

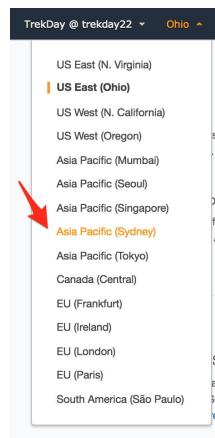
Session 1 - Introduction

In this session we will get introduced to AWS and create our first virtual server.

Take the following steps, one at a time. There is plenty of time to complete the session and ask any questions along the way.

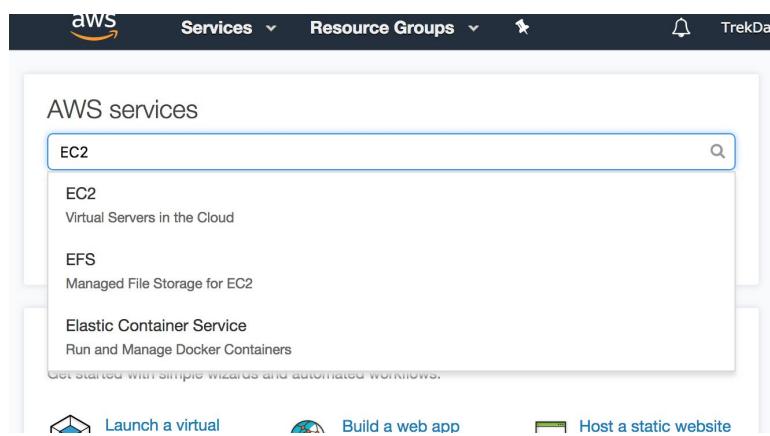
Getting Started

- Log into the AWS console using the details provided on your Credentials Note
- Set the region to Sydney (stay in this region today, check from time to time)



Create a Virtual Server - EC2 Instance

- Navigate to EC2



A screenshot of the AWS Services dashboard. The top navigation bar shows the AWS logo, 'Services' dropdown, 'Resource Groups' dropdown, a bell icon, and the text 'TrekDay'. Below the navigation is a search bar with 'EC2' typed into it. A sidebar on the left lists 'AWS services' with 'EC2' selected, described as 'Virtual Servers in the Cloud'. Other listed services include EFS (Managed File Storage for EC2) and Elastic Container Service (Run and Manage Docker Containers). At the bottom of the sidebar, there is a link to 'Get started with simple wizards and automated workflows.' Below the sidebar are three buttons: 'Launch a virtual' (with a blue house icon), 'Build a web app' (with a green gear icon), and 'Host a static website' (with a yellow folder icon).

- Click "Launch Instance"

Resources

You are using the following Amazon EC2 resources in the Asia Pacific (Sydney) region:

0 Running Instances	0 Elastic IPs
0 Dedicated Hosts	0 Snapshots
0 Volumes	0 Load Balancers
0 Key Pairs	1 Security Groups
0 Placement Groups	

Learn more about the latest in AWS Compute from AWS re:Invent 2017 by viewing the [EC2 Videos](#).

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 Asia Pacific (Sydney) region

- Step 1: Select an Amazon Machine Image (AMI)

Take a look at all the different Amazon Machine Images that are available. There are many variants of Windows and Linux.

- Select "Microsoft Windows Server 2016 Base"

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)

Free tier eligible Support available from Canonical (<http://www.ubuntu.com/cloud/services>). [View details](#)

Root device type: ebs Virtualization type: hvm

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- and PostgreSQL-compatible, enterprise-class database at 1/10th the cost of commercial databases. [Learn more about RDS](#)

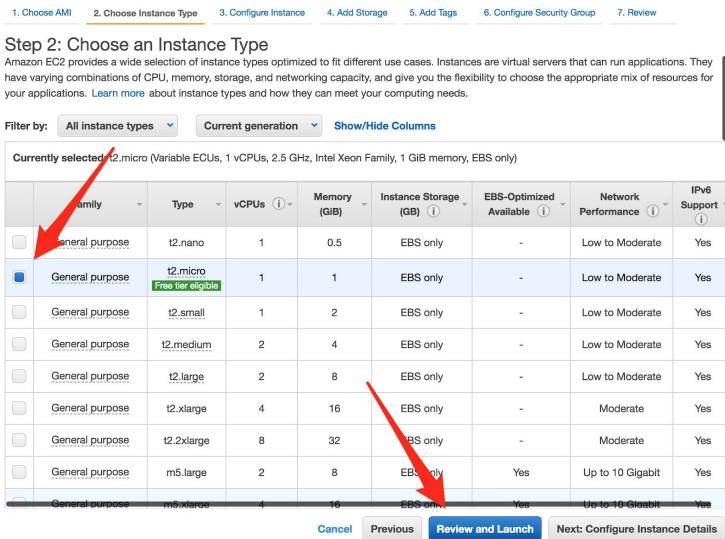
Launch a database using RDS

 Microsoft Windows Server 2016 Base - ami-48d6122a	Select
Windows Free tier eligible	64-bit
Microsoft Windows 2016 Datacenter edition. [English]	
Root device type: ebs Virtualization type: hvm	
 Deep Learning AMI (Ubuntu) Version 4.0 - ami-e723985	Select
Free tier eligible	64-bit
Latest versions of deep learning frameworks pre-installed in separate virtual environments: MXNet, TensorFlow, Caffe, Caffe2, PyTorch, Theano, CNTK, Keras	
Root device type: ebs Virtualization type: hvm	
 Deep Learning AMI (Amazon Linux) Version 4.0 - ami-46c23924	Select
Amazon Linux Free tier eligible	64-bit
Latest versions of deep learning frameworks pre-installed in separate virtual environments: MXNet, TensorFlow, Caffe, Caffe2, PyTorch, Theano, CNTK, Keras	
Root device type: ebs Virtualization type: hvm	

- Step 2: Choose an Instance Type

Take a look at all the different Instance Types available. There are sizes for almost any workload.

- Select "t2.micro"



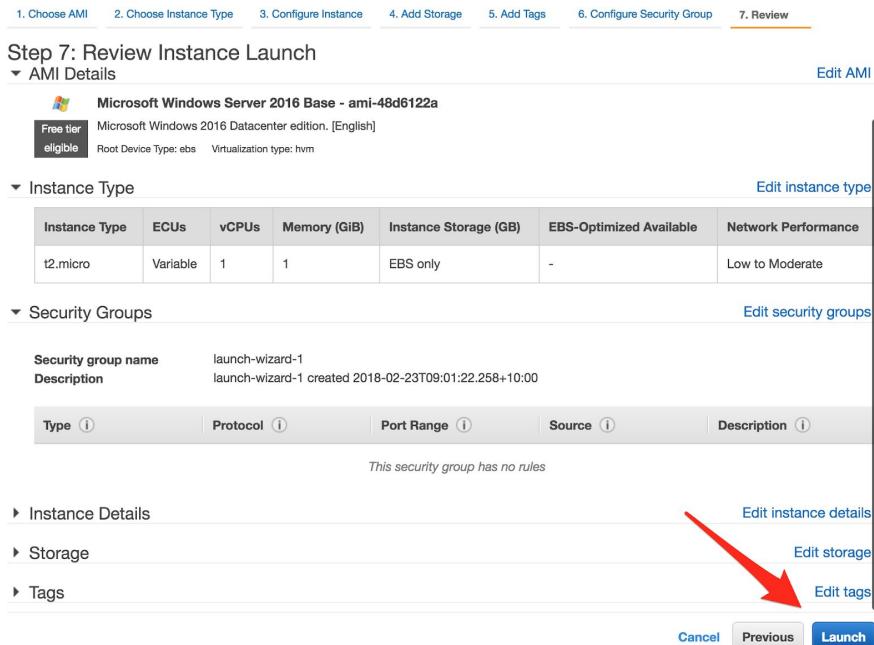
Step 2: Choose an Instance Type

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.

Family	Type	vCPUs	Memory (GiB)	Instance Storage (GB)	EBS-Optimized Available	Network Performance	IPv6 Support
General purpose	t2.nano	1	0.5	EBS only	-	Low to Moderate	Yes
General purpose	t2.micro	1	1	EBS only	-	Low to Moderate	Yes
General purpose	t2.small	1	2	EBS only	-	Low to Moderate	Yes
General purpose	t2.medium	2	4	EBS only	-	Low to Moderate	Yes
General purpose	t2.large	2	8	EBS only	-	Low to Moderate	Yes
General purpose	t2.xlarge	4	16	EBS only	-	Moderate	Yes
General purpose	t2.2xlarge	8	32	EBS only	-	Moderate	Yes
General purpose	m5.large	2	8	EBS only	Yes	Up to 10 Gigabit	Yes
General purpose	m5.xlarge	4	16	EBS only	Yes	Up to 10 Gigabit	Yes

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

- Select “Review and Launch”, then “Launch”



Step 7: Review Instance Launch

AMI Details

Microsoft Windows Server 2016 Base - ami-48d6122a

Free tier eligible Microsoft Windows 2016 Datacenter edition. [English]
Root Device Type: ebs Virtualization type: hvm

Instance Type

Instance Type	ECUs	vCPUs	Memory (GiB)	Instance Storage (GB)	EBS-Optimized Available	Network Performance
t2.micro	Variable	1	1	EBS only	-	Low to Moderate

Security Groups

Security group name: launch-wizard-1
Description: launch-wizard-1 created 2018-02-23T09:01:22.258+10:00

Type	Protocol	Port Range	Source	Description
This security group has no rules				

Instance Details

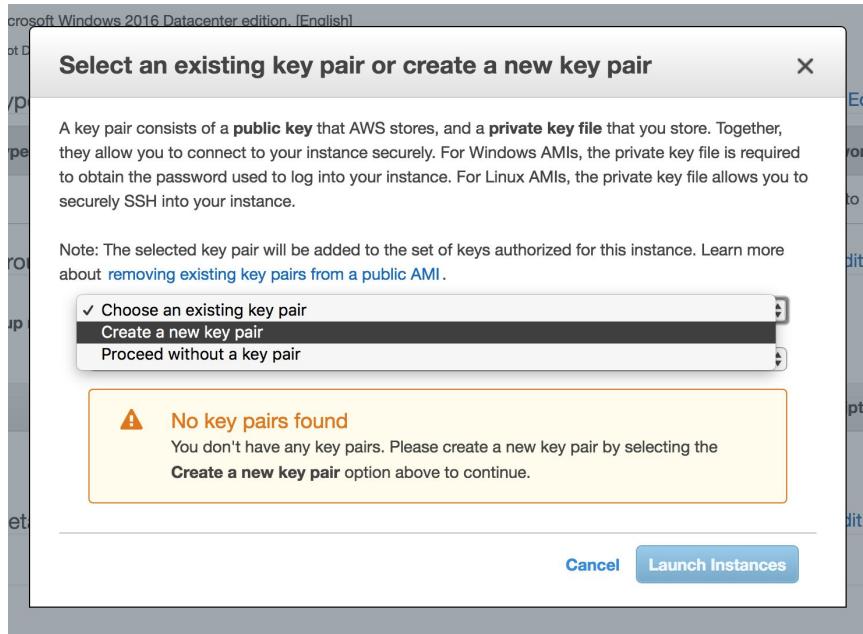
Storage

Tags

Cancel Previous **Launch**

- Popup: "Select an existing key pair or create a new key pair"

- Select: "Create a new key pair"



- Provide: Type a name, then "Download Key Pair" and keep it safe, we will use this a couple of times today.

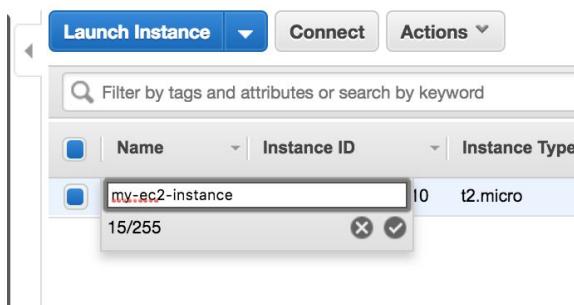
- Select: "Launch instances"

- Scroll down and select "View Instances"

- Review the "Instances" page.

This is where you will see all the instances you have launched. We will be coming back to this list several times today.

- Set the name of the instance, by clicking the pencil in the Name column. Set the name to "my-ec2-instance".



Tip: Make sure you select the 'tick' when changing the name, else it won't take.

- Review the "Description" tab for the running instance.
- Look at the "Tags" tab, notice that the name is set as a tag.



The screenshot shows the 'Tags' tab selected in a AWS interface. A table displays one tag entry:

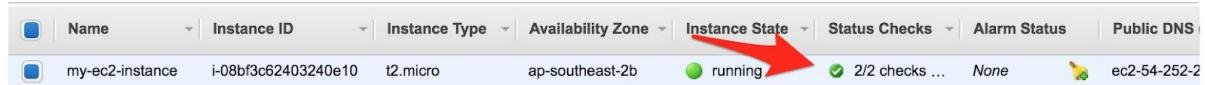
Key	Value	Actions
Name	my-ec2-instance	Edit Delete Add

[Add/Edit Tags](#)

[Hide Column](#)

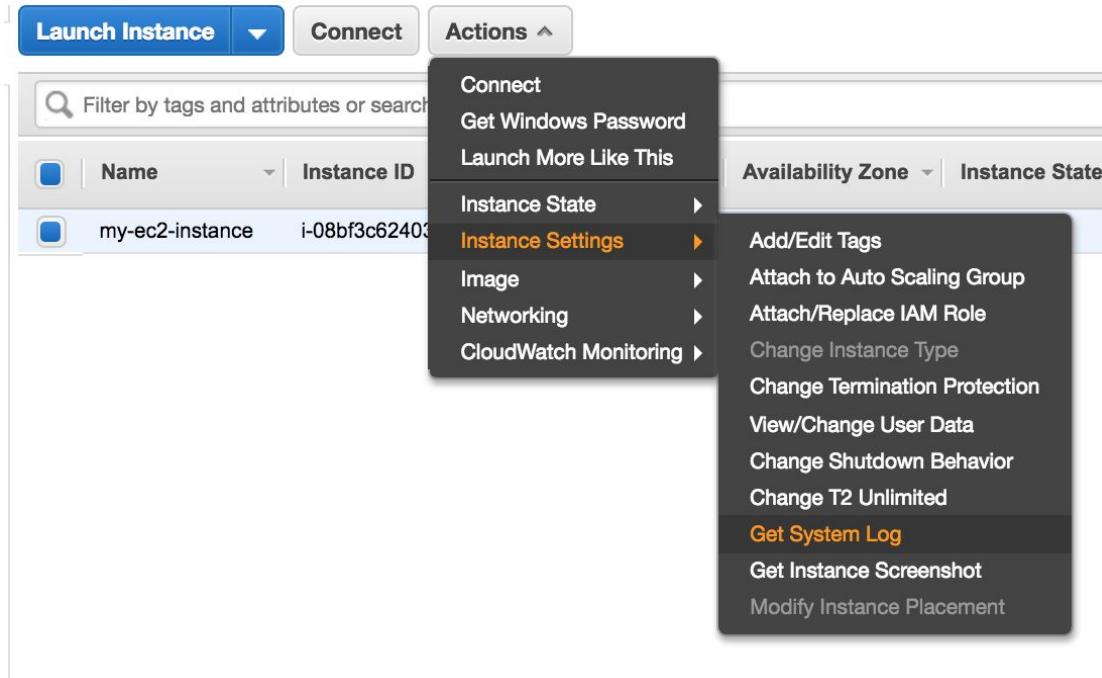
Connect to the Virtual Server - EC2 Instance

- Wait for the "Status Checks" 2/2. This means that the host and guest OS's are passing status checks.

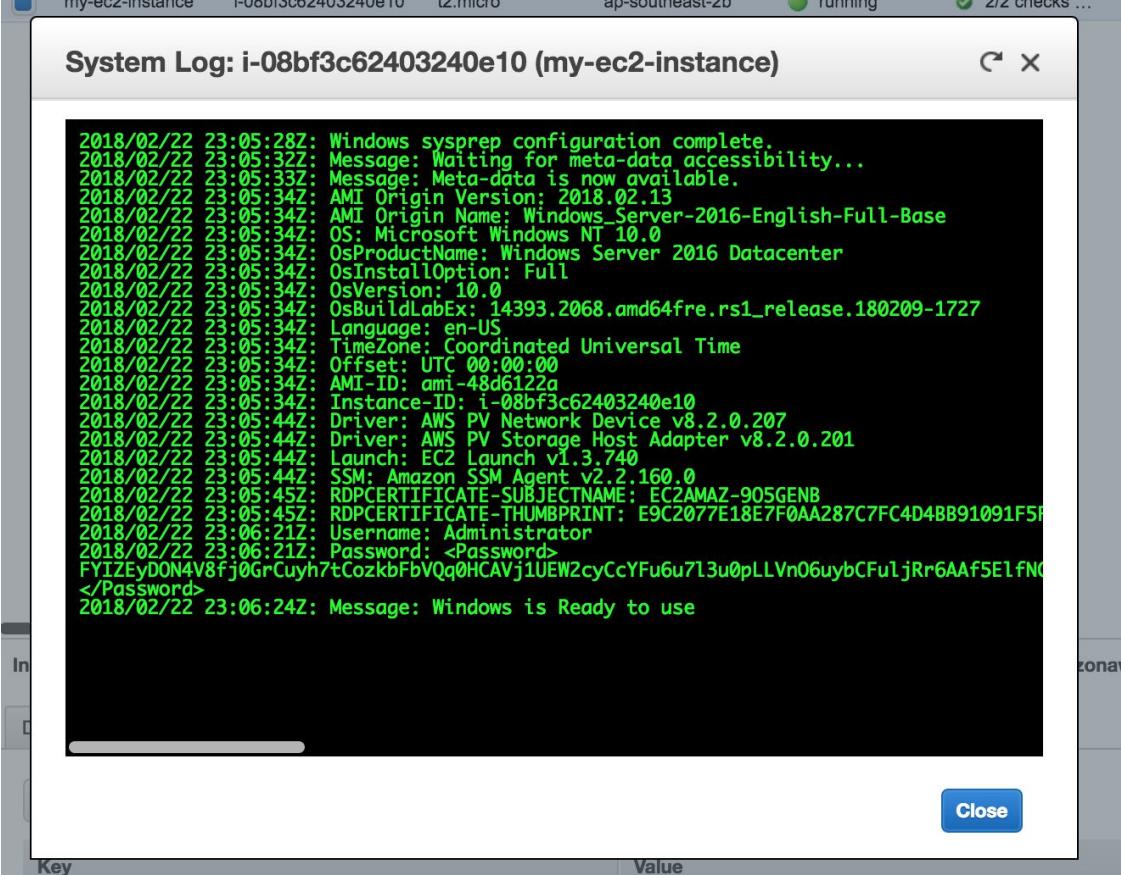


- Let's check that the Windows server is up and running:

- With the instance selected, select "Actions", "Instance Settings", "Get System Log"



- You should be able to see system logs sent from the server.

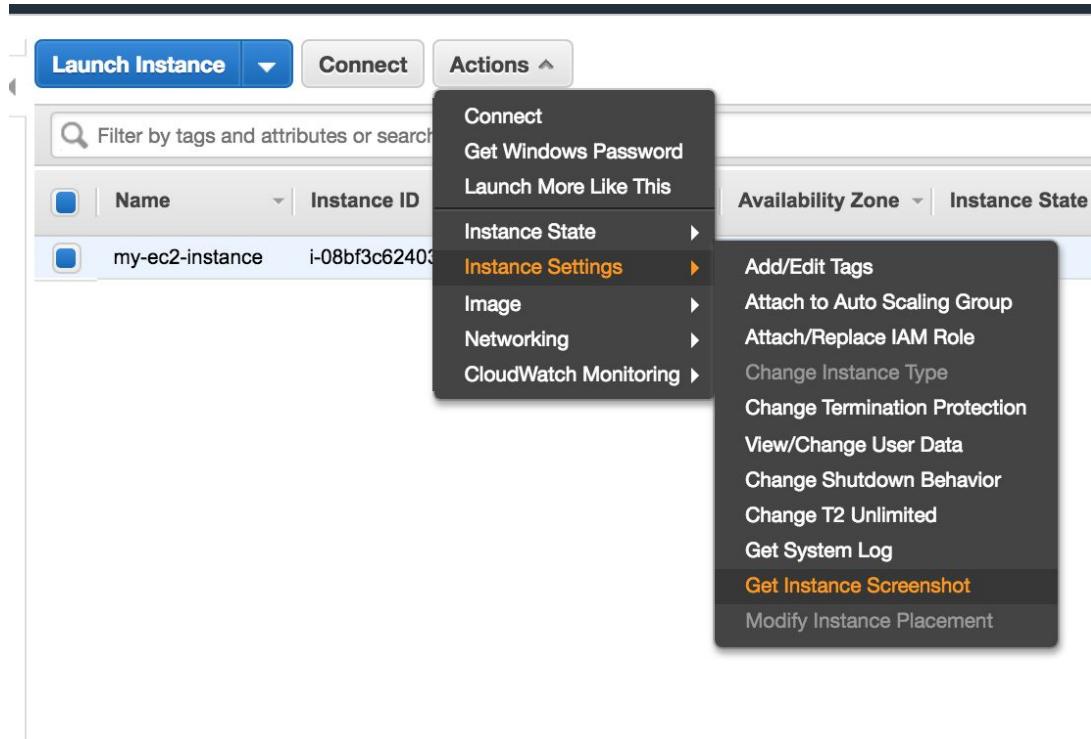


The screenshot shows a modal window titled "System Log: i-08bf3c62403240e10 (my-ec2-instance)". The log output is displayed in a monospaced font. The log entries are as follows:

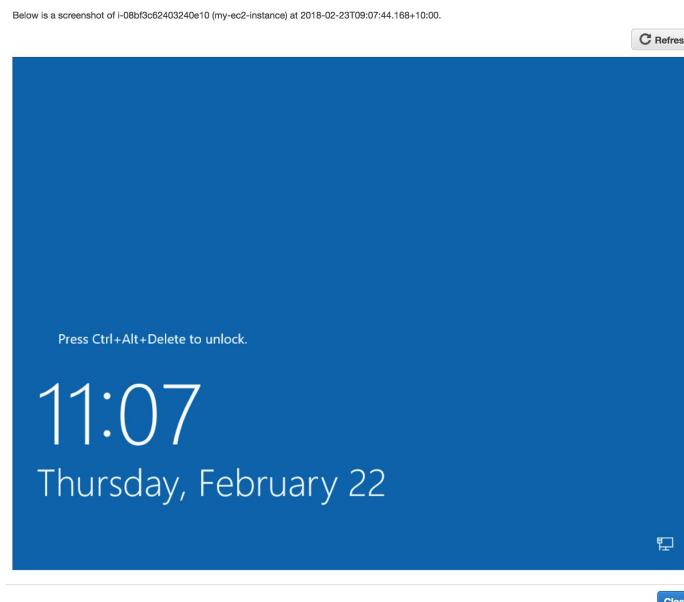
```
2018/02/22 23:05:28Z: Windows sysprep configuration complete.
2018/02/22 23:05:32Z: Message: Waiting for meta-data accessibility...
2018/02/22 23:05:33Z: Message: Meta-data is now available.
2018/02/22 23:05:34Z: AMI Origin Version: 2018.02.13
2018/02/22 23:05:34Z: AMI Origin Name: Windows_Server-2016-English-Full-Base
2018/02/22 23:05:34Z: OS: Microsoft Windows NT 10.0
2018/02/22 23:05:34Z: OsProductName: Windows Server 2016 Datacenter
2018/02/22 23:05:34Z: OsInstallOption: Full
2018/02/22 23:05:34Z: OsVersion: 10.0
2018/02/22 23:05:34Z: OsBuildLabEx: 14393.2068.amd64fre.rs1_release.180209-1727
2018/02/22 23:05:34Z: Language: en-US
2018/02/22 23:05:34Z: TimeZone: Coordinated Universal Time
2018/02/22 23:05:34Z: Offset: UTC 00:00:00
2018/02/22 23:05:34Z: AMI-ID: ami-48d6122a
2018/02/22 23:05:34Z: Instance-ID: i-08bf3c62403240e10
2018/02/22 23:05:44Z: Driver: AWS PV Network Device v8.2.0.207
2018/02/22 23:05:44Z: Driver: AWS PV Storage Host Adapter v8.2.0.201
2018/02/22 23:05:44Z: Launch: EC2 Launch v1.3.740
2018/02/22 23:05:44Z: SSM: Amazon SSM Agent v2.2.160.0
2018/02/22 23:05:45Z: RDPCERTIFICATE-SUBJECTNAME: EC2AMAZ-905GENB
2018/02/22 23:05:45Z: RDPCERTIFICATE-THUMBPRINT: E9C2077E18E7F0AA287C7FC4D4BB91091F5F
2018/02/22 23:06:21Z: Username: Administrator
2018/02/22 23:06:21Z: Password: <Password>
FYIZEyDON4V8fj0GrCuyh7tCozkBFBvQq0HCAVj1UEW2cyCcYFu6u7l3u0pLLVn06uybCFuljRr6AAf5ElfNC
</Password>
2018/02/22 23:06:24Z: Message: Windows is Ready to use
```

