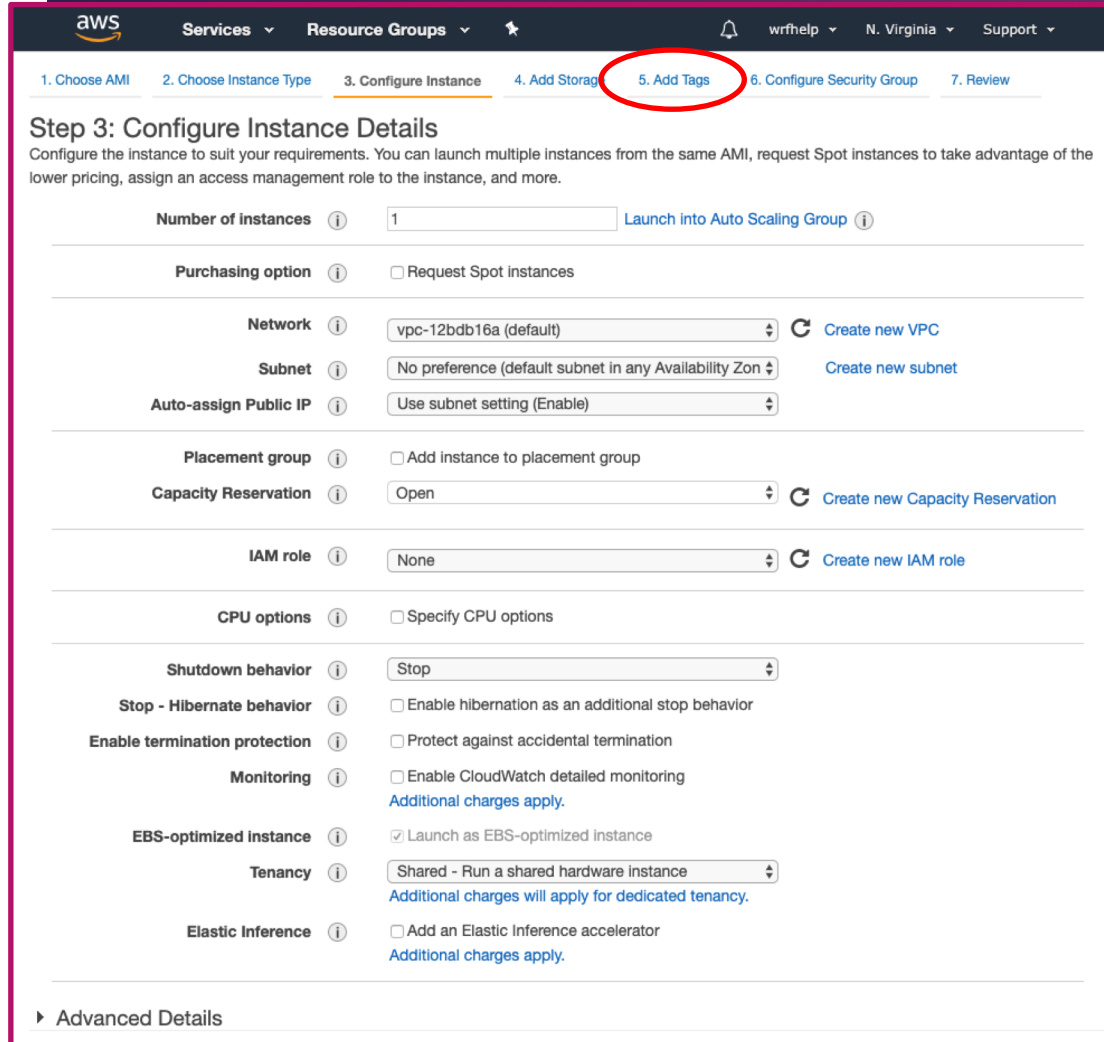
3. Instance Type Selection

	Family	Type	vCPUs	Memory (GiB)	Instance Storage (GB)	EBS-Optimized Available	Network Performance	IPv6 Support
<input type="checkbox"/>	Compute optimized	c5.18xlarge	72	144	EBS only	Yes	25 Gigabit	Yes
<input type="checkbox"/>	Compute optimized	c4.large	2	3.75	EBS only	Yes	Moderate	Yes
<input type="checkbox"/>	Compute optimized	c4.xlarge	4	7.5	EBS only	Yes	High	Yes
<input type="checkbox"/>	Compute optimized	c4.2xlarge	8	15	EBS only	Yes	High	Yes
<input checked="" type="checkbox"/>	Compute optimized	c4.4xlarge	16	30	EBS only	Yes	High	Yes
<input type="checkbox"/>	Compute optimized	c4.8xlarge	36	60	EBS only	Yes	10 Gigabit	Yes
<input type="checkbox"/>	FPGA instances	f1.2xlarge	8	122	1 x 470 (SSD)	Yes	Up to 10 Gigabit	Yes

<https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/Instances.html>

- ▶ **Instance:** Virtual Server
- ▶ Different instance types determine the hardware
 - ▶ Varying compute, memory, & storage capabilities
- ▶ Type for this class: **c4.4xlarge**
 - ▶ 16 vCPUs (8 CPUs)
 - ▶ Reasonable size/speed for basic WRF run

4. Configure Instance Details



aws Services Resource Groups

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, assign an access management role to the instance, and more.

Number of instances 1 [Launch into Auto Scaling Group](#)

Purchasing option ☐ Request Spot instances

Network vpc-12bdb16a (default) [Create new VPC](#)

Subnet No preference (default subnet in any Availability Zone) [Create new subnet](#)

Auto-assign Public IP Use subnet setting (Enable)

Placement group ☐ Add instance to placement group

Capacity Reservation Open [Create new Capacity Reservation](#)

IAM role None [Create new IAM role](#)

CPU options ☐ Specify CPU options

Shutdown behavior Stop

Stop - Hibernate behavior ☐ Enable hibernation as an additional stop behavior

Enable termination protection ☐ Protect against accidental termination

Monitoring ☐ Enable CloudWatch detailed monitoring