- Notice the Password field is encrypted
- Is there a line that says: "Message: Windows is Ready to use"?
- Select "Close"

- With the instance selected, select "Actions", "Instance Settings", "Get Instance Screenshot":



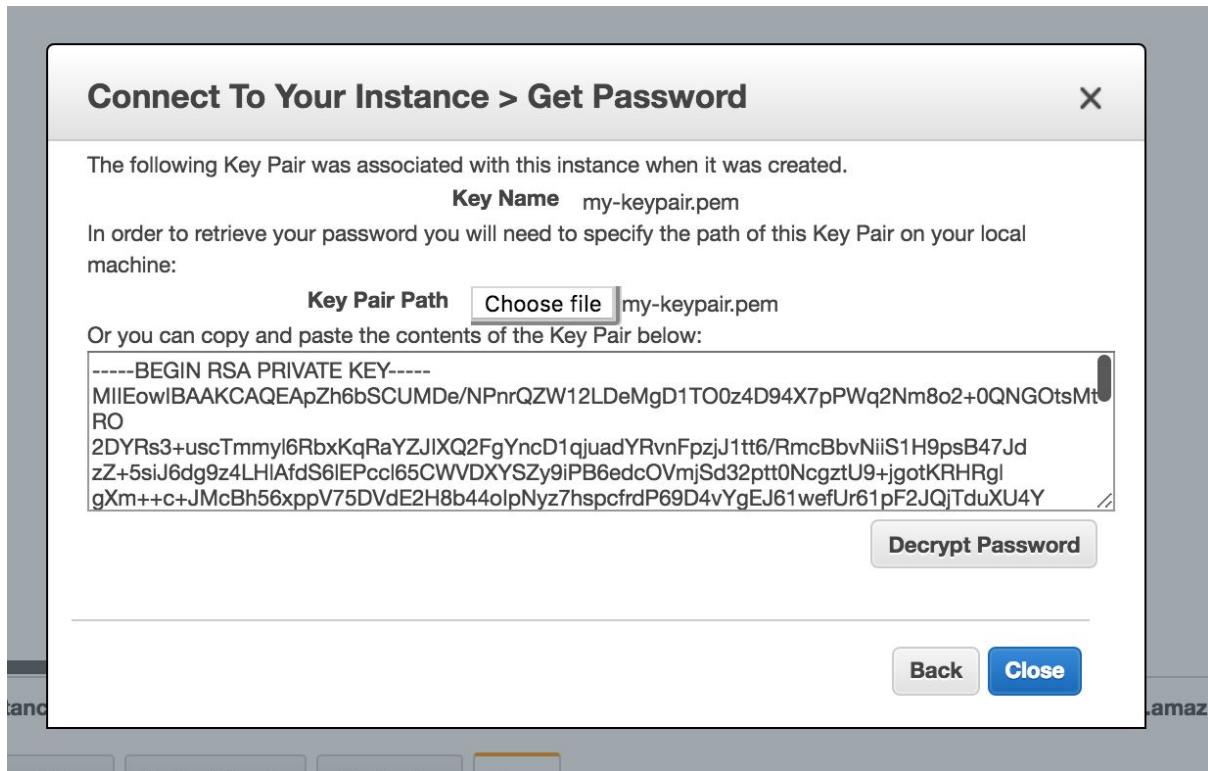
- You should see an image of the screen output of the server.



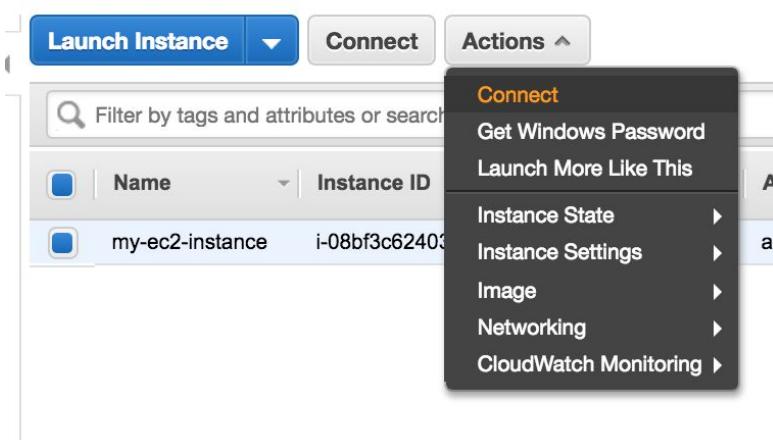
- You can't interact with this image, so lets log in to the server.

- Select "Close"

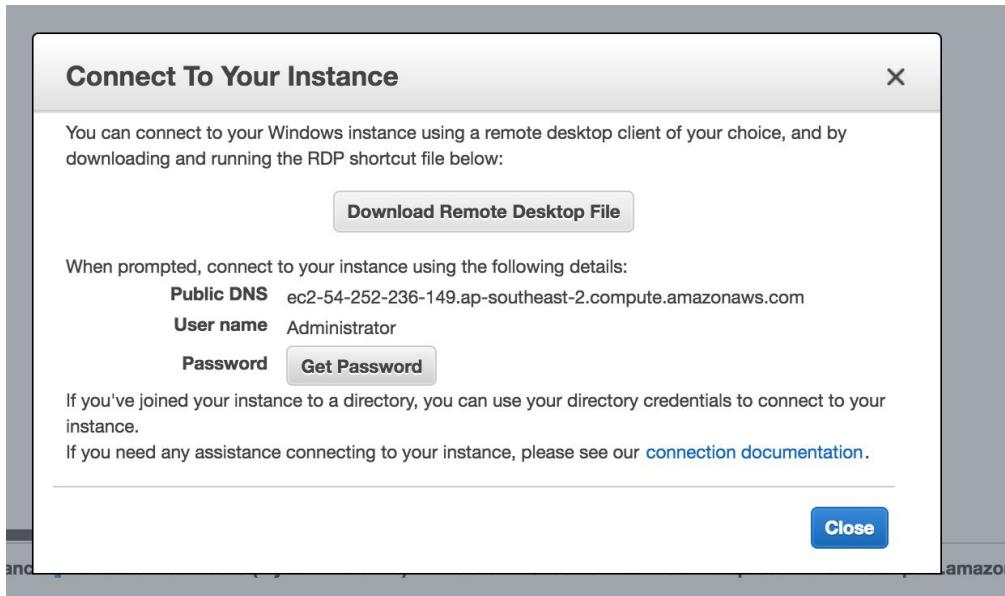
- Let's decrypt the Windows Administrator password:
- With the instance selected, select "Actions", "Get Windows Password"
- Select “Choose file” and browse to the key file you downloaded earlier in this section.
- Select "Decrypt Password"



- Keep the details safe somewhere.
- Again with the instance selected, click "Actions" and select "Connect"



- Popup: "Connect To Your Instance" select "Download Remote Desktop File"



- This should download, automatically open, and configure a Remote Desktop Protocol client to connect to your instance.

If clicking the "Download Remote Desktop File" didn't work for you:

- Open your RDP client, enter the DNS value into the address field
 - When prompted enter the username and password.
- Note: when you connect you may be prompted by security warnings. This is not a production environment, these warnings are ok.

- Back in the AWS console, let's take a look at how our server is doing.

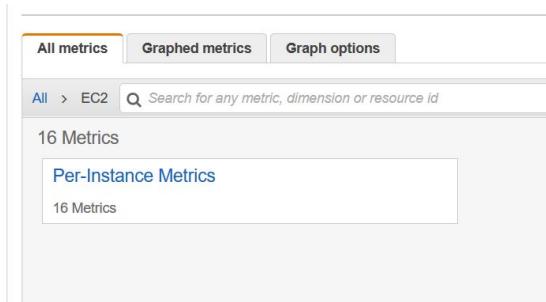
You might want to open this next link in a new tab, it helps to have different AWS services open in different tabs to save time when switching back and forth.

- Navigate to CloudWatch

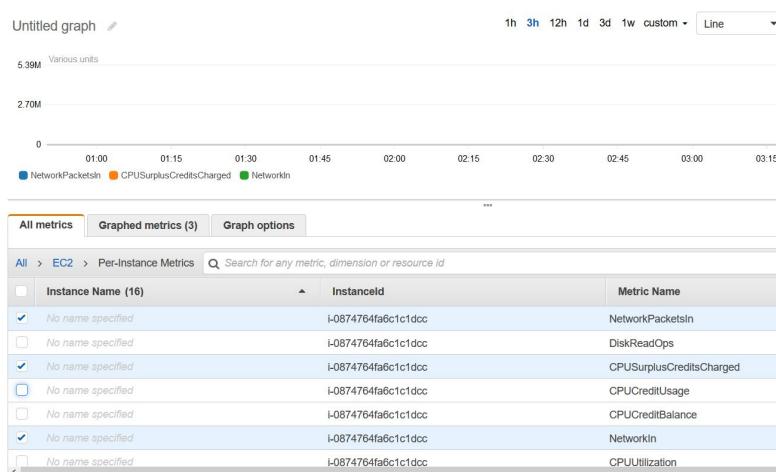
- CloudWatch is the central location of all monitoring and logging in AWS.

- Select “Browse Metrics”

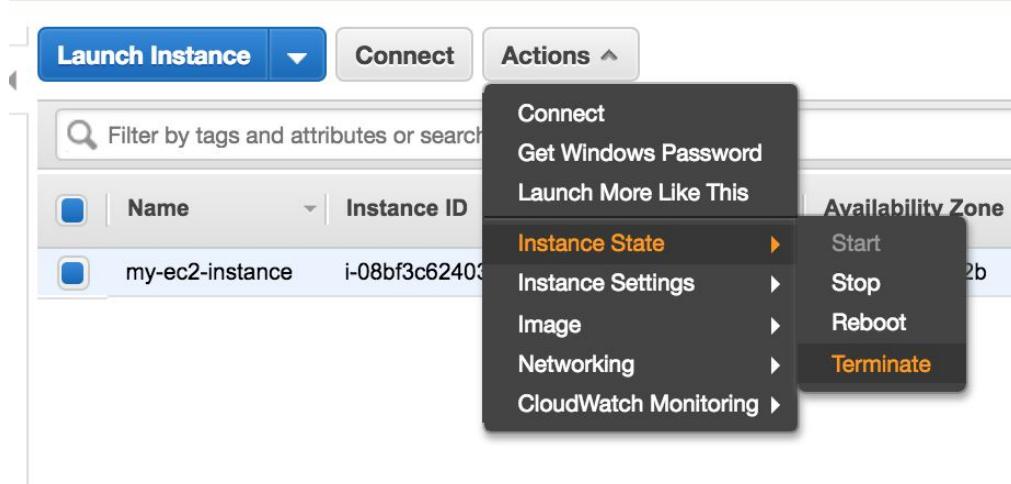
- Towards the bottom, with the tab “All metrics” selected, select “EC2” then “Per-Instance Metrics”



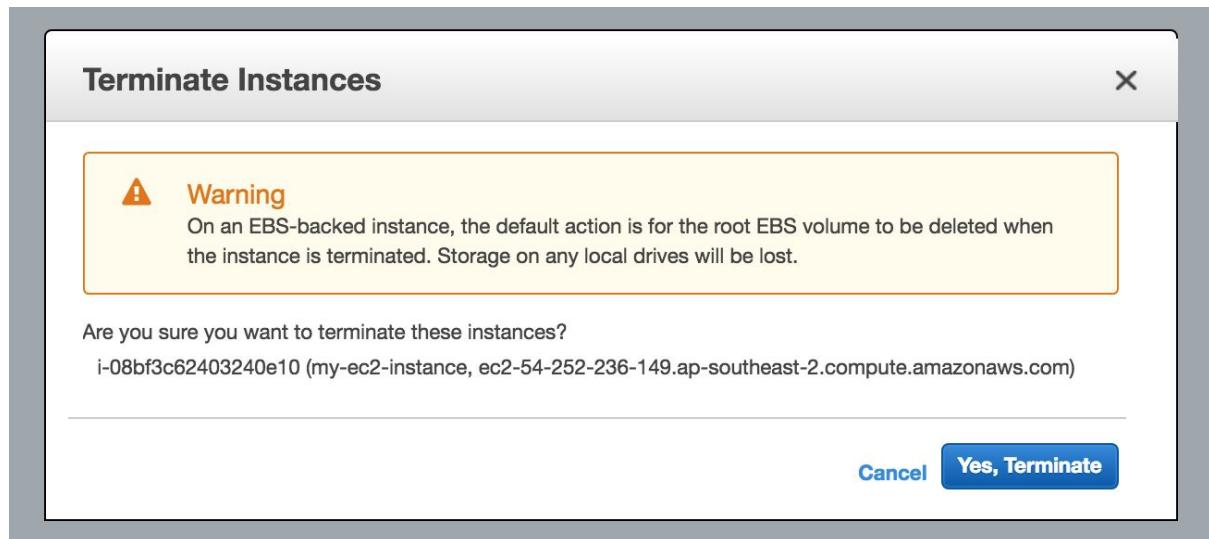
- Have a browse around the available metrics. Over time you will be able to see these metrics graphed here.



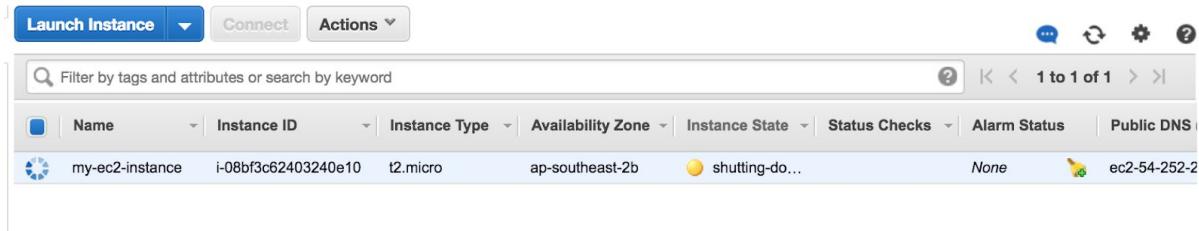
- Now let's clean up. We don't need this server anymore:
- Change your region back to Sydney.
- Back in the EC2 console page, with the instance selected, select "Actions", "Instance State", "Terminate"



- Review the warning to make sure you are terminating the right instance, if so, select "Yes, Terminate"



- Note the Key Pair is still in the account (along with some other stuff), but that's ok we will use it later.



The screenshot shows the AWS EC2 Dashboard. At the top, there are buttons for "Launch Instance", "Connect", and "Actions". Below the header is a search bar with placeholder text "Filter by tags and attributes or search by keyword". The main area displays a table of instances. The columns are: Name, Instance ID, Instance Type, Availability Zone, Instance State, Status Checks, Alarm Status, and Public DNS. One instance is listed: "my-ec2-instance" (Instance ID: i-08bf3c62403240e10, Instance Type: t2.micro, Availability Zone: ap-southeast-2b, Instance State: shutting-down, Status Checks: None, Public DNS: ec2-54-252-2).

- Return to the "EC2 Dashboard", you should have 0 (Zero) running instances.

Resources

You are using the following Amazon EC2 resources in the Asia Pacific (Sydney) region:

0 Running Instances	0 Elastic IPs
0 Dedicated Hosts	0 Snapshots
0 Volumes	0 Load Balancers
1 Key Pairs	2 Security Groups
0 Placement Groups	

Learn more about the latest in AWS Compute from AWS re:Invent 2017 by viewing the [EC2 Videos](#).

Create Instance

END

Session 2 - CloudFormation

In this session we will get introduced to AWS CloudFormation.

Take the following steps, one at a time. There is plenty of time to complete the session and ask any questions along the way.

Steps

- Lets create a CloudFormation template

- Open your favorite text editor (e.g. VSCode) and create a new YAML file.
- File new
- Save as "my-template.yaml"

Build a CloudFormation Template

- Now let's build the template

- First we add the template version:

```
AWSTemplateFormatVersion: 2010-09-09
```

- In this simple template we are only going to use the 'Resources' section:

- In the Resources section we will add an entry for each resource we are going to create. In this case that is one EC2 Instance

- (As we build the YAML file, look to ensure the tab (spaces) are correct)

```
Resources:  
  myInstance:  
    Type: "AWS::EC2::Instance"
```

- The "myInstance" element is used to refer to our instance elsewhere in the template.
That's not of much use in this template, but will be useful in more complex templates later.

- Notice that we have used an AWS specific Type of "AWS::EC2::Instance". This is where we set that we are creating an EC2 Instance

- Lets configure the instance with some properties. Create the following lines, read for some values:

Properties:

ImageId:

InstanceType:

KeyName:

- Let's find the Image Id. When we created the image 'by hand' this was listed during Step 1.

- While we are creating this template, open a browser tab, in the EC2 Dashboard, and select "Launch instance"

- In "Step 1" look at the OS/Images, scroll to "Microsoft Windows Server 2016 Base"

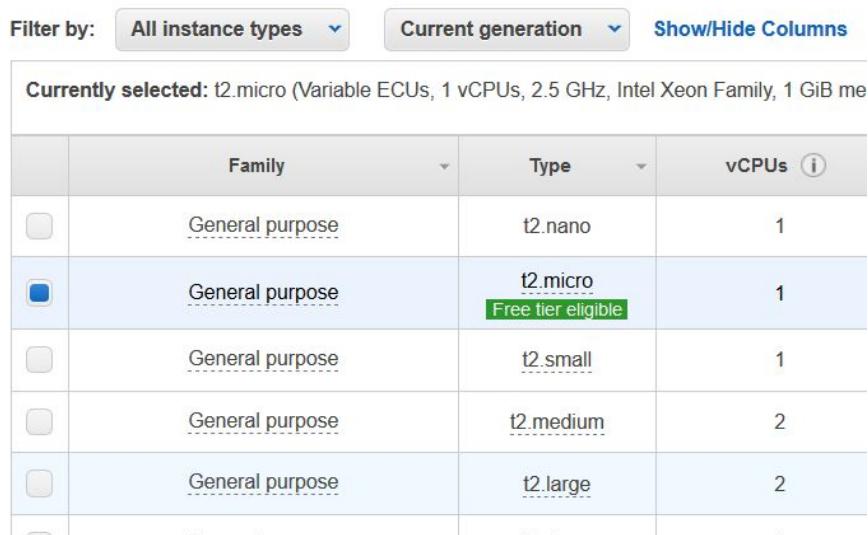


The screenshot shows the AWS EC2 Launch Wizard. At the top, there is a button labeled "Launch a database using RDS". Below it, there are two AMI options listed:

- Microsoft Windows Server 2016 Base - ami-48d6122a**: This is a Windows AMI. It is marked as "Free tier eligible". The description says "Microsoft Windows 2016 Datacenter edition. [English]". It also specifies "Root device type: ebs" and "Virtualization type: hvm".
- Deep Learning AMI (Ubuntu) Version 4.0 - ami-e7c23985**: This is an Ubuntu AMI.

- See the Amazon Machine Image ID (ami-xxxxxxx) written next to the AMI name.
- Make a note of this value.

- Let's find the InstanceType.
- Select the AMI in the wizard to see "Step 2"



The screenshot shows the AWS EC2 Instances page. At the top, there are filters: "Filter by: All instance types" and "Current generation". Below that, a message says "Currently selected: t2.micro (Variable ECUs, 1 vCPUs, 2.5 GHz, Intel Xeon Family, 1 GiB me". The main part is a table of instance types:

	Family	Type	vCPUs
<input type="checkbox"/>	General purpose	t2.nano	1
<input checked="" type="checkbox"/>	General purpose	t2.micro Free tier eligible	1
<input type="checkbox"/>	General purpose	t2.small	1
<input type="checkbox"/>	General purpose	t2.medium	2
<input type="checkbox"/>	General purpose	t2.large	2
	General purpose	On demand	4

- In "Step 2" look at the Type we used. It was "t2.micro"

- Make a note of this value.
- Finally lets find the name of a Key Pair we can use.
- We created a Key Pair in the first session today. Let's find its name.
- Cancel the "Launch Instance" wizard.
- On the left hand side, scroll down to "Key Pairs"



The screenshot shows the AWS Key Pairs management interface. At the top, there are three buttons: 'Create Key Pair' (highlighted in blue), 'Import Key Pair', and 'Delete'. Below the buttons is a search bar with the placeholder 'Filter by attributes or search by keyword'. A table lists key pairs, with columns for 'Key pair name' and 'Fingerprint'. One entry is visible: 'my-key-pair' with the fingerprint '29:b4:d8:fa:49'.

- You should have one Key pair name in here. Make a note of this value. It should be the same name as the PEM file your downloaded in Session 1 (that's another way to find the name)
- Back in the CloudFormation template, enter the values we found into the relevant lines:

Properties:

```
ImageId: ami-82a458e0
InstanceType: t2.micro
KeyName: my-key-pair
```

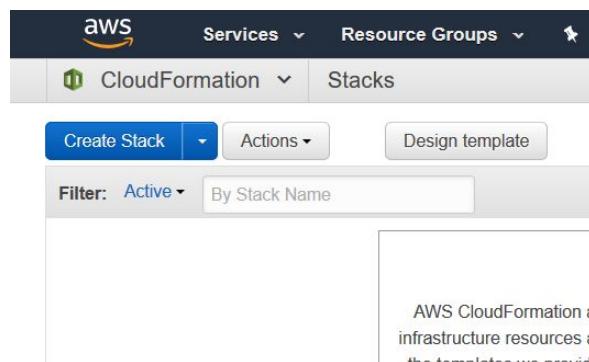
- The complete file should now look like something like this:

```
AWSTemplateFormatVersion: 2010-09-09
Resources:
myInstance:
  Type: "AWS::EC2::Instance"
  Properties:
    ImageId: ami-82a458e0
    InstanceType: t2.micro
    KeyName: my-key-pair
```

- Save your YAML CloudFormation template somewhere on your computer.

Build a CloudFormation Stack

- Let's use the template to create a CloudFormation Stack
- Log into the AWS console
- Set the region to Sydney (stay in this region today, check from time to time)
- Navigate to CloudFormation



- Select "Create Stack"
- From the next screen, select "Upload a template to Amazon S3" and click "Browse" and select the CloudFormation template you created earlier in this session.

Choose a template A template is a JSON/YAML-formatted text file that describes your stack's resources and their properties. [Learn more](#).

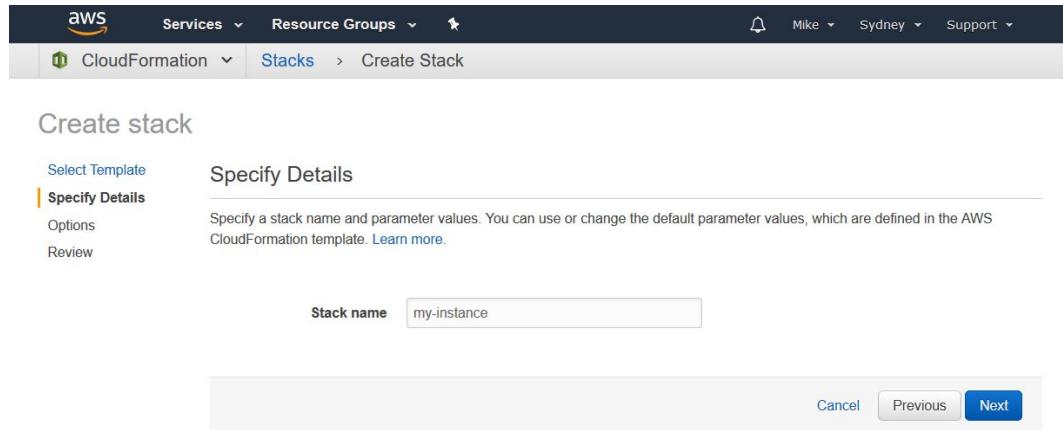
Select a sample template
 Upload a template to Amazon S3
 Specify an Amazon S3 template URL

Browse... No file selected.

- Select "Next"

- If you are shown a validation error at this point, check that your YAML file looks like the example above. Are the spaces correct?

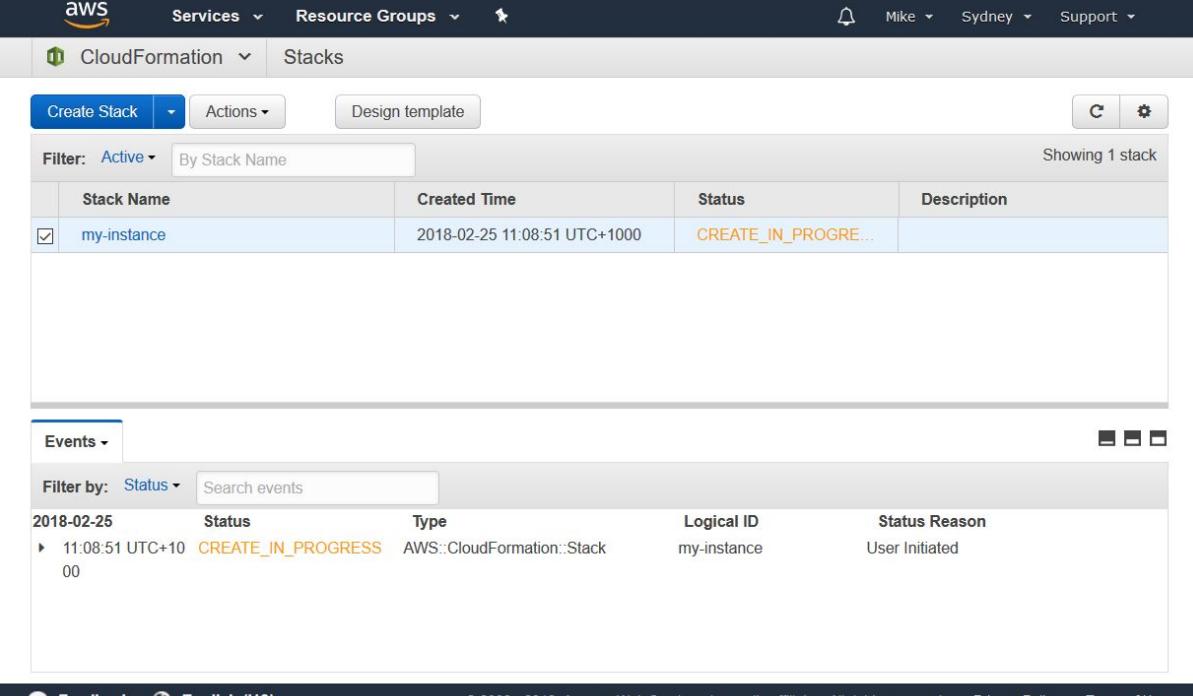
- Enter a name for your CloudFormation Stack. Enter "my-instance".



- Select "Next"

- The next screen provides some more advanced options for your stack. For now, scroll to the bottom of this page and select "Next"

- The next page shows a 'Review' or summary page. Scroll to the bottom of this page and select "Create"



The screenshot shows the AWS CloudFormation Stacks page. At the top, there are buttons for 'Create Stack', 'Actions', and 'Design template'. A filter bar shows 'Active' and 'By Stack Name'. Below the table, an 'Events' section displays a log entry for the creation of the 'my-instance' stack.

Stack Name	Created Time	Status	Description
my-instance	2018-02-25 11:08:51 UTC+1000	CREATE_IN_PROGRESS...	

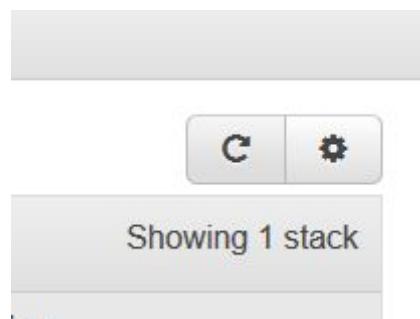
2018-02-25	Status	Type	Logical ID	Status Reason
11:08:51 UTC+10	CREATE_IN_PROGRESS	AWS::CloudFormation::Stack	my-instance	User Initiated 00

Feedback English (US) © 2008 - 2018, Amazon Web Services, Inc. or its affiliates. All rights reserved Privacy Policy Terms of Use

- You should now be taken to the CloudFormation stacks page. You can watch your stack being created.

- With your stack selected:

- Click the refresh button (within the page, NOT the main browser refresh button) from time to time, this will update all the information about the stack build.



- Review the "Events" tab, this shows you the steps being taken by the Cloudformation build engine.

Overview	Outputs	Resources	Events	Template	Parameters	Tags	Stack Policy	Change Sets	Rollback Triggers						
Filter by: Status ▾ Search events															
2018-02-25	Status	Type	Logical ID	Status Reason											
▶ 11:09:30 UTC+1000	CREATE_COMPLETE	AWS::CloudFormation::Stack	my-instance												
▶ 11:09:28 UTC+1000	CREATE_COMPLETE	AWS::EC2::Instance	myInstance												
▶ 11:08:55 UTC+1000	CREATE_IN_PROGRESS	AWS::EC2::Instance	myInstance	Resource creation Initiated											
▶ 11:08:54 UTC+1000	CREATE_IN_PROGRESS	AWS::EC2::Instance	myInstance												
▶ 11:08:51 UTC+1000	CREATE_IN_PROGRESS	AWS::CloudFormation::Stack	my-instance	User Initiated											

- If there are any errors during the build, they will be shown here.
- Review the "Resources" tab, this shows the resources built in this stack. In this case you should see just one EC2 Instance.

Overview	Outputs	Resources	Events	Template	Parameters	Tags	Stack Policy	Change Sets	Rollback Triggers			
Logical ID	Physical ID	Type	Status	Status	Status	Status						
myInstance	i-0ecd6f05e7e49f4d9	AWS::EC2::Instance	CREATE_COMPLETE	CREATE_COMPLETE	CREATE_COMPLETE	CREATE_COMPLETE						

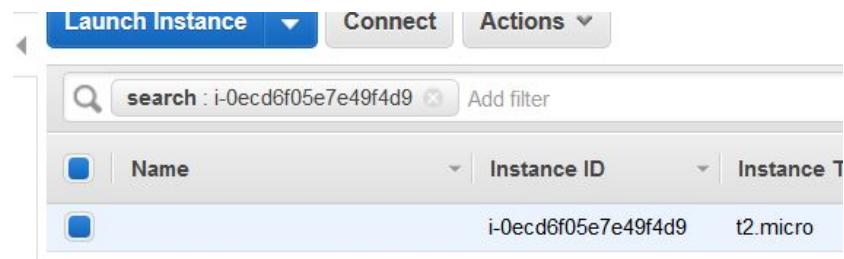
- When the stack has built, with no errors, you should see the status as "CREATE_COMPLETE"

- Not seeing "CREATE_COMPLETE"? If the build has failed, see if you can see why from the "Events" tab. If you need help please ask :)
- With your stack selected, from the "Resources" tab click the "Physical ID" of the instance that the strack created.

Outputs	Resources	Events	Template
	Physical ID		
▶	i-0ecd6f05e7e49f4d9		

- This will take you to the EC2 Instance list. Note the list of instances is now filtered to just the instance you created (which is your only instance)

- See that the Instance does not have a Name tag. Let's fix that... not in the console, but in the CloudFormation template.



The screenshot shows the AWS EC2 Instances list. At the top, there are buttons for 'Launch Instance', 'Connect', and 'Actions'. Below that is a search bar with the placeholder 'search : i-0ecd6f05e7e49f4d9' and a 'Add filter' button. The main table has columns for 'Name', 'Instance ID', and 'Instance Type'. A single row is visible, showing 'i-0ecd6f05e7e49f4d9' under 'Instance ID' and 't2.micro' under 'Instance Type'. There is also a small blue square icon next to the 'Name' column header.

Name	Instance ID	Instance Type
	i-0ecd6f05e7e49f4d9	t2.micro

Update a CloudFormation Stack

- Back in your text editor, with your CloudFormation YAML file loaded, add a Name tag to your instance. Under KeyName, and on the same level as KeyName, add:

Tags :

-

```
Key: Name
Value: my-ec2-instance
```

- Again make sure the spacing is correct, your template should now look something like this:

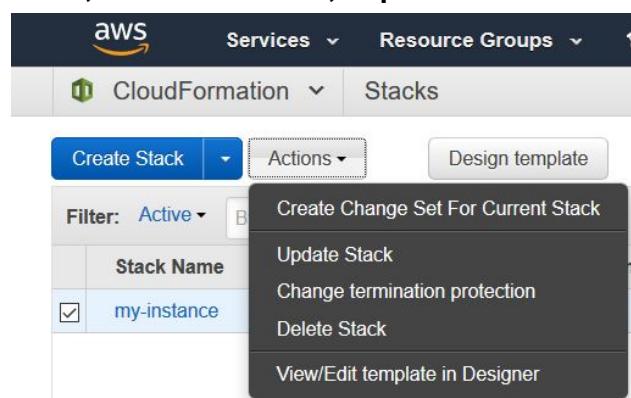
```
AWSTemplateFormatVersion: 2010-09-09
Resources:
myInstance:
  Type: "AWS::EC2::Instance"
  Properties:
    ImageId: ami-82a458e0
    InstanceType: t2.micro
    KeyName: mykey
  Tags:
  -
    Key: Name
    Value: my-ec2-instance
```

- Save your updated template.

- Now, let's update the stack so the changes take effect.

- Navigate to CloudFormation

- With your stack selected, select "Actions", "Update Stack"



- From the next screen, select "Upload a template to Amazon S3" and click "Browse" and select your updated CloudFormation template.

Choose a template A template is a JSON/YAML-formatted text file that describes your stack's resources.

Use current template

Upload a template to Amazon S3

test.yaml

Specify an Amazon S3 template URL

- Select "Next"

- If you are shown a validation error at this point, check that your YAML file looks like the example above. Are the spaces correct?

- There should be nothing to change on the next screen, select "Next"

Update my-instance stack

Select Template **Specify Details**

Specify parameter values. You can use or change the default parameter values, which are defined in the AWS CloudFormation template. [Learn more](#).

Options Review

Stack name

- The next screen provides some more advanced options for your stack. For now, scroll to the bottom of this page and select "Next"

- The next page shows a 'Review' or summary page.

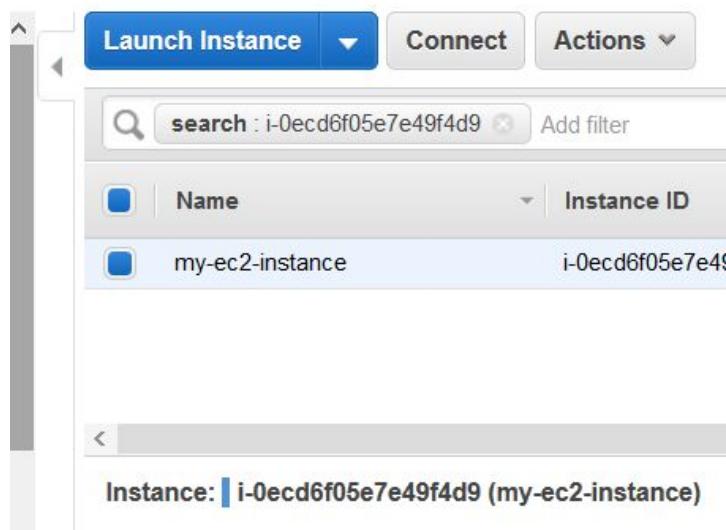
- Notice the "Preview your changes" section, this gives you a very high level summary of what will change in the stack.

Preview your changes

Based on your input, CloudFormation will change the following resources. For more information, choose [View change set details](#).

Action	Logical ID	Physical ID	Resource type	Replacement
Modify	myInstance	i-0ecd6f05e7e49f4d9	AWS::EC2::Instance	False

- Notice that while the EC2 Instance will be updated, it will not get replaced, so the instance will remain running.
- Select "Update"
- As before watch CloudFormation do its work, refresh, and review the "Events" tab.
- When the stack status shows as "UPDATE_COMPLETE", follow the link from the "Resources" tab to the instance view.
- Notice that the instance is now Named.

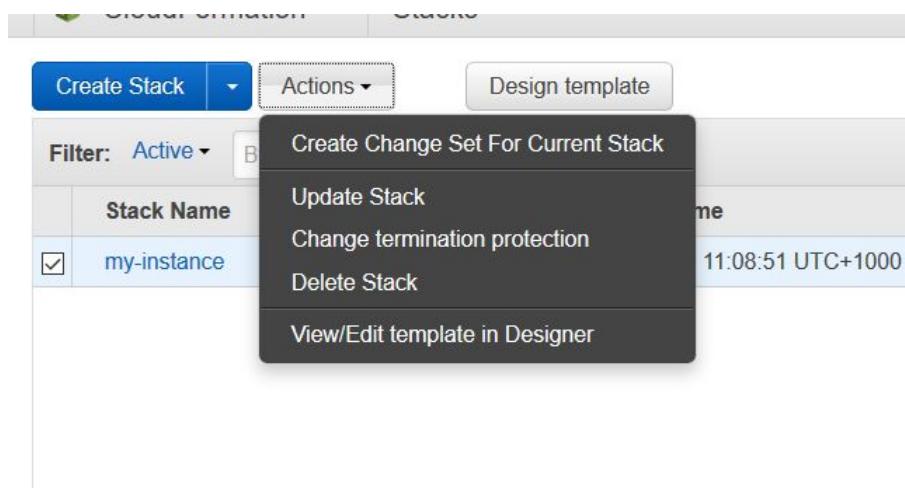


- If you have time you can connect to the server via RDP if you wish. When you're finished...
- We don't need this server anymore, let's clean up.

Delete a CloudFormation Stack

- Cleaning up the CloudFormation stack is super easy, and super clean. The instance and all related configurations (but not the Key Pair as this wasn't created in the stack) are gone in one action:

- Navigate to CloudFormation
- With your stack selected, select "Actions", "Delete Stack"



- Review the warning and, if you are ready select "Yes, Delete"
- Monitor as the stack is deleted.
- Want to build the server again, name tag and all? Just build a new stack with your saved template!

Extra Challenge

While we wait for everyone to complete this session....

- Can you make a change to the CloudFormation template to create 2 servers?
- Are the servers created one after each other, or do they build at the same time?

END

Session 3 - Virtual Private Cloud

In this session we will get introduced to AWS VPC.

Take the following steps, one at a time. There is plenty of time to complete the session and ask any questions along the way.

Steps

- Log into the AWS console
- Set the region to Sydney (stay in this region today, check from time to time)
- The nice thing about CloudFormation templates is that they can be easily shared around. In the resource file for this session you should find 'vpc-network.yaml'
- Take a look at the file, we've added comments along the way to explain how it works. Notice that this template uses 'Parameters' to pass values in to the stack as it builds.

This template creates a VPC with four subnets. Two of these subnets are 'Public' in that they have access to and from the Internet. Two of the subnets are 'Private' in that they have no access to the Internet.

- Navigate to CloudFormation
- Select "Create Stack"
- From the next screen, select "Upload a template to Amazon S3" and click "Browse" and select the CloudFormation template 'vpc-network.yaml'.

Choose a template A template is a JSON/YAML-formatted text file that describes your stack's resources and their properties. [Learn more](#).

Select a sample template

Upload a template to Amazon S3 No file selected.

Specify an Amazon S3 template URL

- Select "Next"
- This time as well as a stack name, you have the opportunity to alter some of the values that the stack will use.

Specify Details

Specify a stack name and parameter values. You can use or change the default parameter values, which are defined in the AWS CloudFormation template. [Learn more.](#)

Stack name 

Parameters

<code>subnetPrivateCidr1</code>	<input type="text" value="10.0.4.0/24"/>	Enter a valid CIDR for the first private subnet
<code>subnetPrivateCidr2</code>	<input type="text" value="10.0.5.0/24"/>	Enter a valid CIDR for the second public subnet
<code>subnetPublicCidr1</code>	<input type="text" value="10.0.0.0/24"/>	Enter a valid CIDR for the first public subnet
<code>subnetPublicCidr2</code>	<input type="text" value="10.0.1.0/24"/>	Enter a valid CIDR for the second public subnet
<code>vpcCidr</code>	<input type="text" value="10.0.0.0/16"/>	Enter a valid CIDR for the VPC

[Cancel](#) [Previous](#) [Next](#)

- Enter a 'Stack name' and leave the other Parameters as default. Select 'Next'.

- Scroll to the bottom of the next page and select 'Next' and then select 'Create'.

Stacks			
Actions	Design template	C	G
Filter: Active ▾ <input type="text" value="By Stack Name"/>	Showing 1 stack		
Stack Name	Created Time	Status	Description
<input checked="" type="checkbox"/> my-network	2018-03-21 14:37:34 UTC+1000	CREATE_IN_PROGRESS...	This template creates an ...

- As before watch CloudFormation do its work, refresh, and review the "Events" tab.

- Navigate to VPC

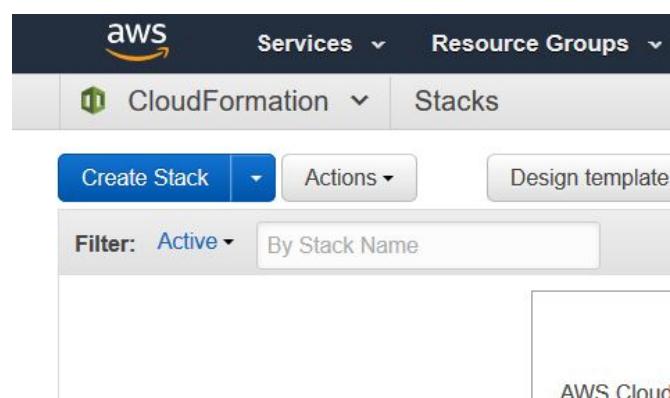
- Take a look at the VPC you created.

<input type="text"/> Search VPCs and their properties X				
	Name	VPC ID	State	IPv4 CIDR
<input type="checkbox"/>	demo-vpc	vpc-c1f244a6	available	10.0.0.0/16
<input type="checkbox"/>		vpc-17754273	available	172.31.0.0/16

- Notice that this is not the only VPC in your account. You should also see an (unnamed) 'default' VPC.
- We have now created a VPC with four subnets, two are public (in that it can route to and from the Internet) and two subnets are private.

Testing the Virtual Private Cloud (VPC)

- Now let's create a couple of EC2 Instances to test with.
- First we will create an EC2 instance in a public subnet.
- Navigate to CloudFormation
- Select "Create Stack"



- Select "Upload a template to Amazon S3" and "Browse"...
- In the resources provided for this workshop, in the folder "Session 3" select the template: "ec2-instance-for-subnet.yaml"

Choose a template A template is a JSON/YAML-formatted text file that describes your stack's resources.

Select a sample template

Upload a template to Amazon S3

Browse... ec2-instance-for-subnet.yaml

Specify an Amazon S3 template URL

- Select "Next"

- This template has been created to prompt you for parameters before the stack is built.

- Enter a name for the stack "my-public-instance-stack" (this is not the name that will be given to the instance)
- The Key parameter is now a dropdown box. Select a key that you have on your machine.
- Enter a server name "my-demo-server-public"
- In the subnet dropdown menu you will have a few options. Select the subnet that has "(subnet-public-1)"
- Before moving on, make a note of the subnet ID and its corresponding name. We will need this information later. (e.g. subnet-public-1 : subnet-beb4bd9de9893478983)

Specify Details

Specify a stack name and parameter values. You can use or change the default parameter values, which are defined in the AWS CloudFormation template. [Learn more](#).

Stack name

Parameters

key Enter a valid EC2 Key

serverName Enter the name for the server. (This is not a hostname)

subnet Select a subnet for this instance.

[Cancel](#) [Previous](#) [Next](#)

- Select "Next"
- Leave all the settings on the next screen.
- Scroll to the bottom and select "Next"
- Scroll to the bottom of the Review page and select "Create"
- We won't wait for this stack to create, let's create another EC2 Instance in the private subnet.
- Select "Create Stack" and follow the same process, but this time:

- Enter a name for the stack "my-private-instance-stack"
- Enter a server name "my-demo-server-private"
- Select the subnet with "(subnet-private-1)"

Specify Details

Specify a stack name and parameter values. You can use or change the default parameter values, which are defined in the AWS CloudFormation template. [Learn more](#).

Stack name

Parameters

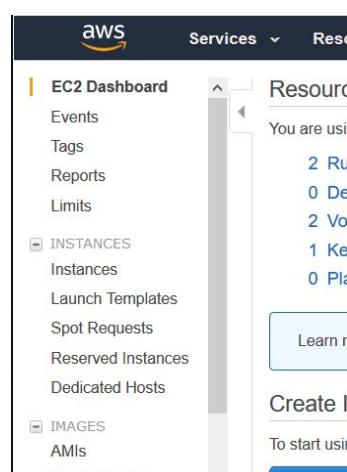
key Enter a valid EC2 Key

serverName Enter the name for the server. (This is not a hostname)

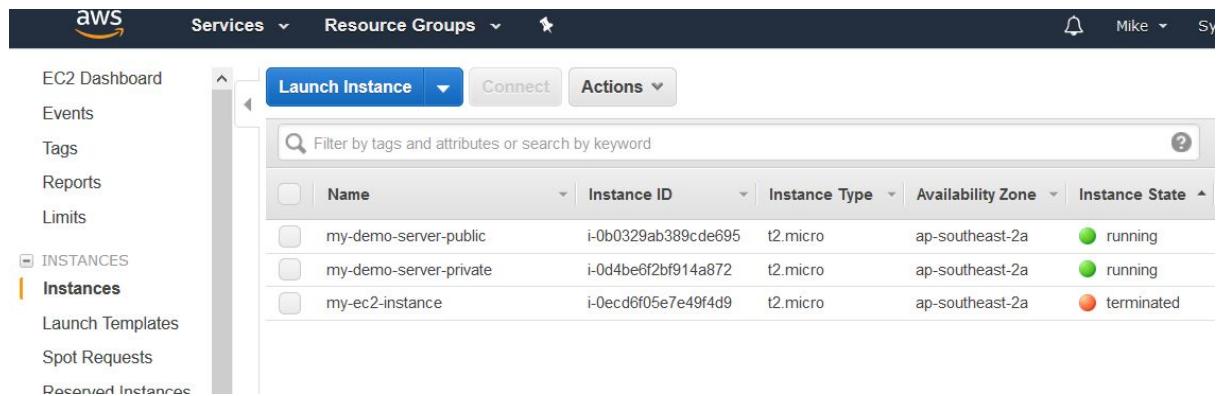
subnet Select a subnet for this instance.