[Additional charges apply.](#)

EBS-optimized instance ☒ Launch as EBS-optimized instance

Tenancy Shared - Run a shared hardware instance
[Additional charges will apply for dedicated tenancy.](#)

Elastic Inference ☐ Add an Elastic Inference accelerator
[Additional charges apply.](#)

► Advanced Details

- ▶ Can launch multiple instances from one AMI
- ▶ Configure behavior (e.g., shutdown behavior)
- ▶ Add storage
- ▶ Add tags (name your instance)
- ▶ Configure security options
- ▶ For this tutorial, only use “Add Tags”

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 = Webserver. 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)
Name	your_name_goes_here!
Add another tag (Up to 50 tags maximum)	

5. Key Pair Preparation

- ▶ AWS uses public-key cryptography to encrypt/decrypt login information
 - ▶ AWS uses a *public* key to encrypt data
 - ▶ Recipient uses *private* key to decrypt data
- ▶ User must download a key pair
 - ▶ Change permissions (`> chmod 600 key_pair.pem`)
- ▶ Specify key pair when launching an instance (command-line)
 - ▶ `> ssh -i key_pair.pem ec2-user@Public_IP_address`

6. Launch Instance

► After downloading a key pair -> [Launch Instances](#) -> [View Instances](#)

<div>Launch Instance ▼ Connect Actions ▼</div> <div>Filter by tags and attributes or search by keyword ? ⏪ < 1</div>									
<input type="checkbox"/>	Name	Instance ID	Instance Type	Availability Zone	Instance State	Status Checks	Alarm Status	Public DNS (IPv4)	IPv4 Public IP
<input type="checkbox"/>	user_1	i-00227d76497b432...	t2.micro	us-east-1b	● stopped		None		-
<input type="checkbox"/>	user_2	i-01da0c26c532f66c0	t2.micro	us-east-1b	● stopped		None		-
<input checked="" type="checkbox"/>	user_3	i-01e8ddd1f22457e97	t2.micro	us-east-1b	● running	Initializing	None	ec2-54-152-48-166.co...	54.152.48.166
<input type="checkbox"/>	user_4	i-01fccdc6acd3e85ed	t2.micro	us-east-1b	● stopped		None		-
<input type="checkbox"/>	user_5	i-022cd9df4105eee33	t2.micro	us-east-1b	● stopped		None		-