[Cancel](#) [Previous](#) [Next](#)

- Select “Next”, “Next”, then “Create”
- By now, our public EC2 Instance may have built. Let's connect into the server.
- Navigate to EC2
- Select "Instances" on the left hand side.



- By now, you should see two running EC2 instances, if you need to expand the Name column.



The screenshot shows the AWS EC2 Dashboard. On the left sidebar, under the 'INSTANCES' section, 'Instances' is selected. The main area displays a table of running instances:

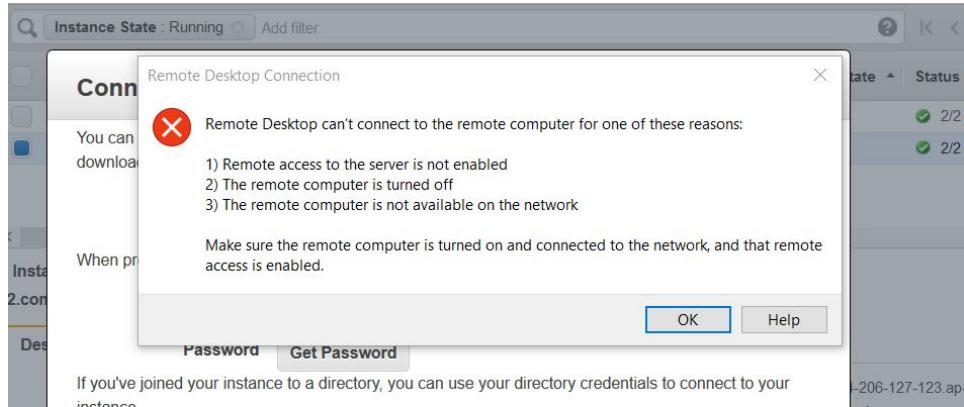
Name	Instance ID	Instance Type	Availability Zone	Instance State
my-demo-server-public	i-0b0329ab389cde695	t2.micro	ap-southeast-2a	running
my-demo-server-private	i-0d4be6f2bf914a872	t2.micro	ap-southeast-2a	running
my-ec2-instance	i-0ecd6f05e7e49f4d9	t2.micro	ap-southeast-2a	terminated

- Connect with RDP into the public server:
- Select the public instance, "Actions" and "Get Windows Password"
- Password not available yet? Hold on a minute, and try again.



- Try to connect to the public instance via your RDP client...

- Its not working!!



- In our VPC we have Security Groups that control the traffic that can reach our Instances.

- Both of our instances are in the 'default' security group.

- Let's create a rule in the default security group to allow access to our Instances via RDP.

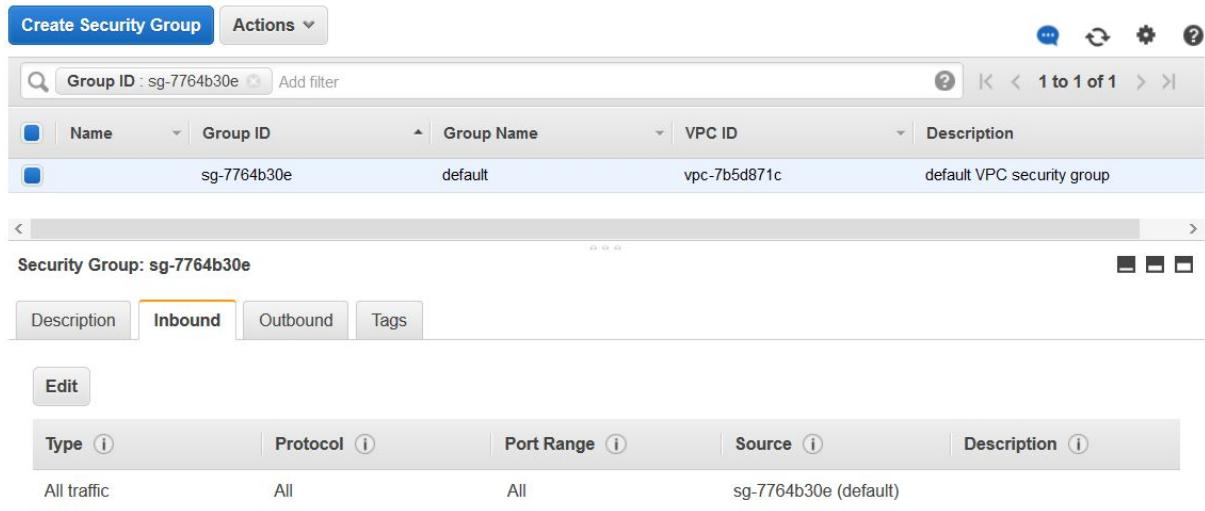
- From the EC2 list select either of the instances, look at the details section in the lower half of the screen.

- In the "Description" tab, find the "Security groups" and click on "default"

Instance ID	i-0c3729f391f788b4e
Instance state	running
Instance type	t2.micro
Elastic IPs	
Availability zone	ap-southeast-2a
Security groups	default . view inbound rules
Scheduled events	No scheduled events
AMI ID	Windows_Server-2016-English-Full-Base-2018.01.12 (ami-82a458e0)
Platform	windows
IAM role	-
Key pair name	my-key-pair

- On the following screen in the lower half, you will see the details of the security group.

- Select the "Inbound" tab.



Name	Group ID	Group Name	VPC ID	Description
sg-7764b30e	default	vpc-7b5d871c	default VPC security group	

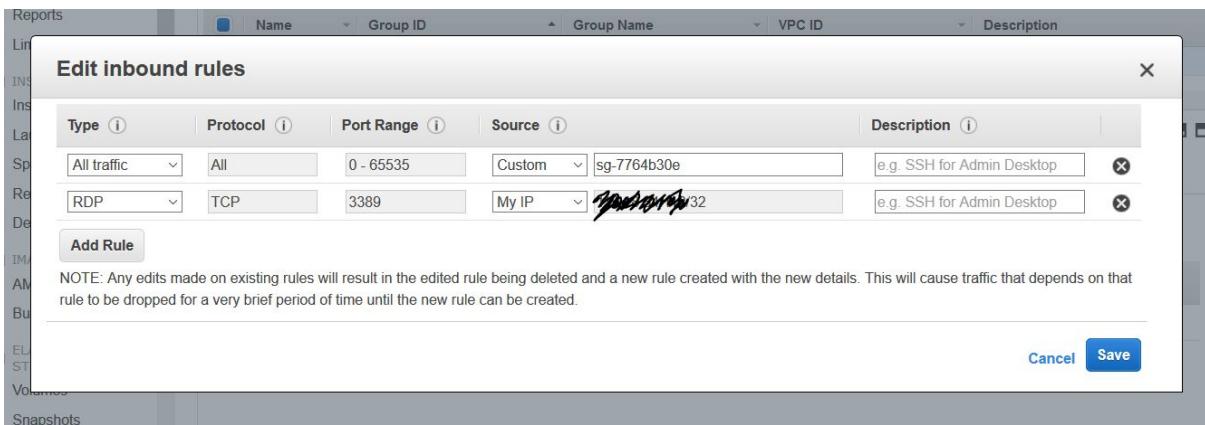
Inbound

Type	Protocol	Port Range	Source	Description
All traffic	All	All	sg-7764b30e (default)	e.g. SSH for Admin Desktop

- Notice that currently, "All traffic" is allowed inbound to our instances, but the source is this (the same) security group.

- This means that the instances can RDP to each other, but we can't RDP from outside the network.

- Select "Edit", then "Add Rule"



Type	Protocol	Port Range	Source	Description
All traffic	All	0 - 65535	Custom sg-7764b30e	e.g. SSH for Admin Desktop
RDP	TCP	3389	My IP	e.g. SSH for Admin Desktop

Add Rule

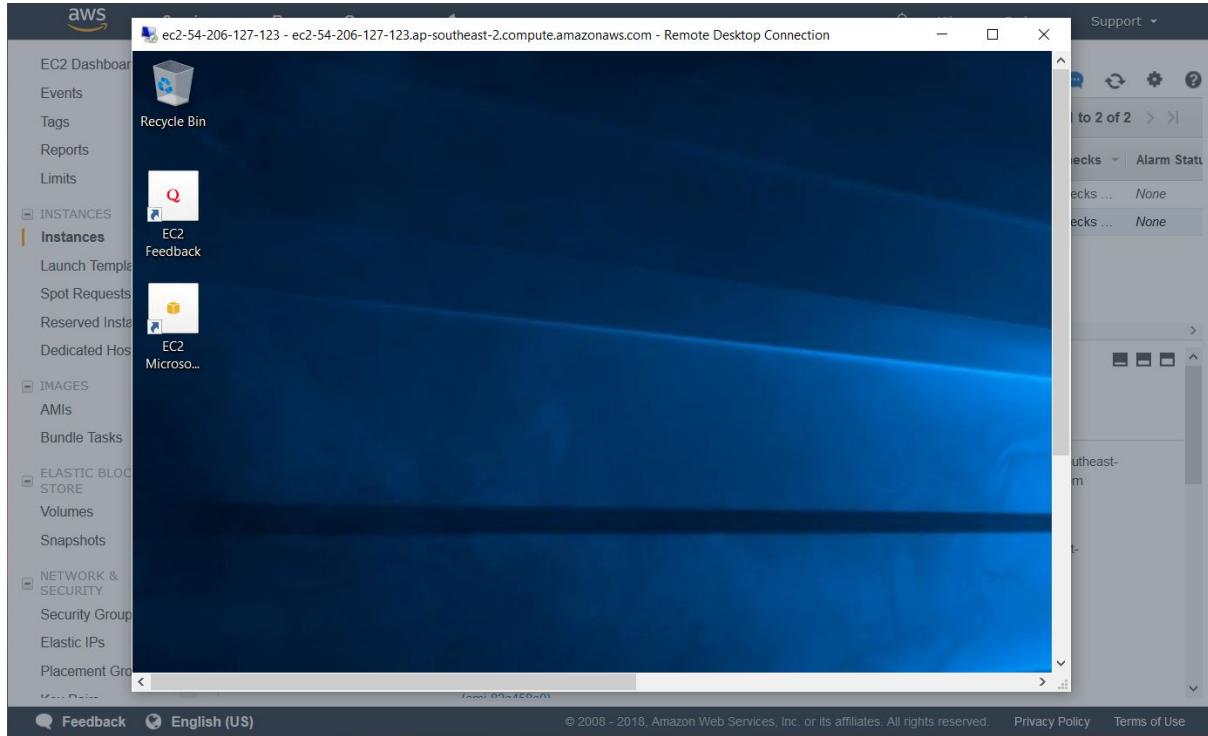
NOTE: Any edits made on existing rules will result in the edited rule being deleted and a new rule created with the new details. This will cause traffic that depends on that rule to be dropped for a very brief period of time until the new rule can be created.

Cancel **Save**

- On the new rule, for Type select "RDP", and for Source select "My IP"

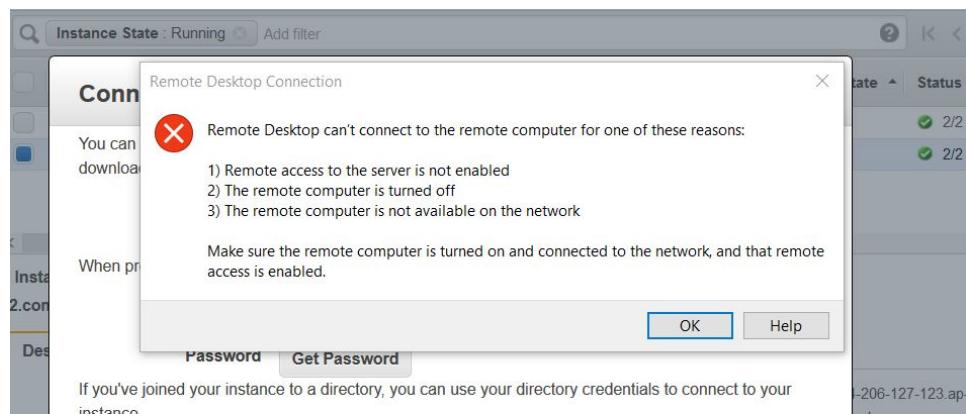
- By selecting "My IP" we restrict RDP access from the outside world to only machines using your Internet connection.

- Now, try again to RDP to your public instance.



- Yay, it works.

- OK, minimise the RDP session to the public instance. Let's see if we can RDP to the private instance:



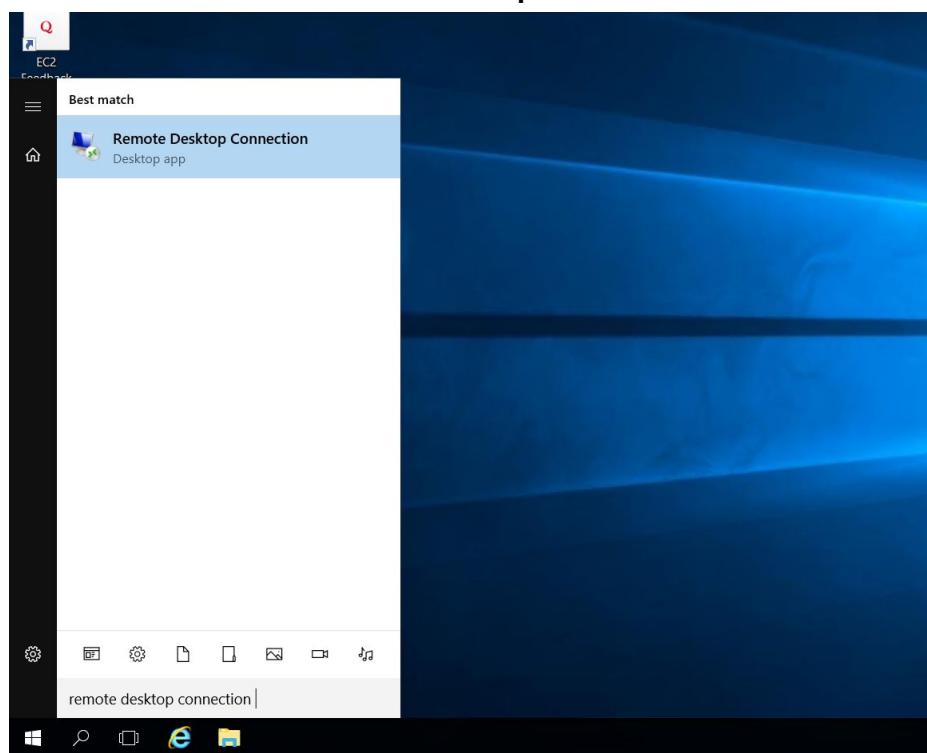
- Of course, it doesn't work. That's because, even though the instance has been allocated a public IP address, its not 'visible' to the Internet through the VPC.

- But we can still get there:

- From the EC2 Instance list, select the private server, and in the details in the bottom half of the screen find "Private IPs".

Public DNS (IPv4)	ec2-13-211-29-84.ap-southeast-2.compute.amazonaws.com
IPv4 Public IP	13.211.29.84
IPv6 IPs	-
Private DNS	ip-10-0-1-7.ap-southeast-2.compute.internal
Private IPs	10.0.1.7
Secondary private IPs	
VPC ID	vpc-7b5d871c
Subnet ID	subnet-942949f3

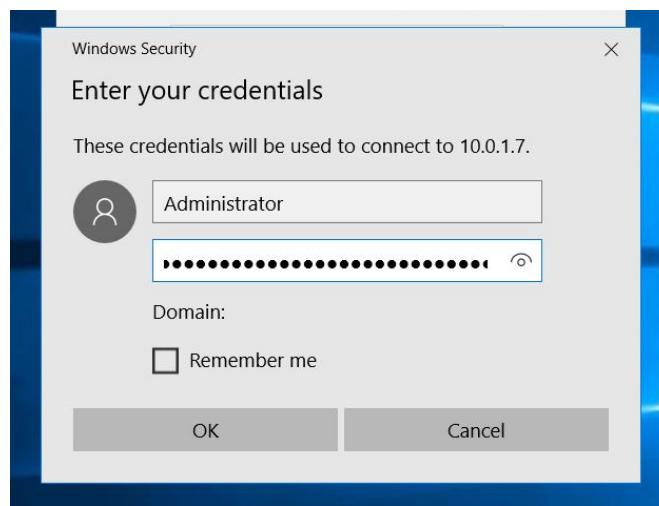
- Make a note of this address.
- Also take a moment to discover the password for this server.
- Go back to the RDP session with your 'public' instance.
- From the 'Start' menu select "Remote Desktop Connection"



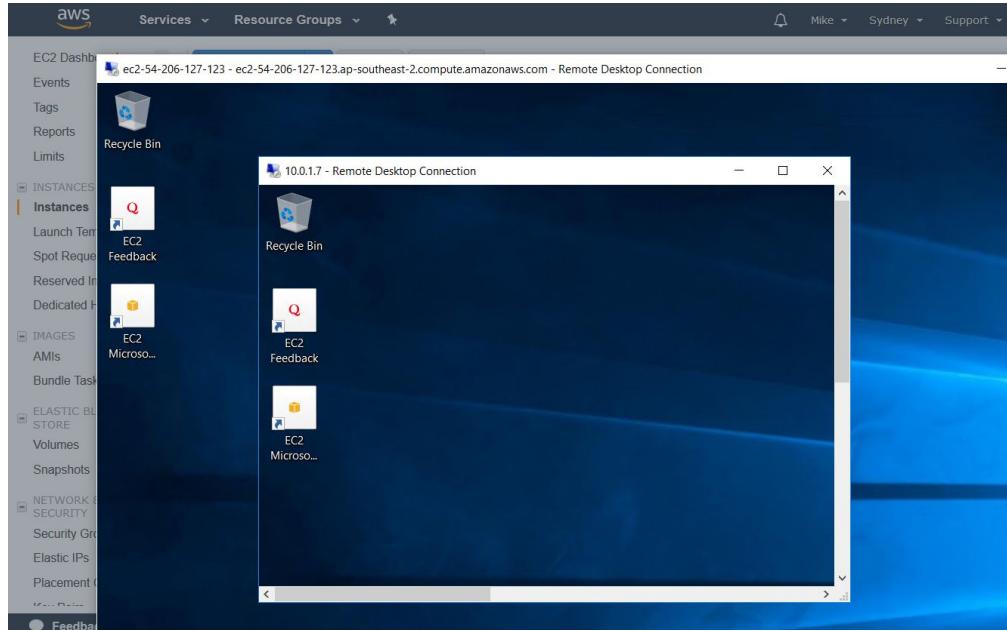
- Enter in to the "Computer" field the Private IP address of the private server that you noted down just before.



- Select "Connect"



- In the credentials box, enter the User name "Administrator" and the password for the private server.



- Yay, it works.

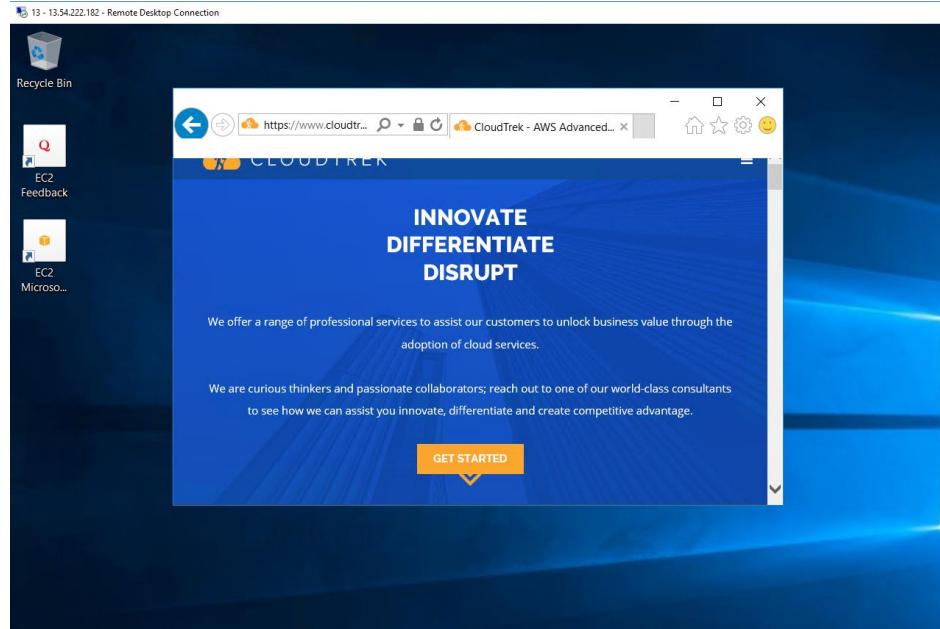
- You have created a private server, in a private network, that is accessible via what we call a bastion host.

Before we finish...

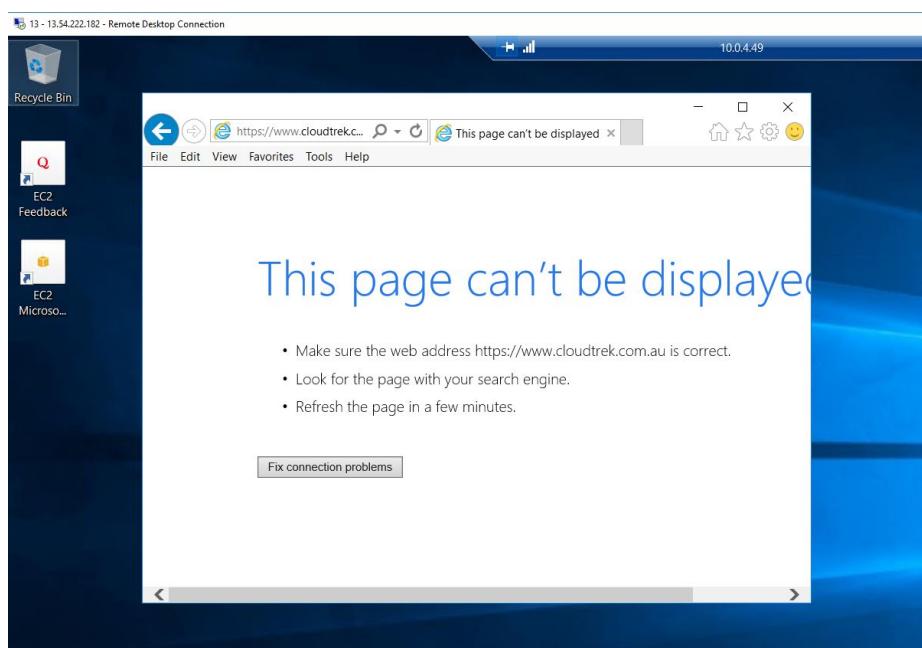
- Let's look at Internet FROM connectivity for our EC2 Instances. It's often the case that servers need to connect to the Internet to download patches etc.

- From the RDP session with the public server, open up Internet Explorer and navigate to <http://www.cloudtrek.com.au> (or any website).

- As this is a default 'super paranoid' server configuration, there will be a slew of warning messages, but you should end up looking at a webpage.



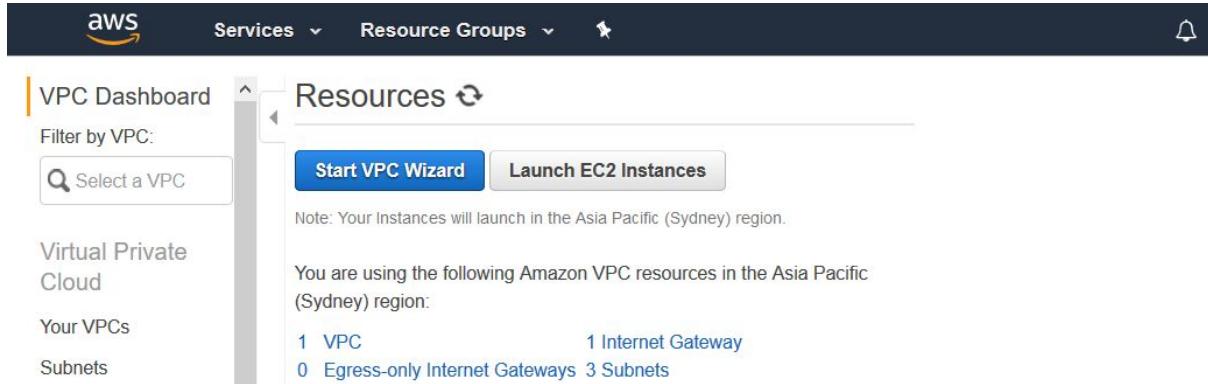
- If you do the same from the Private server... It does not work!



- Let's fix this. Let's allow our private server to have access to reach out to the Internet, keeping it in its private subnet so that nobody can connect to it from the outside world directly.

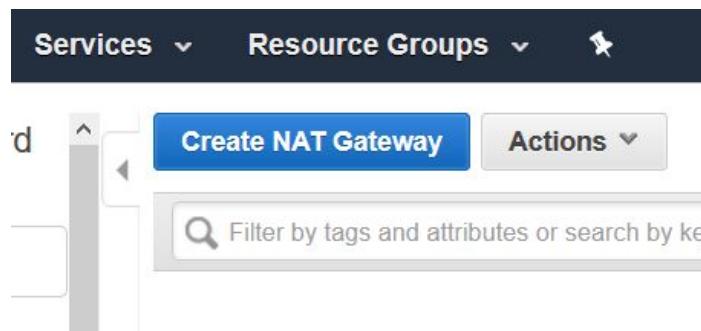
- Leave your RDP sessions open....

- Navigate to VPC in the AWS console.



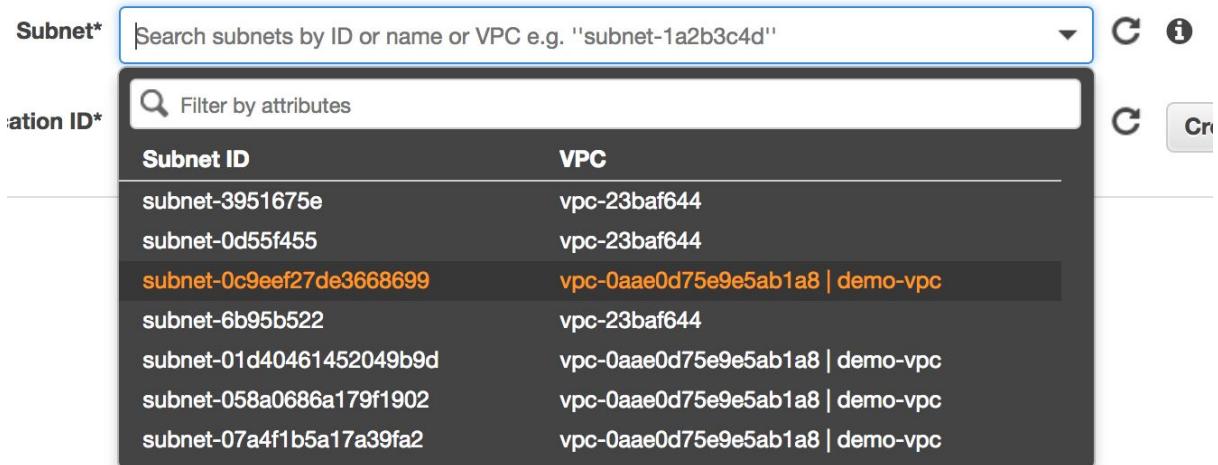
The screenshot shows the AWS VPC Dashboard. On the left, there's a sidebar with 'VPC Dashboard' selected, followed by 'Virtual Private Cloud', 'Your VPCs', and 'Subnets'. The main area is titled 'Resources' with a 'Start VPC Wizard' and 'Launch EC2 Instances' button. It notes that instances will launch in the Asia Pacific (Sydney) region. Below this, it says you are using the following Amazon VPC resources in the Asia Pacific (Sydney) region: 1 VPC, 1 Internet Gateway, 0 Egress-only Internet Gateways, and 3 Subnets.

- From the menu on the left select 'NAT Gateways' then 'Create NAT Gateway'.



The screenshot shows the AWS Services menu with 'Create NAT Gateway' highlighted in blue. Below it is a search bar with the placeholder 'Filter by tags and attributes or search by key'.

- For 'Subnet' choose the subnet ID that correspond to 'subnet-public-1' in demo-vpc. You should have this information noted down from earlier. If not, you will have to go to VPC -> Subnet and copy down the subnet-public-1 subnet ID.



The screenshot shows the 'Create NAT Gateway' wizard. In the 'Subnet*' field, there is a dropdown menu with a search bar containing 'Search subnets by ID or name or VPC e.g. "subnet-1a2b3c4d"'. Below the search bar is a 'Filter by attributes' section. A list of subnets is displayed in a table:

Subnet ID	VPC
subnet-3951675e	vpc-23baf644
subnet-0d55f455	vpc-23baf644
subnet-0c9eef27de3668699	vpc-0aae0d75e9e5ab1a8 demo-vpc
subnet-6b95b522	vpc-23baf644
subnet-01d40461452049b9d	vpc-0aae0d75e9e5ab1a8 demo-vpc
subnet-058a0686a179f1902	vpc-0aae0d75e9e5ab1a8 demo-vpc
subnet-07a4f1b5a17a39fa2	vpc-0aae0d75e9e5ab1a8 demo-vpc

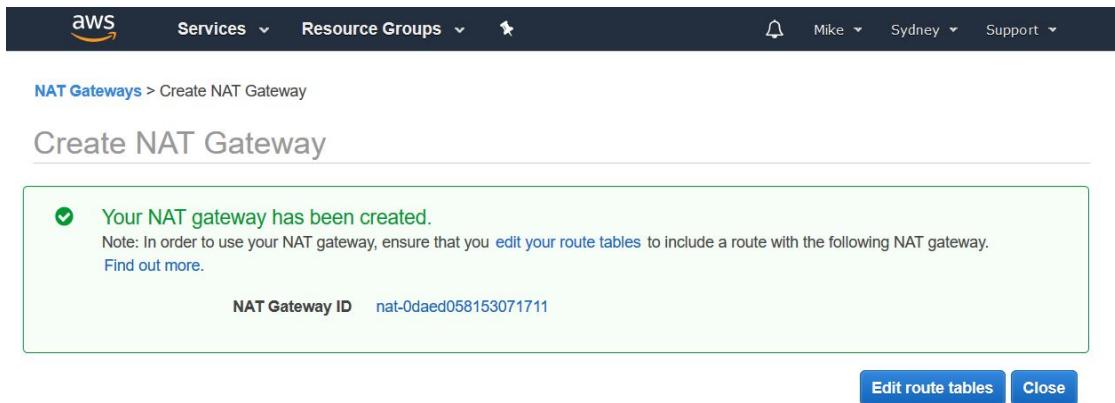
- For 'Elastic IP Allocation ID' click on Create New EIP to fill in the box.

Create NAT Gateway

Create a NAT gateway and assign it an Elastic IP address. [Learn more.](#)

Subnet*	subnet-f1512796		
Elastic IP Allocation ID*	eipalloc-351b180f		
<small>* Required</small>		Cancel	Create a NAT Gateway

- Select 'Create a NAT Gateway'



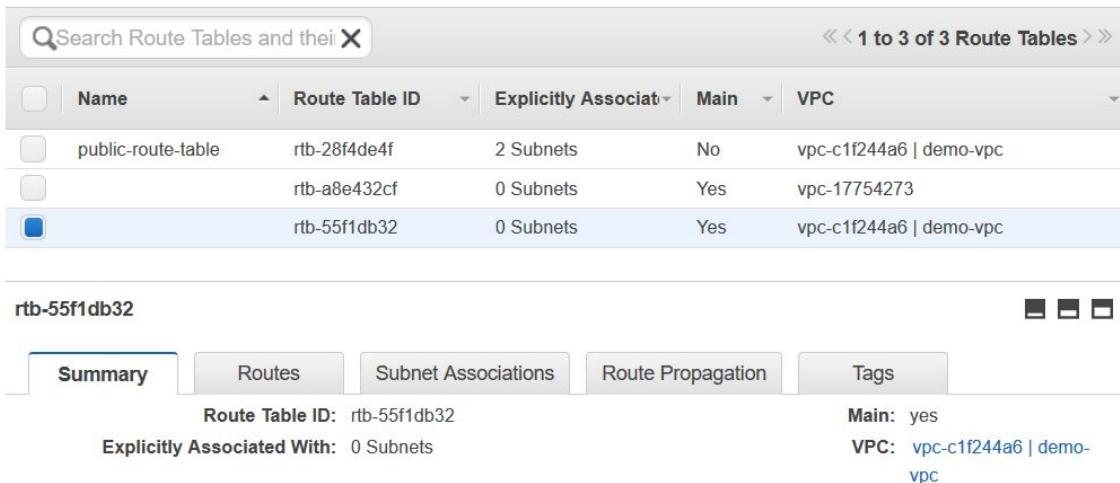
Your NAT gateway has been created.
Note: In order to use your NAT gateway, ensure that you [edit your route tables](#) to include a route with the following NAT gateway.
[Find out more.](#)

NAT Gateway ID nat-0daed058153071711

[Edit route tables](#) [Close](#)

- Select 'Edit route tables'

- Look for the entry in this table for a 'Main' table for the 'demo-vpc'



Search Route Tables and their						1 to 3 of 3 Route Tables
	Name	Route Table ID	Explicitly Associated With	Main	VPC	
	public-route-table	rtb-28f4de4f	2 Subnets	No	vpc-c1f244a6 demo-vpc	
		rtb-a8e432cf	0 Subnets	Yes	vpc-17754273	
<input checked="" type="checkbox"/>		rtb-55f1db32	0 Subnets	Yes	vpc-c1f244a6 demo-vpc	

rtb-55f1db32				
Summary	Routes	Subnet Associations	Route Propagation	Tags
Route Table ID: rtb-55f1db32			Main: yes	
Explicitly Associated With: 0 Subnets			VPC: vpc-c1f244a6 demo-vpc	

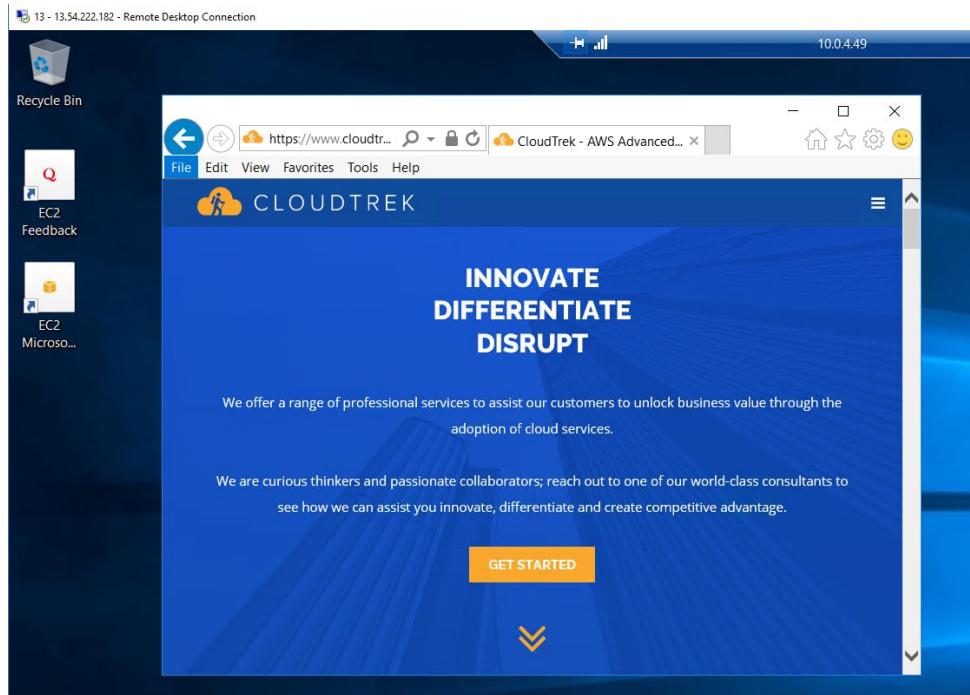
- Select the 'Routes' tab, then 'Edit', then 'Add another route'

Destination	Target	Status	Propagated	Remove
10.0.0.0/16	local	Active	No	
0.0.0.0/0		No		
Add another route	igw-32b8d156 demo-igw nat-0daed058153071711			

- In 'Destination' enter '0.0.0.0/0' and under 'Target' start to type 'nat..' and select the ID of the NAT gateway.

- Select 'Save'

- Go back to the RDP session on your Private server (inside the session to your public server. Try the web page again... it works!

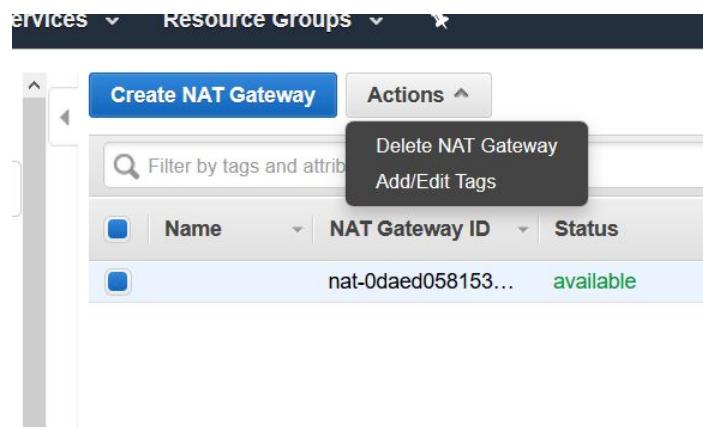


- Note, you still can't RDP directly to the Private server.

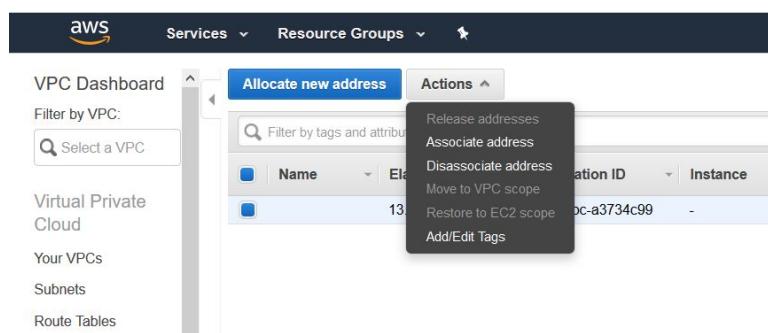
- Once your done experimenting with this environment its time to clean up.

Deleting your Resources

- Navigate to VPC
- Select 'Nat Gateways' from the left hand menu. If the gateway list is empty, you may have to click the 'refresh' button.
- With the Gateway selected, use the action menu and select 'Delete NAT Gateway' and then confirm again 'Delete NAT Gateway'.



- Navigate back to CloudFormation
- Delete the Server stacks by selecting the stack, and selecting "Actions", "Delete Stack", then "Yes, Delete"
- Wait for these stacks to fully delete before doing the same to the network stack.
- One last thing, select "Elastic IPs" from the menu on the left.
- Select the Elastic IP entry, and select "Actions", "Release addresses", then "Release"



Extra Challenge

While we wait for everyone to complete this session....

- At the end of this session the Private server had access to the Internet.
- What are some of the reasons that this server needed access to the Internet?
 - Hint 'AWS'
- Are there other methods for this server to get access to the Internet? Ways that are more controlled?

END

Session 4 - Bootstrapping

In this session we will get introduced to the concept of “Bootstrapping”.

Take the following steps, one at a time. There is plenty of time to complete the session and ask any questions along the way.

Steps

- In this session we are going to get a Windows server to do a “real job”. But we're not going to configure it by hand.
- Log into the AWS console
- Set the region to Sydney (stay in this region today, check from time to time)
- Navigate to EC2
- First let's use the AWS console to build a web server:

Bootstrap an Instance with the Console

- Select "Launch Instance"
- Select "Microsoft Windows Server 2016 Base"
- This time, as we will be getting the server to do some work, and we don't want to wait too long, let's select a bigger instance type:
- Select "t2.large"

Currently selected: t2.large (Variable ECUs, 2 vCPUs, 2.3 GHz, Intel Broadwell E5-2686v4, 8 GiB memory, EBS only)								
	Family	Type	vCPUs	Memory (GiB)	Instance Storage (GB)	EBS-Optimized Available	Network Performance	IPv6 Support
<input type="checkbox"/>	General purpose	t2.nano	1	0.5	EBS only	-	Low to Moderate	Yes
<input type="checkbox"/>	General purpose	t2.micro	1	1	EBS only	-	Low to Moderate	Yes
<input type="checkbox"/>	General purpose	t2.small	1	2	EBS only	-	Low to Moderate	Yes
<input type="checkbox"/>	General purpose	t2.medium	2	4	EBS only	-	Low to Moderate	Yes
<input checked="" type="checkbox"/>	General purpose	t2.large	2	8	EBS only	-	Low to Moderate	Yes
<input type="checkbox"/>	General purpose	t2.xlarge	4	16	EBS only	-	Moderate	Yes
<input type="checkbox"/>	General purpose	t2.2xlarge	8	32	EBS only	-	Moderate	Yes

[Cancel](#) [Previous](#) [Review and Launch](#) [Next: Configure Instance Details](#)

- Select "Next: Configure Instance Details"

- Take a look at the different settings options here.
- We will leave all the default values, but we will scroll down and expand the "Advanced Details":
- Here we see the "User data" entry.

▼ Advanced Details

User data (i)

As text As file Input is already base64 encoded

(Optional)

- Text we add to "User data" will be passed to the Instance as it boots, and if it's script, will be executed by an Administrator user automatically.
- In the resources for this session you will find user-data.txt
- Open up this file in a text editor and take a look. You will see that its some Powershell script that copies an ASPX web page, enables the IIS Web Role, and starts a website.
- Copy the text (completely) and paste it into the "User data" section of the EC2 console.

▼ Advanced Details

User data (i)

As text As file Input is already base64 encoded

```
new-websit... -name test -port 80 -physicalpath c:\inetpub\test  
-ApplicationPool " .NET v4.5" -force  
remove-website -name "Default Web Site"  
start-website -name test  
</powershell>
```

- Select "Next: Add Storage"
- We won't make any changes here, but see how we could change the size and number of disks that the server has.

Volume Type	Device	Snapshot	Size (GiB)	Volume Type	IOPS	Throughput (MB/s)	Delete on Termination	Encrypted
Root	/dev/sda1	snap-049c714bc2eed31ff	30	General Purpose SSD (GP2)	100 / 3000	N/A	<input checked="" type="checkbox"/>	Not Encrypted
Add New Volume								

- Select "Next: Add Tags"
- We want the server configuration to be named

- Select "Add Tag"

- Enter "Name" for the key (note ensure the N is capital)

- Enter "my-web-server" for the value.

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)	In
Name	my-web-server	<input checked="" type="checkbox"/>
Add another tag	(Up to 50 tags maximum)	

- Select "Next: Configure Security Group"

- Let's change the name of the Security Group so we can find it in the future, change the name to "web-server-group"

- Using descriptions is always a good idea in AWS, add a description so that when you look at this in the future you understand what it is

- Now we are going to allow connection on port 80 (HTTP) from the outside world

- Select "Add Rule"

- In the Type dropdown select "HTTP"

- In the "Source" dropdown select "Anywhere"

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:	web-server-group			
Description:	Mike told me to write a description in here...			
Type (i)	Protocol (i)	Port Range (i)	Source (i)	Description (i)
Custom TCP	TCP	3389	Custom	0.0.0.0/0
HTTP	TCP	80	Custom	0.0.0.0/0, ::/0

[Add Rule](#)

- Select: “Review and Launch”

- Review the details of the instance, and if all looks ok select "Launch"

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 instances' security. Your security group, web-server-group, is open to the world.
Your instances may be accessible from any IP address. We recommend that you update your security group rules to allow access from known IP addresses only. You can also open additional ports in your security group to facilitate access to the application or service you're running, e.g., HTTP (80) for web servers. [Edit security groups](#)

⚠ Your instance configuration is not eligible for the free usage tier
To launch an instance that's eligible for the free usage tier, check your AMI selection, instance type, configuration options, or storage devices. Learn more about [free usage tier](#) eligibility and usage restrictions.

[Don't show me this again](#) [Edit AMI](#)

AMI Details

 Microsoft Windows Server 2016 Base - ami-48d6122a
 Free tier eligible Microsoft Windows 2016 Datacenter edition. [English]
Root Device Type: ebs Virtualization type: hvm

If you plan to use this AMI for an application that benefits from Microsoft License Mobility, fill out the [License Mobility Form](#). [Don't show me this again](#)

Instance Type [Edit instance type](#)

Instance Type	ECUs	vCPUs	Memory (GiB)	Instance Storage (GB)	EBS-Optimized Available	Network Performance

[Cancel](#) [Previous](#) [Launch](#)

- Select the Key Pair that you created earlier today

- You will need to acknowledge that you DO have the Key Pair

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](#).

Choose an existing key pair ▾

Select a key pair

my-key-pair ▾

I acknowledge that I have access to the selected private key file (my-key-pair.pem), and that without this file, I won't be able to log into my instance.