7. Instance environment access

The screenshot displays the AWS Management Console interface for EC2 instances. At the top, there are buttons for 'Launch Instance', 'Connect', and 'Actions'. Below these is a search bar and a table of instances. The table has columns for Name, Instance ID, Instance Type, Availability Zone, Instance State, Status Checks, Alarm Status, Public DNS (IPv4), IPv4 Public IP, and IPv6 Public IP. Instance 'user_3' is highlighted in blue, and its IPv4 Public IP, 54.152.48.166, is circled in red. Below the table, the details for instance 'user_3' are shown, including its Instance ID, state (running), type (t2.micro), and availability zone (us-east-1b). The IPv4 Public IP is also circled in red in the details section.

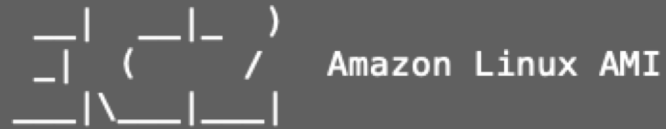
Name	Instance ID	Instance Type	Availability Zone	Instance State	Status Checks	Alarm Status	Public DNS (IPv4)	IPv4 Public IP	IPv6 Public IP
user_1	i-00227d76497b432...	t2.micro	us-east-1b	stopped		None		-	-
user_2	i-01da0c26c532f66c0	t2.micro	us-east-1b	stopped		None		-	-
user_3	i-01e8ddd1f22457e97	t2.micro	us-east-1b	running	2/2 checks ...	None	ec2-54-152-48-166.co...	54.152.48.166	-
user_4	i-01fcdcc6acd3e85ed	t2.micro	us-east-1b	stopped		None		-	-
user_5	i-022cd9df4105eee33	t2.micro	us-east-1b	stopped		None		-	-
user_6	i-02eb27425c37a2dae	t2.micro	us-east-1b	stopped		None		-	-
user_7	i-035ccfc74bad182f	t2.micro	us-east-1b	stopped		None		-	-
user_8	i-049f5380f7c6a9ae2	t2.micro	us-east-1b	stopped		None		-	-

Instance: i-01e8ddd1f22457e97 (user_3) Public DNS: ec2-54-152-48-166.compute-1.amazonaws.com

Description		Status Checks	Monitoring	Tags
Instance ID	i-01e8ddd1f22457e97			
Instance state	running			
Instance type	t2.micro			
Elastic IPs				
Availability zone	us-east-1b			
Public DNS (IPv4)	ec2-54-152-48-166.compute-1.amazonaws.com			
IPv4 Public IP	54.152.48.166			
IPv6 IPs				
Private DNS	ip-172-31-95-125.ec2.internal			
Private IPs	172.31.95.125			

- ▶ Create a new directory on your local machine (terminal window) from where you will work
- ▶ Place *key_pair.pem* file in that directory (and change permissions)
- ▶ Locate the IPv4 Public IP (2 locations)
 - ▶ This changes each time you stop and re-start the instance
- ▶ SSH into instance environment (e.g., `> ssh -X -i key_pair.pem ec2-user@54.152.48.166`)

8. Configure WRF environment



```
https://aws.amazon.com/amazon-linux-ami/2018.03-release-notes/  
22 package(s) needed for security, out of 29 available  
Run "sudo yum update" to apply all updates.  
/home/ec2-user>ls  
graphics  input_data  libraries  tar_files  WPS  WPS_GEOG  WRF  
/home/ec2-user>
```

- ▶ Configure environment – set up shell preferences, environment scripts (e.g., .cshrc)
- ▶ Install compiler/library packages (e.g., gcc, m4, make, perl, which, X11, etc.)
 - ▶ > sudo yum install *package*
- ▶ Install libraries needed for WRF/WPS (netCDF, mpich, jasper, libpng, zlib)
- ▶ Compile WRF/WPS (and any post-processors)
- ▶ Import input data (static geographic fields, meteorological first-guess data)