Cancel

Launch Instances

- Select: "Launch Instances"

- Scroll to the bottom of the screen and select "View Instances"

Bootstrap an Instance with CloudFormation

- Wow that was a lot of steps to create a server, there must be a better way?
- Well there are a few, but let's do the same thing again with CloudFormation.
- This time we have created the template for you, let's review it, you will find it in this sessions resources called "web-server-instance.yaml"

```
AWSTemplateFormatVersion: 2010-09-09
```

```
Description: This template creates an EC2 web server instance and allows you to select the subnet it will use. To keep it simple it only works in Sydney (ap-southeast-2).
```

```
Parameters:
```

```
serverName:
```

```
Type: String
```

```
Default: my-demo-server
```

```
Description: Enter the name for the server. (This is not a hostname)
```

```
subnet:
```

```
Type: AWS::EC2::Subnet::Id
```

```
Description: Select a subnet for this instance.
```

```
key:
```

```
Type: AWS::EC2::KeyPair::KeyName
```

```
Description: Enter a valid EC2 Key
```

```
Resources:
```

```
instance:
```

```
Type: "AWS::EC2::Instance"
```

```
Properties:
```

```
ImageId: ami-82a458e0
```

```
InstanceType: t2.large
```

```
KeyName: !Ref key
```

```
NetworkInterfaces:
```

```
- DeviceIndex: 0
```

```
SubnetId: !Ref subnet
```

```
AssociatePublicIpAddress: true
```

UserData:

```
Fn::Base64: !Sub |
    <powershell>
        mkdir c:\inetpub\test
        curl https://s3-ap-southeast-2.amazonaws.com/test-aspx/default.aspx
    -OutFile C:\inetpub\test\default.aspx
        add-windowsfeature web-webserver -includeallsubfeature -logpath
$env:temp\webserver_addrole.log
        new-website -name test -port 80 -physicalpath c:\inetpub\test
    -ApplicationPool ".NET v4.5" -force
        remove-website -name "Default Web Site"
        start-website -name test
    </powershell>
```

Tags:

```
-  
    Key: Name  
    Value: !Ref serverName
```

- In the template you will see a "Description", "Parameters", and "Resources" sections.
- Parameters are the inputs you're prompted for when you launch the template.
- Resources are the service instances that will be created in the stack.
- Notice that the EC2 Instance has UserData in the template, the same user data that you pasted into the console earlier.
- Go ahead and create a stack from this template. If you need a hand follow these steps:
 - Navigate to CloudFormation
 - Select "Create Stack"
 - Select "Upload a template to Amazon S3"
 - Select "Browse..." and select the template file from your computer
 - Select "Next"

Specify Details

Specify a stack name and parameter values. You can use or change the default parameter values, which are defined in the AWS CloudFormation template. [Learn more](#).

Stack name my-web-server-stack

Parameters

key my-key-pair

Enter a valid EC2 Key

serverName my-cfn-web-server

Enter the name for the server. (This is not a hostname)

subnet subnet-9bdd84c2 (172.31.16.0/20)

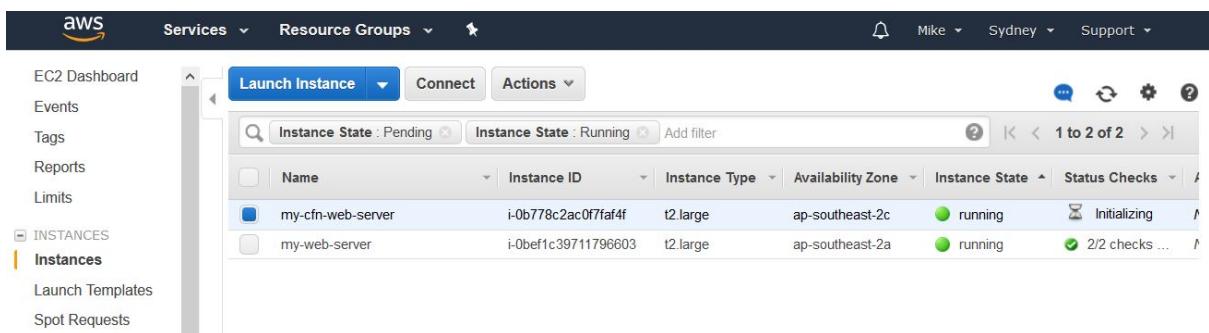
Select a subnet for this instance.

[Cancel](#)

[Previous](#)

[Next](#)

- Enter "my-web-server-stack" for the "Stack name"
- Select your "Key Pair" for the "key"
- Enter a "serverName", let's call this one "my-cfn-web-server"
- For "subnet" choose any subnet in the "172." range.
- Select "Next"
- Scroll down and select "Next"
- Select "Create"
- Now lets navigate back to EC2 and see how our other build is going
- By now we should see 2 EC2 Instances building.



Name	Instance ID	Instance Type	Availability Zone	Instance State	Status Checks
my-cfn-web-server	i-0b778c2ac0f7faf4f	t2.large	ap-southeast-2c	running	Initializing
my-web-server	i-0bef1c39711796603	t2.large	ap-southeast-2a	running	2/2 checks ...

- It can take a few mins for the servers to build. This is mainly them enabling the Web Role and setting up the website.

- While the CloudFormation template was excellent, it didn't enable port 80 for the Security Group. Let's fix this:

- Navigate to VPC

- Select 'Security Group' from the menu on the left hand side.

- Locate and select the security group with the description 'default VPC security group'.

- Select the 'Inbound Rules' tab, and select 'Edit'



- In the new entry, select 'HTTP' for 'Type', and add '0.0.0.0/0' to the 'Source'

- Select 'Save'

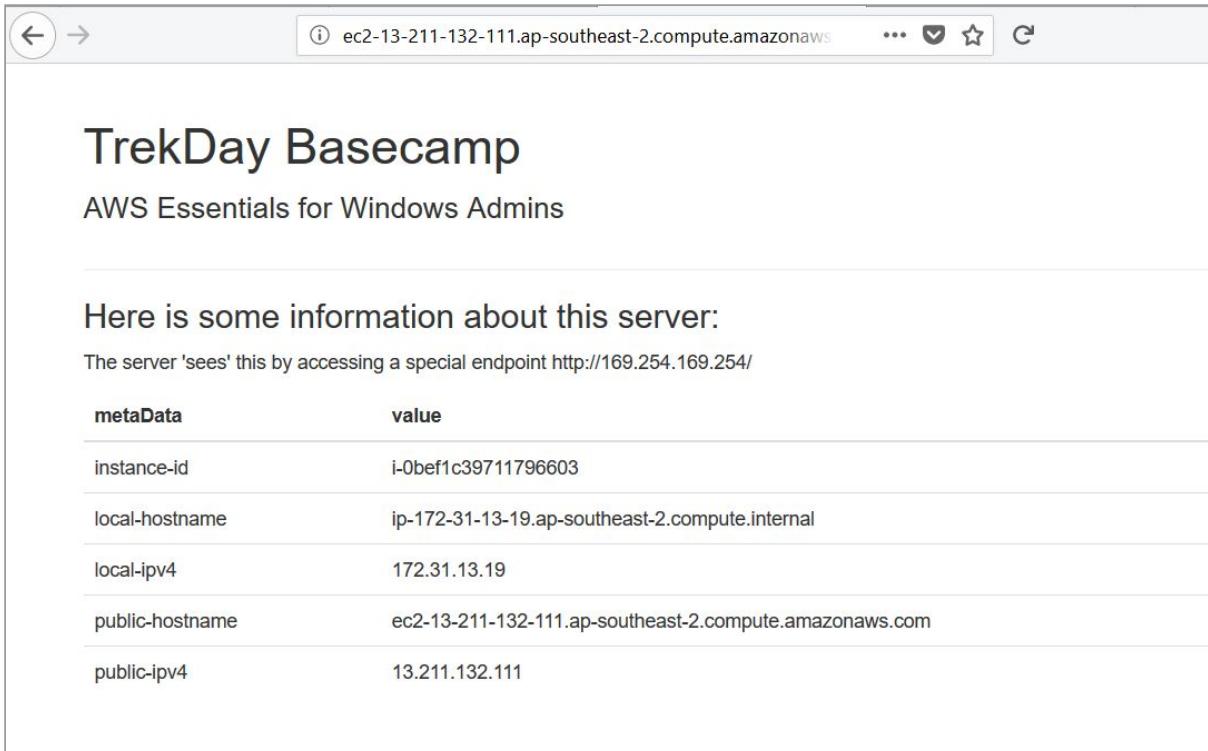
- Ok, let's take a look at our handy work!

- Select the first server we created "my-web-server"

- In the details towards the bottom of the page find and copy the "Public DNS" value for the server.

Description	Status Checks	Monitoring	Tags
Instance ID: i-0bef1c39711796603 (my-web-server)	Public DNS: ec2-13-211-132-111.ap-southeast-2.compute.amazonaws.com		
Instance state: running	IPv4 Public IP: 13.211.132.111		
Instance type: t2.large	IPv6 IPs: -		
Elastic IPs:	Private DNS: ip-172-31-13-19.ap-southeast-2.compute.internal		
Availability zone: ap-southeast-2a	Private IPs: 172.31.13.19		
Security groups: web-server-group, view	Secondary private IPs:		

- Enter this value into a new tab of your web browser.
- Do you see a webpage?

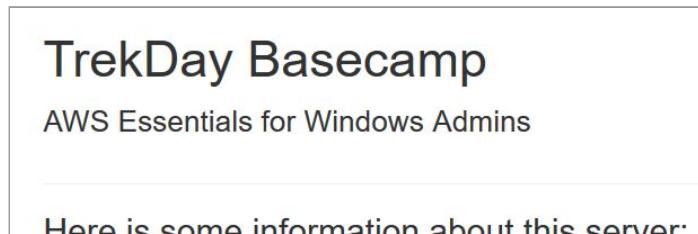


The screenshot shows a web browser window with the following details:

- Address bar: ec2-13-211-132-111.ap-southeast-2.compute.amazonaws.com
- Page title: TrekDay Basecamp
- Subtitle: AWS Essentials for Windows Admins
- Text: "Here is some information about this server:"
- Text: "The server 'sees' this by accessing a special endpoint http://169.254.169.254/"
- Table:

metaData	value
instance-id	i-0bef1c39711796603
local-hostname	ip-172-31-13-19.ap-southeast-2.compute.internal
local-ipv4	172.31.13.19
public-hostname	ec2-13-211-132-111.ap-southeast-2.compute.amazonaws.com
public-ipv4	13.211.132.111

- No need to RDP into the server, the job is done.
- In a few minutes, do the same with your Cloudformation server.
- You should see a webpage there too.



The screenshot shows a web browser window with the following details:

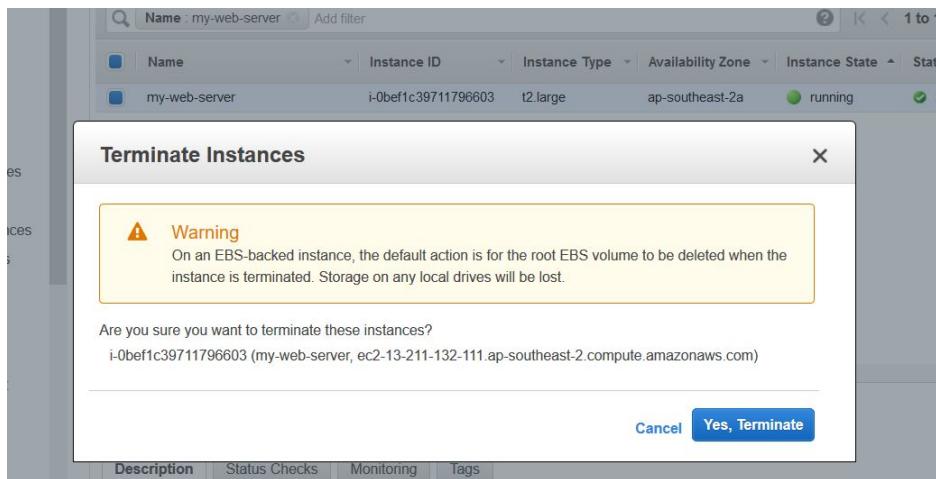
- Page title: TrekDay Basecamp
- Subtitle: AWS Essentials for Windows Admins
- Text: "Here is some information about this server..." (partially visible)

- Discuss: How could we get this build to be even faster?

Cleanup the Resources

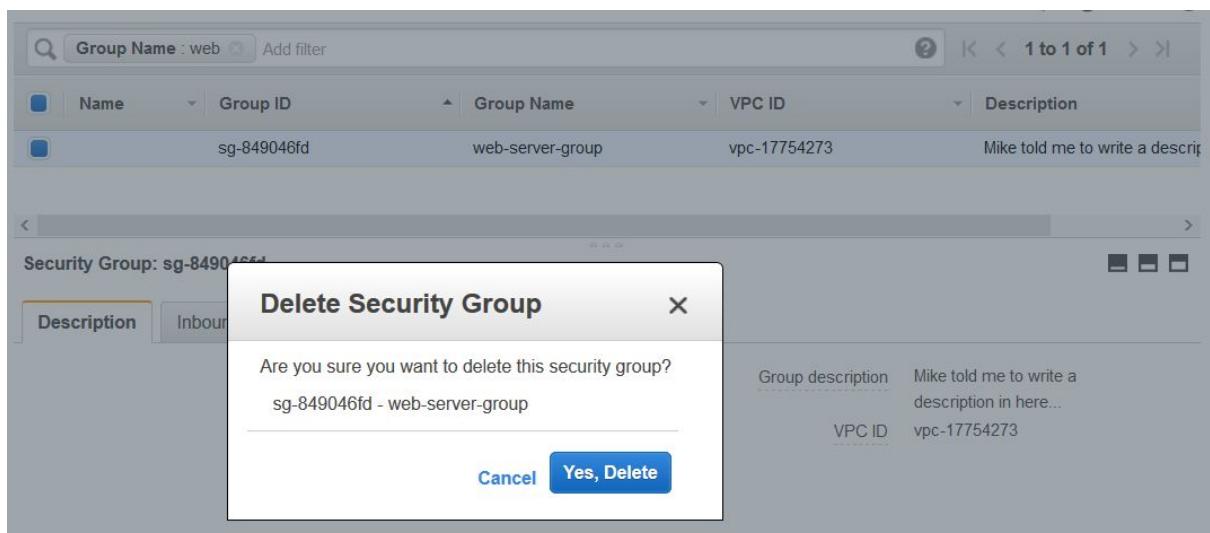
- We don't need these servers anymore. Let's clean up:

- Navigate to EC2
- Select the instance called "my-web-server"
- Select "Actions", "Instance State", "Terminate" and "Yes, Terminate"



- From the menu on the left hand side, select "Security Groups"

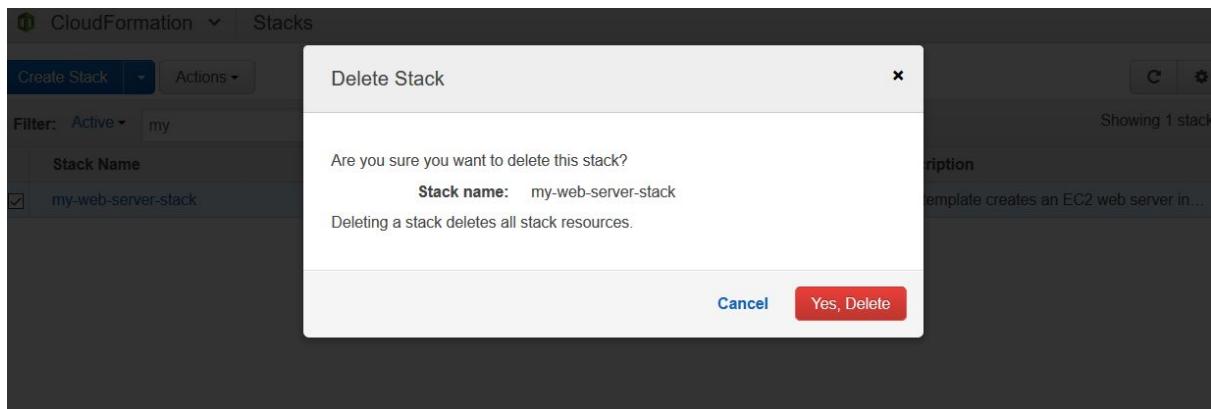
- Look for a Security Group with the GROUP NAME (note this is not the "Name" column) of "web-server-group", select "Actions", "Delete Security Group", and "Yes, Delete".



- Cleaning up the CloudFormation server/stack is much easier

- Navigate to CloudFormation

- Select your web server stack, then "Actions", "Delete Stack", "Yes, Delete"



Extra Challenge

While we wait for everyone to complete this session....

- What other things could you do during a boot-strap?
- Could you pass in credentials to do tasks such as domain join?

END

Session 5 - AWS Systems Manager

In this session we will get introduced to AWS Systems Manager (SSM).

Take the following steps, one at a time. There is plenty of time to complete the session and ask any questions along the way.

Steps

- First we're going to set up a (small) fleet of servers to manage.
- Navigate to CloudFormation
- Select "Create Stack"
- Select "Upload a template to Amazon S3" and "Browse..."
- With the resources for this session find "three-server-instances.yaml"
- Select "Next"
- For the stack name enter "my-instance-fleet"

Specify Details

Specify a stack name and parameter values. You can use or change the default parameter values, which are defined in the AWS CloudFormation template. [Learn more](#).

Stack name

[Cancel](#) [Previous](#) [Next](#)

- Select "Next"
- Scroll to the bottom and select "Next"

Capabilities

i The following resource(s) require capabilities: [AWS::IAM::Role]

This template contains Identity and Access Management (IAM) resources that might provide entities access to make changes to your AWS account. Check that you want to create each of these resources and that they have the minimum required permissions.

[Learn more.](#)

I acknowledge that AWS CloudFormation might create IAM resources.

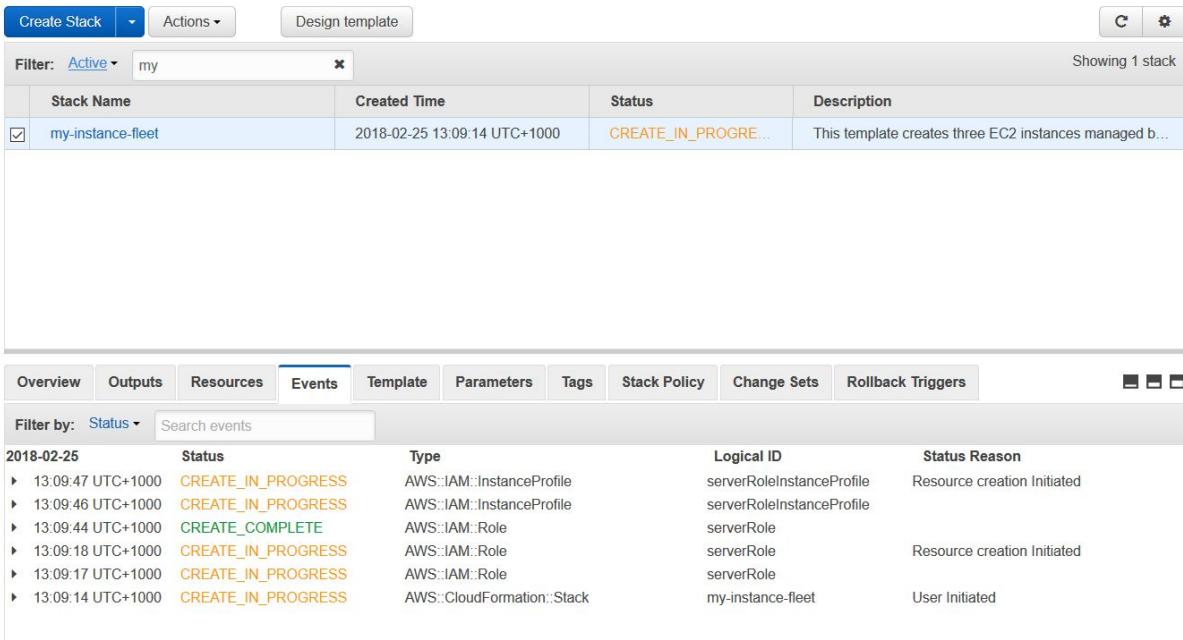
[Quick Create Stack](#) (Create stacks similar to this one, with most details auto-populated)

[Cancel](#) [Previous](#) [Create](#)

- As this template will adjust the security of your account you will now need to check the box for "I acknowledge that AWS CloudFormation might create IAM resources."

- Select "Create"

- Wait a minute or two while the instances build. You can watch them build from the "Events" tab in the CloudFormation stack page, and from the Instance list in the EC2 page.



Created Time	Status	Description
2018-02-25 13:09:14 UTC+1000	CREATE_IN_PROGRESS	This template creates three EC2 instances managed b...

Event	Type	Logical ID	Status Reason
13:09:47 UTC+1000	AWS::IAM::InstanceProfile	serverRoleInstanceProfile	Resource creation Initiated
13:09:46 UTC+1000	AWS::IAM::InstanceProfile	serverRoleInstanceProfile	
13:09:44 UTC+1000	AWS::IAM::Role	serverRole	
13:09:18 UTC+1000	AWS::IAM::Role	serverRole	Resource creation Initiated
13:09:17 UTC+1000	AWS::IAM::Role	serverRole	
13:09:14 UTC+1000	AWS::CloudFormation::Stack	my-instance-fleet	User Initiated

- The stack can take longer than you might expect to create the "AWS::IAM::InstanceProfile", but things should speed along after that step is complete.

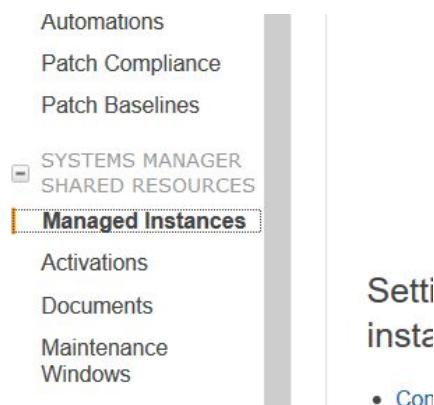
- Still waiting? Have a look at the template you are building. In some ways its more complex than other templates we've built.

...

- Now the instances are built, let's check to make sure that SSM can 'see' them.

- Navigate to EC2

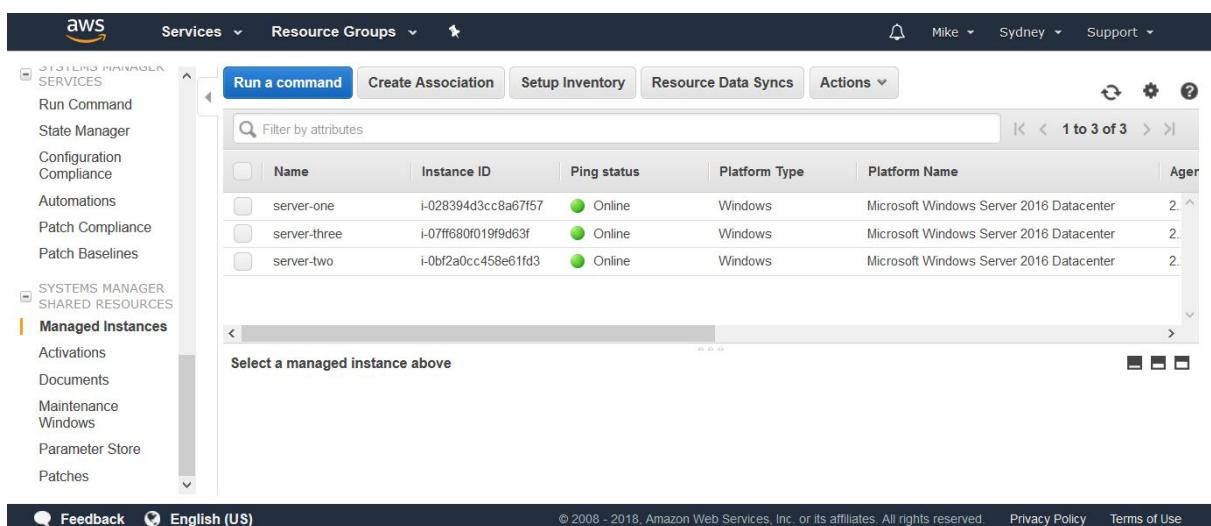
- From the menu on the left hand side, scroll down and select "Managed Instances" under the heading "SYSTEMS MANAGER SHARED RESOURCES".



- If there is nothing listed yet, the SSM service might not have discovered the Instances yet. Wait a moment.

- When we created the Instances (via CloudFormation) we gave them the security rights to communicate with SSM. We are now waiting for the SSM agent on the server to poll the services.

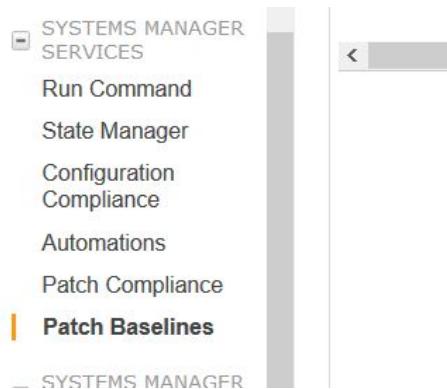
- Refresh the page (maybe a few times) until you see all three servers listed.



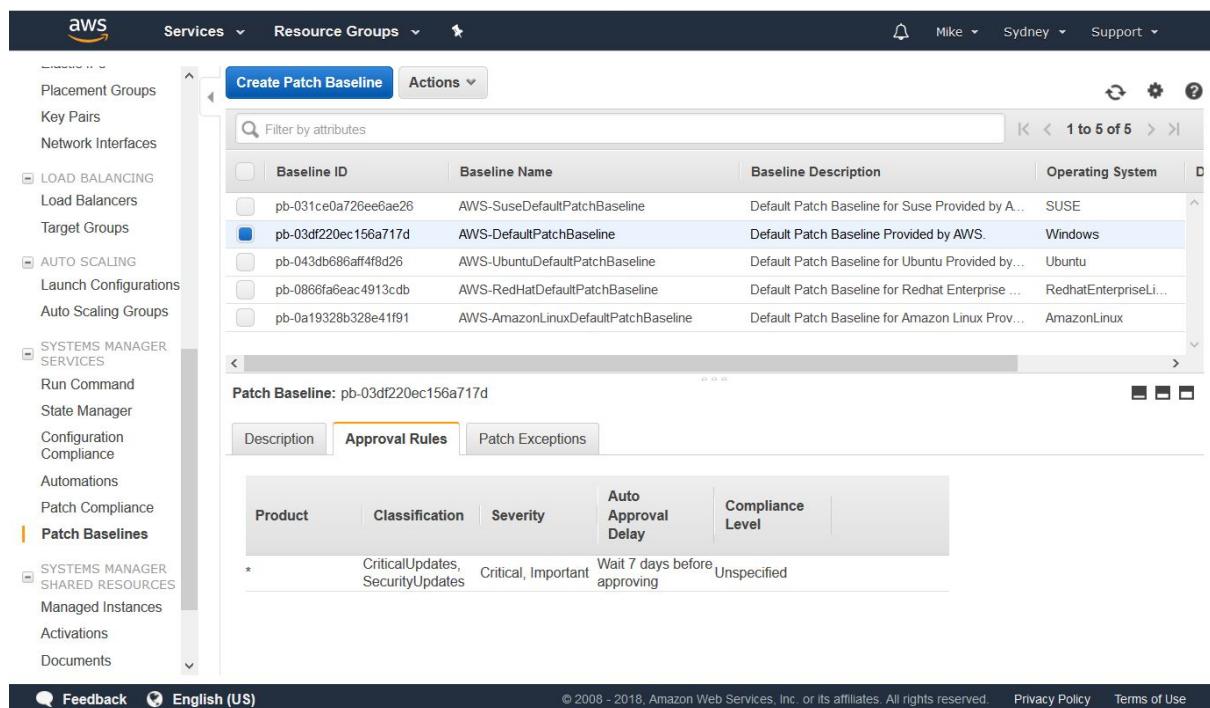
The screenshot shows the AWS Systems Manager Shared Resources - Managed Instances page. The left sidebar lists 'Run Command', 'State Manager', 'Configuration', 'Compliance', 'Automations', 'Patch Compliance', 'Patch Baselines', and 'SYSTEMS MANAGER SHARED RESOURCES' (with 'Managed Instances' selected). The main content area has a header with 'Run a command', 'Create Association', 'Setup Inventory', 'Resource Data Syncs', and 'Actions'. Below is a table with columns: Name, Instance ID, Ping status, Platform Type, and Platform Name. The table contains three rows: 'server-one' (Instance ID: i-028394d3cc8a67f57, Ping status: Online, Platform Type: Windows, Platform Name: Microsoft Windows Server 2016 Datacenter), 'server-three' (Instance ID: i-07ff680f019fd63f, Ping status: Online, Platform Type: Windows, Platform Name: Microsoft Windows Server 2016 Datacenter), and 'server-two' (Instance ID: i-0bf2a0cc458e61fd3, Ping status: Online, Platform Type: Windows, Platform Name: Microsoft Windows Server 2016 Datacenter). At the bottom, there is a note 'Select a managed instance above' and footer links for Feedback, English (US), Copyright notice (© 2008 - 2018, Amazon Web Services, Inc. or its affiliates. All rights reserved.), Privacy Policy, and Terms of Use.

SSM - Check server patch status

- First let's check the patch status of our Instances against the default Patch Baseline.
- To see the default baseline select "Patch Baselines" from the menu on the left hand side.



- You will see several Default Baselines provided by AWS. Select the one with the name "AWS-DefaultPatchBaseline" as this applies to Windows.



The screenshot shows the AWS Systems Manager Patch Baselines page. The 'AWS-DefaultPatchBaseline' is selected and highlighted. The details for this baseline are displayed below:

Baseline ID	Baseline Name	Baseline Description	Operating System
pb-03df220ec156a717d	AWS-DefaultPatchBaseline	Default Patch Baseline Provided by AWS.	Windows
pb-031ce0a726ee6ae26	AWS-SuseDefaultPatchBaseline	Default Patch Baseline for Suse Provided by A...	SUSE
pb-043db686aff4f8d26	AWS-UbuntuDefaultPatchBaseline	Default Patch Baseline for Ubuntu Provided by...	Ubuntu
pb-0866fa6eac4913cdb	AWS-RedHatDefaultPatchBaseline	Default Patch Baseline for Redhat Enterprise ...	RedhatEnterpriseLi...
pb-0a19328b328e41f91	AWS-AmazonLinuxDefaultPatchBaseline	Default Patch Baseline for Amazon Linux Prov...	AmazonLinux

Patch Baseline: pb-03df220ec156a717d

Approval Rules

Product	Classification	Severity	Auto Approval Delay	Compliance Level
*	CriticalUpdates, SecurityUpdates	Critical, Important	Wait 7 days before approving	Unspecified

- In the lower section of the screen select "Approval Rules"

- You can see that this Baseline defines all patches with:

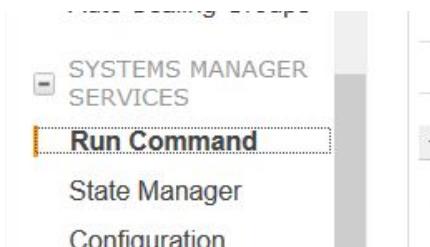
- Classification of "CriticalUpdates, SecurityUpdates"
- Severity of Critical, Important
- Older than 7 days

Description	Approval Rules	Patch Exceptions		
Product	Classification	Severity	Auto Approval Delay	Compliance Level
*	CriticalUpdates, SecurityUpdates	Critical, Important	Wait 7 days before approving	Unspecified

- Let's check our servers against this baseline.

- Normally we would do this using a "Maintenance Window", or "State Manager" that checks periodically, but today we will run a one off check using "Run Command"

- Select "Run Command" from the menu on the left hand side.



- Select "Run a command"

- Under "Command document" select "AWS-RunPatchBaseline". This is a prewritten document to check any type of instance against a Patch Baseline.

<input type="radio"/>	AWS-InstallApplication	Amazon	Windows
<input type="radio"/>	AWS-JoinDirectoryServiceDomain	Amazon	Windows
<input checked="" type="radio"/>	AWS-RunPatchBaseline	Amazon	Windows,Linux
<input type="radio"/>	AWS-InstallSpecificWindowsUpdates	Amazon	Windows
<input type="radio"/>	AWS-RunShellScript	Amazon	Linux

- Scroll down and select "Manually Selecting Instances"

Select Targets by* Manually Selecting Instances Specifying a Tag

i-028394d3cc8a67f57 i-07ff680f019f9d63f i-0bf2a0cc458e61fd3 ⓘ

Select instances ▾

Where are my instances? ⚙						
	Name	Instance ID	Instance State	Availability Zone	Ping Status	Last Ping DateT Agent Version
<input type="checkbox"/>	server-one	i-028394d3cc8a67f57	running	ap-southeast-2a	Online	February 25, 2018 2.2.93.0
<input type="checkbox"/>	server-three	i-07ff680f019f9d63f	running	ap-southeast-2a	Online	February 25, 2018 2.2.93.0
<input type="checkbox"/>	server-two	i-0bf2a0cc458e61fd3	running	ap-southeast-2a	Online	February 25, 2018 2.2.93.0

- Select "Select instances" and select all of your three instances.

Execute on Targets concurrently

Stop after errors

Operation* ⓘ

- Enter "3" for "Execute on", this specifies how many instances of the document should run at once across our fleet.

- Enter "1" for "Stop after" ... "errors".

- For "Operation" select "Scan". We are not going to install missing patches, but rather scan for missing patches.

Timeout (seconds) ⏱

▶ Advanced Options

▶ AWS Command Line Interface command

Cancel

- Scroll to the bottom and select "Run"

- Ensure on the next page that there are entries under "Instance IDs"

- Select "View result"

✓ Success
We are running your command against the instances listed below.

Instance IDs i-028394d3cc8a67f57, i-07ff680f019f9d63f, i-0bf2a0cc458e61fd3

Command ID 5d4fda43-82a8-45bc-8f46-08848572317f

[View result](#)

- You should see a list of commands "In Progress", the list has a refresh button on it so you can watch the progress.

[Run a command](#) [Actions ▾](#)

Filter by attributes

	Command ID	Instance ID	Document name	Status
<input type="checkbox"/>	a6fafafa84-fdc3-4a1b-...	i-07ff680f019f9d63f	AWS-RunPatchBaseline	In Progress
<input type="checkbox"/>	a6fafafa84-fdc3-4a1b-...	i-028394d3cc8a67f57	AWS-RunPatchBaseline	In Progress
<input type="checkbox"/>	5d4fda43-82a8-45b...	i-0bf2a0cc458e61fd3	AWS-RunPatchBaseline	In Progress

- When any of the commands are complete, shown as "Success" you can view the output.

- Select the command from the list, in the lower section of the screen select the "Output" tab.

[Description](#) [Output](#)

Output

Plugin name	Status	Response code	Start Time	Finish Time	Output
PatchLinux	Success	0	February 25, 2018 at 1:26:56 PM...	February 25, 2018 at 1:26:56 PM...	View Output
PatchWindows	Success	0	February 25, 2018 at 1:22:38 PM...	February 25, 2018 at 1:26:56 PM...	View Output

- In the output list you should see two entries. The command/document that we ran operates on both Windows and Linux OS's. There should be one output per OS type, and only any useful information in the Windowseat man output.

- You will (probably) see that "Scan found no missing updates."

[Commands](#) > Output

Output for PatchWindows

```
Patch Summary for i-0bf2a0cc458e61fd3
PatchGroup      :
BaselineId      : pb-03df220ec156a717d
SnapshotId     : d4b95834-a53f-4a19-b1ef-cbb7d75f3426
OwnerInformation :
OperationType   : Scan
OperationStartTime : 2018-02-25T03:22:44.2777320Z
OperationEndTime  : 2018-02-25T03:26:53.8367563Z
InstalledCount   : 0
InstalledOtherCount : 10
FailedCount      : 0
MissingCount     : 0
NotApplicableCount : 2390

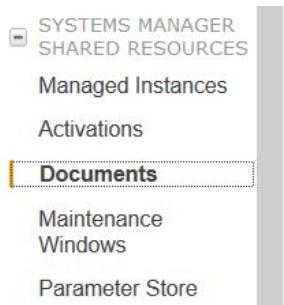
EC2AMAZ-95VG6Q9 - PatchBaselineOperations Assessment Results -
2018-02-25T03:26:56.461

Scan found no missing updates.
```

- Great.

SSM - Run our own command/document

- Now let's create our own document, and run a bigger command.
- Navigate to EC2, scroll down the left hand menu and select "Documents"



- Select "Create Document"
- Enter "myWebServerInstall" into the "Name"
- Make sure that "Document Type" is set to "Command"

Create Document

Specify the following parameters to create a document

Name*	<input type="text" value="myWebServerInstall"/>	
Document Type	<input type="text" value="Command"/>	

- In the resources for this session you will find a file called "ssm-document.json". Open this in a text editor and copy all of the content to the clipboard.
- Under "Content" delete the curly braces "{}" and paste in the contents of your clipboard.

Content*

```

7   "name": "SetupWebsite",
8   "inputs": {
9     "timeoutSeconds": 300,
10    "runCommand": [
11      "mkdir c:\\inetpub\\test",
12      "",
13      "curl https://s3-ap-southeast-2.amazonaws.com/test-aspx/def",
14      "",
15      "add-windowsfeature web-webserver -includeallsubfeature -lo",
16      "",
17      "new-website -name test -port 80 -physicalpath c:\\inetpub\\",
18      "",
19      "remove-website -name \"Default Web Site\"",
20      "",
21      "start-website -name test"
22    ]
23  }
24
25
26

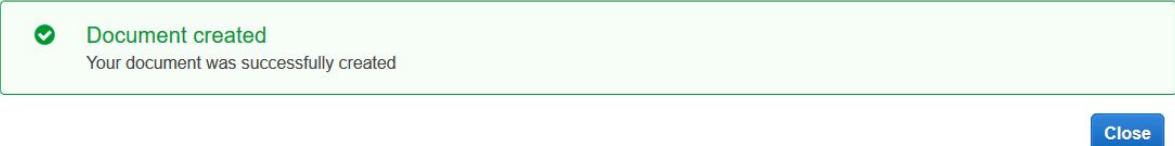
```



- There will be some time in a moment to review the document.

- Scroll down and select "Create Document", then "Close"

Create Document

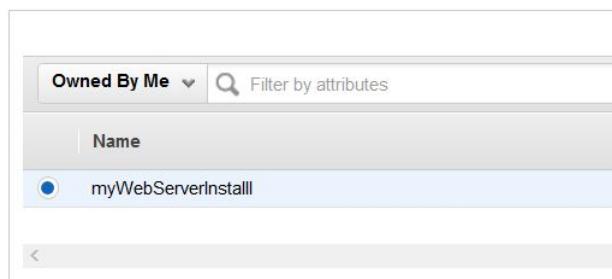


- Let's run this document as a command against our fleet.

- Select "Run Command" from the menu on the left hand side.

- Select "Run a command"

- Under "Command document" change the dropdown filter to "Owned By Me"



- Select your document "myWebServerInstall"

- Again select "Manually Selecting Instances" and select all your instances from "Select instances"

Select Targets by* Manually Selecting Instances Specifying a Tag

i-028394d3cc8a67f57 i-07ff680f019f9d63f i-0bf2a0cc458e61fd3 

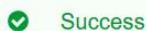
Select instances ▾

Filter by attributes					
<input type="checkbox"/>	Name	Instance ID	Instance State	Availability Zone	Ping Status
<input type="checkbox"/>	server-one	i-028394d3cc8...	running	ap-southeast-2a	Online
<input type="checkbox"/>	server-three	i-07ff680f019f9...	running	ap-southeast-2a	Online
<input type="checkbox"/>

- Again select "3" and "1" for "Execute on" and "Stop after"

- Scroll down and select "Run", ensure the instances are listed, and select "View result"

Run a command



Success

We are running your command against the instances listed below.

Instance IDs i-028394d3cc8a67f57, i-07ff680f019f9d63f, i-0bf2a0cc458e61fd3

Command ID 6b919209-98c9-4251-b294-8aeab9a8b405

View result

- This might take a couple of minutes.

....

- Let's go and take another look at the result of the patch baseline we did.

- From the menu on the left hand side select "Patch Compliance"

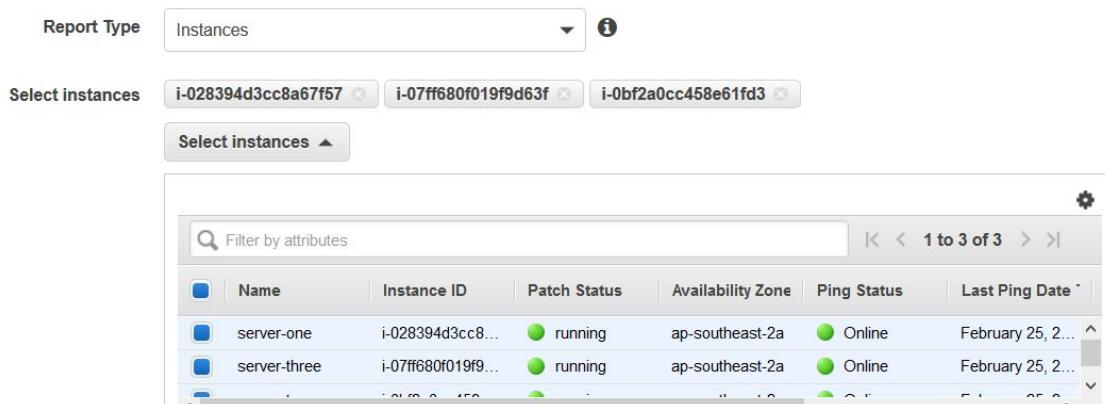


- This page will show a report on the patch compliance of our managed fleet.

- In the "Select instances" dropdown, select all our instances

Patch Compliance

Patch compliance reporting allows you to view compliance information across a number of different axis. Select the report type below to see a summary of the information that you are interested in.



The screenshot shows the 'Patch Compliance' dashboard. At the top, there is a 'Report Type' dropdown set to 'Instances'. Below it is a 'Select instances' dropdown containing three instance IDs: i-028394d3cc8a67f57, i-07ff680f019f9d63f, and i-0bf2a0cc458e61fd3. A 'Select instances' button is also present. Below these, a table displays patch compliance information for three instances:

Name	Instance ID	Patch Status	Availability Zone	Ping Status	Last Ping Date
server-one	i-028394d3cc8a67f57	running	ap-southeast-2a	Online	February 25, 2018
server-three	i-07ff680f019f9d63f	running	ap-southeast-2a	Online	February 25, 2018
	i-0bf2a0cc458e61fd3	running	ap-southeast-2a	Online	February 25, 2018

- The dashboard should update with a review of the current patch status.

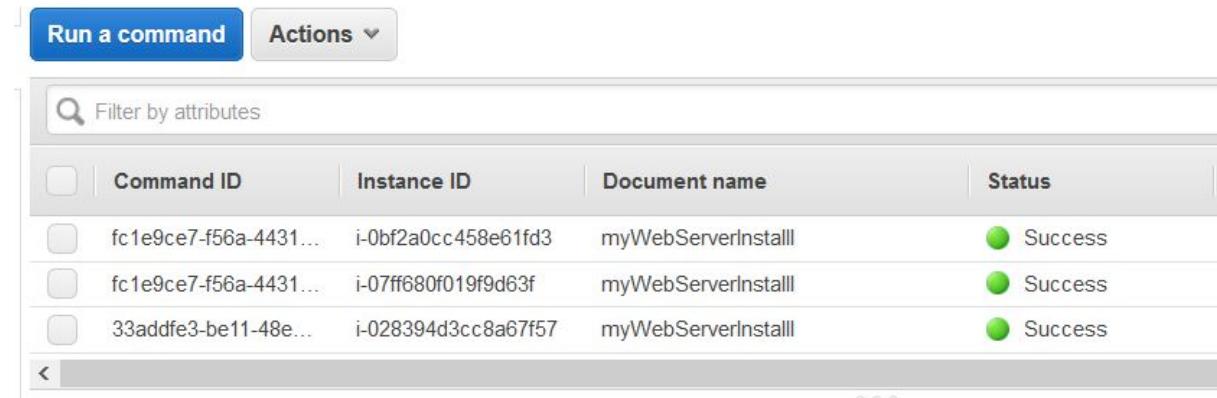
Compliance Summary



- This is a useful tool, especially if you are running "AWS-RunPatchBaseline" on a regular basis.

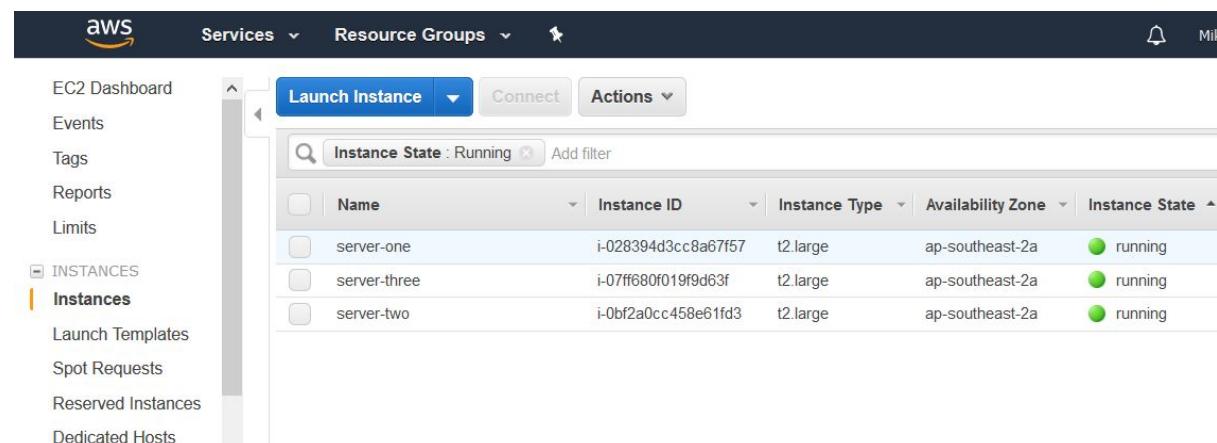
- To catch up on the status of our web server installs, select "Run Command" again from the left hand menu.