Importing Data or Files

- ▶ SCP Option

- ▶ `scp -i key_pair.pem file_name ec2-user@public_IP:/home/ec2-user`

- ▶ Obtain data from an S3 storage source

- (one file)** `aws s3 cp <s3 bucket name/path/file> <local path>`

- (e.g.): `aws s3 cp s3://wpsgeog/albedo_modis/00001-01200.00001-01200 .`

- (multiple files or directories)** `aws s3 cp <s3://bucket_name/path> <local path> --recursive`

- (e.g.): `aws s3 cp s3://wpsgeog/ /home/ec2-user/wps_geog --recursive`

- ▶ Use 'git' to aquire WRF & WPS source code

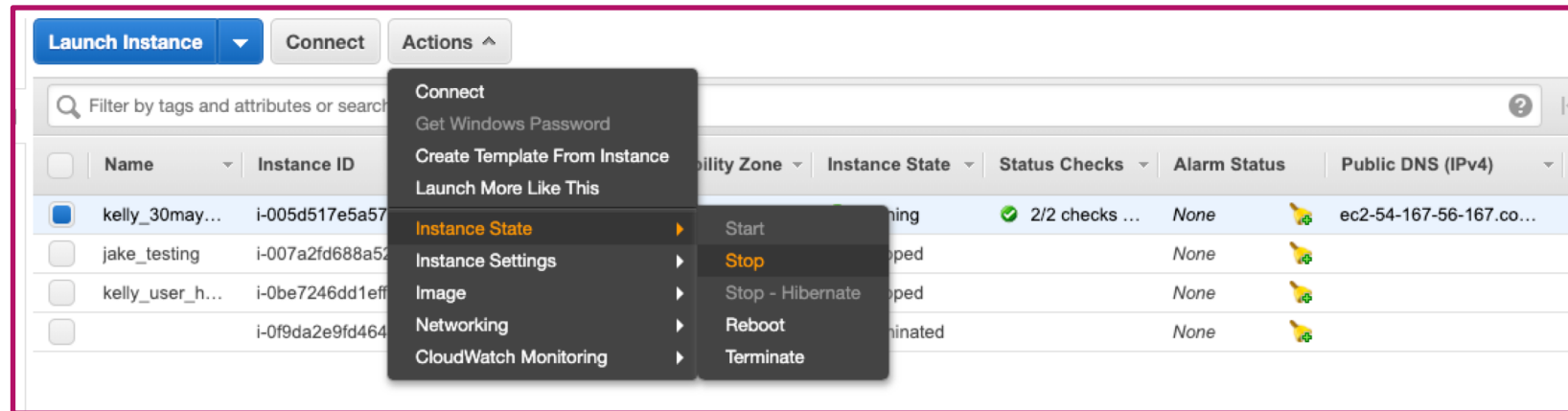
- ▶ `git clone https://github.com/wrf-model/WRF.git`

Running WPS/WRF

- ▶ Run WPS executables serially
 - ▶ (e.g.) ./geogrid.exe
- ▶ Run real and wrf with distributed memory (parallel dmpar)
 - ▶ (e.g.) mpirun -np 8 ./wrf.exe
- ▶ Visualization
 - ▶ NCL and ncview available in workshop AMI (uses X11)

- ▶ Job 'stop' or 'terminate'

- ▶ **Stop:** like “pausing” the instance, or logging out of a server – will be just as it was when you return (start)
- ▶ **Terminate:** kills the instance forever



Useful Resources

- ▶ Using AWS ParallelCluster:
<https://jiaweizhuang.github.io/blog/aws-hpc-guide/>
 - WRF examples with ParallelCluster
- ▶ AWS Research Cloud Program
<https://aws.amazon.com/government-education/research-and-technical-computing/research-cloud-program/>
- ▶ Workshop Mini-tutorial Web Page:
http://www2.mmm.ucar.edu/wrf/OnLineTutorial/wrf_in_cloud_aws_tutorial_2019.php