- When the commands are showing as "Success"



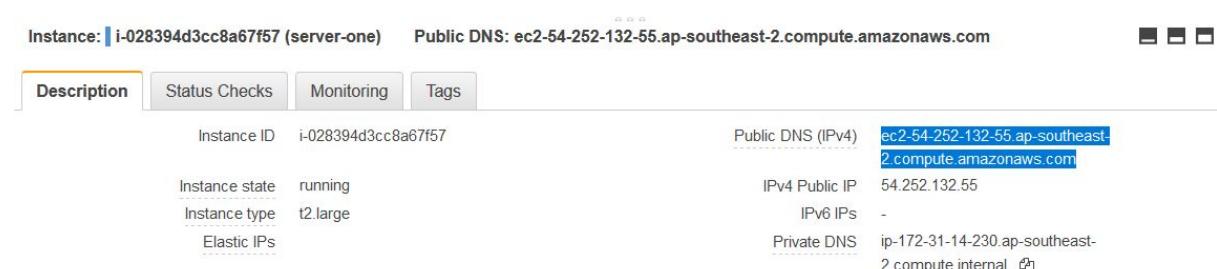
Command ID	Instance ID	Document name	Status
fc1e9ce7-f56a-4431...	i-0bf2a0cc458e61fd3	myWebServerInstall	Success
fc1e9ce7-f56a-4431...	i-07ff680f019f9d63f	myWebServerInstall	Success
33addfe3-be11-48e...	i-028394d3cc8a67f57	myWebServerInstall	Success

- Navigate to EC2 Instances



Name	Instance ID	Instance Type	Availability Zone	Instance State
server-one	i-028394d3cc8a67f57	t2.large	ap-southeast-2a	running
server-three	i-07ff680f019f9d63f	t2.large	ap-southeast-2a	running
server-two	i-0bf2a0cc458e61fd3	t2.large	ap-southeast-2a	running

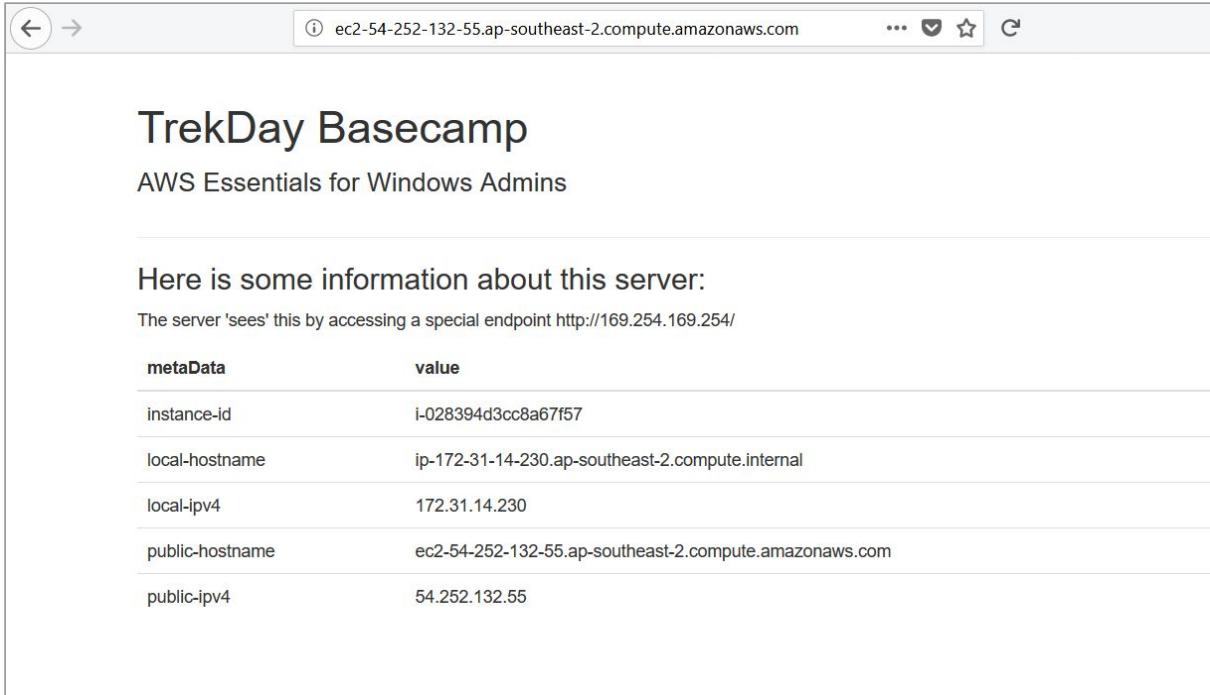
- Locate the "Public DNS" for the instances and test the website from the details section at the bottom of the screen.



Instance: i-028394d3cc8a67f57 (server-one)	Public DNS: ec2-54-252-132-55.ap-southeast-2.compute.amazonaws.com		
Description	Status Checks	Monitoring	Tags
Instance ID: i-028394d3cc8a67f57	Public DNS (IPv4): ec2-54-252-132-55.ap-southeast-2.compute.amazonaws.com		
Instance state: running	IPv4 Public IP: 54.252.132.55		
Instance type: t2.large	IPv6 IPs: -		
Elastic IPs:	Private DNS: ip-172-31-14-230.ap-southeast-2.compute.internal		

- You may have to wait a few moments after the command is complete before the website appears.

- Issue: Sometimes the run commands don't show in the list. Try creating them again.
- **Open a new tab in your browser and navigate to the Public DNS name of one of the instances:**



The screenshot shows a web browser window with the URL `ec2-54-252-132-55.ap-southeast-2.compute.amazonaws.com`. The page title is "TrekDay Basecamp" and the subtitle is "AWS Essentials for Windows Admins". Below this, a section titled "Here is some information about this server:" displays the following table of metadata:

metaData	value
instance-id	i-028394d3cc8a67f57
local-hostname	ip-172-31-14-230.ap-southeast-2.compute.internal
local-ipv4	172.31.14.230
public-hostname	ec2-54-252-132-55.ap-southeast-2.compute.amazonaws.com
public-ipv4	54.252.132.55

- **It's our website again.**
- **Fancy logging into the Instance via RDP? Well you can't we didn't use a Key Pair, this server is 100% not a pet.**
- **No need to clean up, leave the server fleet there, we'll be using it in the next session.**

Extra Challenge

While we wait for everyone to complete this session....

- Let's talk about how we can use SSM to get logging data from the EC2 Instance into CloudWatch Logs

END

Session 6 - Security

In this session we will get introduced to Amazon Inspector.

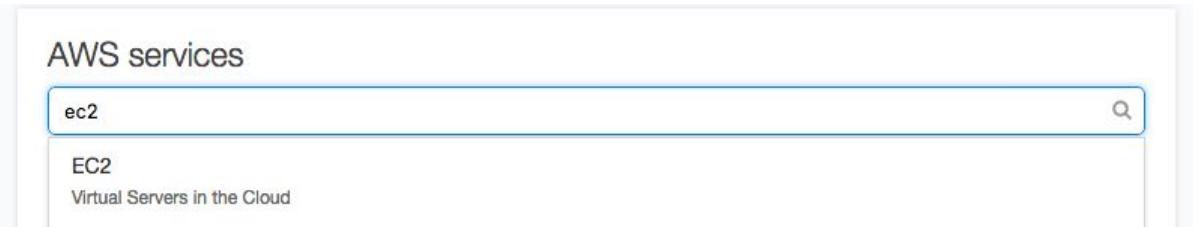
Take the following steps, one at a time. There is plenty of time to complete the session and ask any questions along the way.

Steps

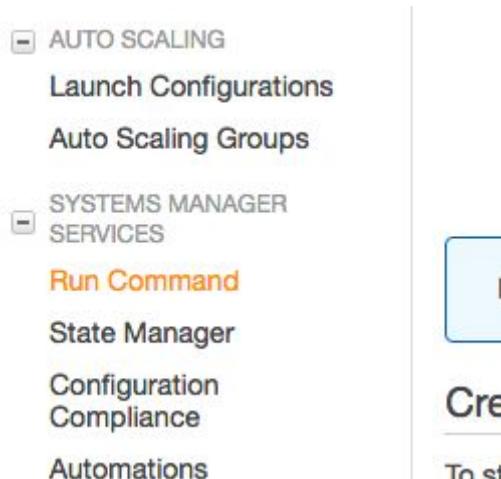
- Do you still have your small fleet of servers?
- If you deleted the stack after the SSM session, now would be a good time to recreate it!
- We're going to start this session off, back in SSM.

Run an SSM Command

- Navigate to EC2



- From the menu on the left hand side select "Run Command"



- We're going to install the "AWS Inspector" agent on our fleet.

- Select "Run a command"

Welcome to EC2 Systems Manager

Use EC2 Systems Manager to easily configure and manage your on-premises machines configured for Systems Manager.

Run Command automates common administrative tasks such as running Linux shell scripts or Windows PowerShell commands.

Before you can run a command, you must configure your targets by adding them to a fleet and setting up permissions as described in the Help topic for the target type. You can activate your VMs with Systems Manager using the activation process.

[Run a command](#)[More about Run Command](#)**- Select "AmazonInspector-ManageAWSAgent"**

Name	Owner	Platform type
AWS-InstallSpecificWindowsUpdates	Amazon	Windows
AWS-RunShellScript	Amazon	Linux
AWS-ConfigureCloudWatch	Amazon	Windows
AWS-RunPowerShellScript	Amazon	Windows,Linux
AWS-ApplyPatchBaseline	Amazon	Windows
AWS-UpdateEC2Config	Amazon	Windows
AWS-InstallWindowsUpdates	Amazon	Windows
AWS-InstallMissingWindowsUpdates	Amazon	Windows
AWS-Support-RunEC2RescueForWindowsTool	Amazon	Windows
AmazonInspector-ManageAWSAgent	Amazon	Windows,Linux
AWSEC2-CreateVssSnapshot	Amazon	Windows
AWSEC2-RunSysprep	Amazon	Windows
AWSEC2-ManageVssIO	Amazon	Windows
AmazonCloudWatch-MigrateCloudWatchAgent	Amazon	Windows
AmazonCloudWatch-ManageAgent	Amazon	Windows,Linux

- From the "Select instances" dropdown, select all your instances

Select instances ▾								
Where are my instances? ⚙								
Filter by attributes								
Name	Instance ID	Instance State	Availability Zone	Ping Status	Last Ping DateT	Agent Version	Platform Type	F
server-three	i-0499adaecea...	running	ap-southeast-2b	Online	February 25, 2...	2.2.93.0	Windows	⋮
server-one	i-0969c84d54a...	running	ap-southeast-2b	Online	February 25, 2...	2.2.93.0	Windows	⋮
server-two	i-0fa4c2402ae...	running	ap-southeast-2b	Online	February 25, 2...	2.2.93.0	Windows	⋮

Close

- Enter "3" and "1" for "Execute on" and "Stop after"

- For "Operation" ensure that "Install" is selected

Execute on	3	Targets	concurrently
Stop after	1	errors	
Operation*	Install	<i>?</i>	

- Scroll down and select "Run", then "View result"

Run a command

✓ Success
We are running your command against the instances listed below.

Instance IDs i-0fa4c2402aaeae6e33, i-0969c84d54a607bb2, i-0499adaecea7df3b6

Command ID 2aa166b7-c6d8-4388-91b4-2fbe551027dc

[View result](#)

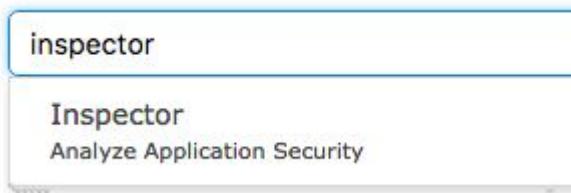
- When the command is a "Success" the Amazon Inspector Agent is installed on the instance.

Run a command Actions ▾

Filter by attributes

	Command ID	Instance ID	Document name	Status	Requested date	C
<input type="checkbox"/>	2aa166b7-c6d8-438...	i-0fa4c2402aeae6e33	AmazonInspector-M...	Success	February 25, 2018 ...	-
<input type="checkbox"/>	2aa166b7-c6d8-438...	i-0969c84d54a607bb2	AmazonInspector-M...	Success	February 25, 2018 ...	-
<input type="checkbox"/>	2aa166b7-c6d8-438...	i-0499adaecea7df3b6	AmazonInspector-M...	Success	February 25, 2018 ...	-

- Navigate to "Amazon Inspector"



- Select "Get Started"



- You should now see the "Get started with Amazon Inspector" page

- Notice that an IAM role will be created for us called "AWSServiceRoleForAmazonInspector", AWS creates this on our behalf.

- Select "Next"

- Under "Define an assessment target"

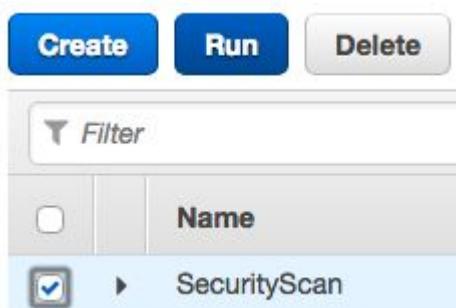
- Enter a name for the assessment "WindowsDevAssessment"
- Under Tags, select Key "Environment" and Value "dev"
- Select "Next"
- Under "Define an assessment template"
- Enter a name of "SecurityScan"
- Expand the "Rules packages" dropdown and review the types of packages that can be run.
- For now select "Common Vulnerabilities and Exposures-1.1"
- For "Duration" select "1 Hour (Recommended)"
- Select "Next" and "Create"

Define an assessment template

An assessment template allows you to specify various properties for an assessment run, including rules packages, duration, SNS notifications, and labels to label any findings. [Learn more](#).

Name*	<input type="text" value="SecurityScan"/>
Rules packages*	Common Vulnerabilities and Exposures-1.1 <input type="button" value="Select an Inspector rules package"/>
<small>Amazon Inspector runs assessments for the assessment target against selected rules package(s). Learn more</small>	
Duration*	<input type="button" value="1 Hour (Recommended)"/>
<small>The default Amazon Inspector assessment template duration is 1 hour. You can modify the duration, but longer assessment templates with longer durations can deliver fuller sets of findings.</small>	

- Now that we have defined an assessment, let's run it
- Select "SecurityScan" from the list
- Select "Run"



A screenshot of the AWS CloudWatch Metrics console. At the top, there are three buttons: "Create" (blue), "Run" (blue), and "Delete" (grey). Below the buttons is a search bar labeled "Filter". A table follows, with the first column containing checkboxes and the second column containing names. The row for "SecurityScan" has a checked checkbox in the first column and "SecurityScan" in the second column.

	Name
<input type="checkbox"/>	SecurityScan

- Notice that the "Last run" column shows "Collecting data"

	Name	Duration	Target name	Last run	All runs
<input type="checkbox"/>	SecurityScan	1 Hour	WindowsDevAssessment	Collecting data	1

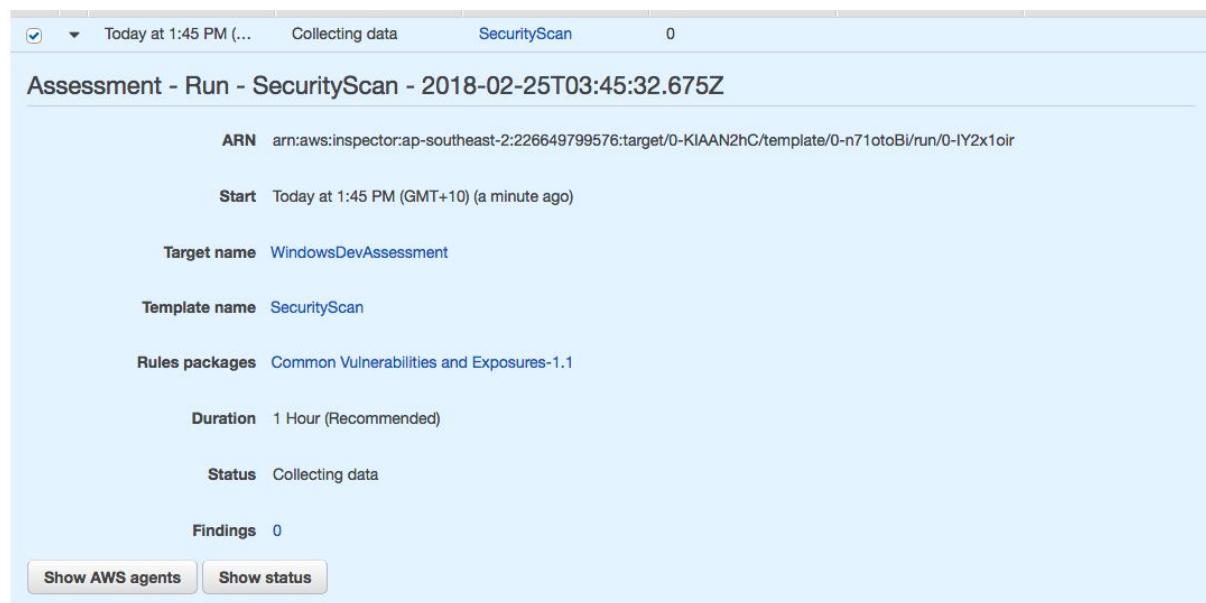
- If you navigate around while the run is collecting data, and want to check on the progress navigate to "Assessment runs" on the left hand side and expand the row entry to see the "Status"



The screenshot shows the AWS Inspector interface under the 'Assessment runs' section. A single row is selected, showing the following details:

Start time	Status	Template name	Findings	Findings by severity	Reports
Today at 1:45 PM (...)	Collecting data	SecurityScan	0		

- Click "Show status" for a detailed report on the scan progress.

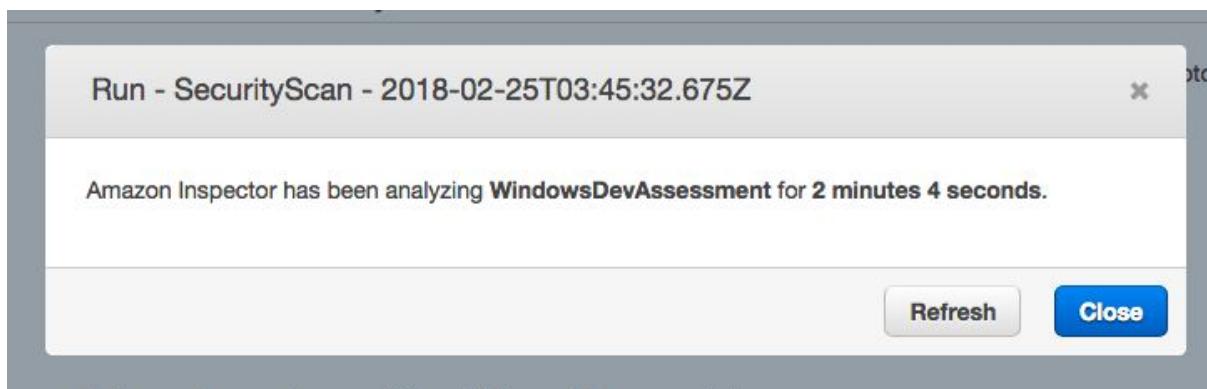


The screenshot shows the 'Show status' view for the 'SecurityScan' run. It displays the following details:

- ARN: arn:aws:inspector:ap-southeast-2:226649799576:target/0-KIAAN2hC/template/0-n71otoBi/run/0-lY2x1oir
- Start: Today at 1:45 PM (GMT+10) (a minute ago)
- Target name: WindowsDevAssessment
- Template name: SecurityScan
- Rules packages: Common Vulnerabilities and Exposures-1.1
- Duration: 1 Hour (Recommended)
- Status: Collecting data
- Findings: 0

At the bottom, there are two buttons: 'Show AWS agents' and 'Show status'.

- The scan will take some time, so lets come back to this later today.



The screenshot shows a modal window titled 'Run - SecurityScan - 2018-02-25T03:45:32.675Z'. The message inside the window reads: "Amazon Inspector has been analyzing WindowsDevAssessment for 2 minutes 4 seconds." At the bottom right are 'Refresh' and 'Close' buttons.

END

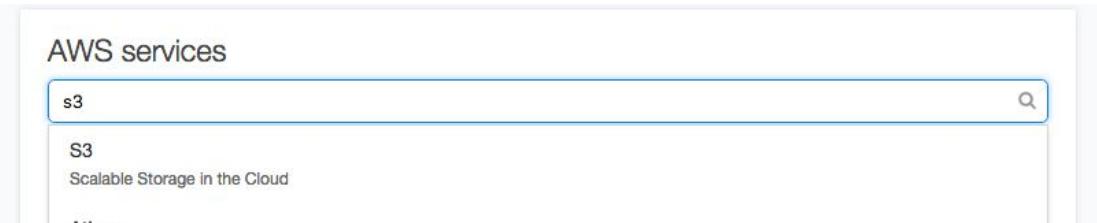
Session 7 - Serverless

In this session we will get introduced to the concept of “Serverless” event driven architectures.

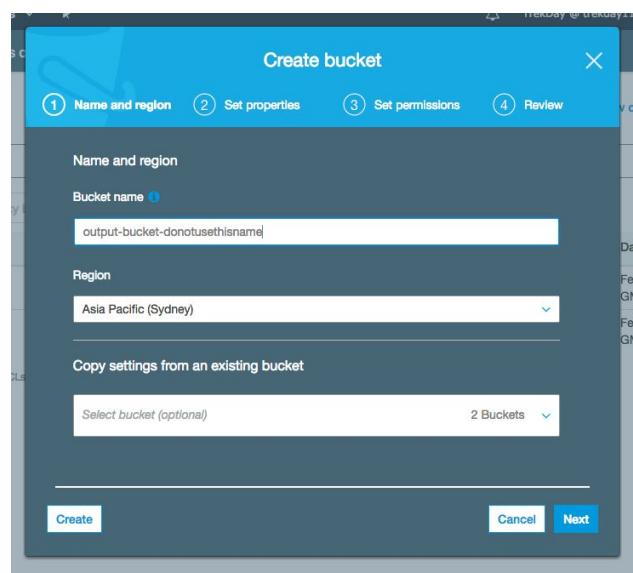
Take the following steps, one at a time. There is plenty of time to complete the session and ask any questions along the way.

Creating S3 Buckets

- For this workflow we will need a couple of S3 buckets
- Open the AWS console, make sure you are in Sydney, and navigate to S3



- Select Create bucket
- Enter a bucket name (this must be globally unique) don't use the same name as in the screenshot, and don't use the same as your neighbour!



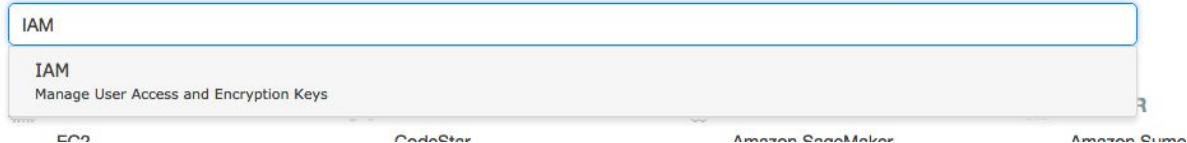
- Tip: Enter the name "input-bucket-" then add some pseudorandom characters by mashing the keyboard.

- The S3 console page is global, so you need to select a region to create the bucket. **Make sure this is Sydney.**
- Select "Create"
- Repeat this step and create a "output-bucket-xxxxxxxx"
- Make a note of the bucket names.

 input-bucket-donotusethisname	Not public *	Asia Pacific (Sydney)	Feb 25, 2018 11:46:33 AM GMT+1000
 output-bucket-donotusethisname	Not public *	Asia Pacific (Sydney)	Feb 25, 2018 11:46:50 AM GMT+1000

Creating an IAM Role

- Before we create our Lambda function, we need to create a security profile (called an IAM Role) that will give the function all the necessary permissions it needs to run.
- Navigate to IAM (Identity and Access Management)



- Select "Roles"



- Select "Create role"

Create role **Delete role**

Search

Role name ▾	Description
-------------	-------------

- Ensure that "AWS Service" is selected, then select "Lambda", and "Next: Permissions"



AWS service
EC2, Lambda and others



Another AWS account
Belonging to you or 3rd party



Web identity
Cognito or any OpenID provider



SAML
Saml 2.0 federation
Your corporate directory

Allows AWS services to perform actions on your behalf. [Learn more](#)

Choose the service that will use this role

API Gateway	Data Pipeline	ElasticLoadBalancing	MediaConvert	Service Catalog
Auto Scaling	DeepLens	Glue	OpsWorks	Step Functions
Batch	Directory Service	Greengrass	RDS	Storage Gateway
CloudFormation	DynamoDB	GuardDuty	Redshift	
CloudHSM	EC2	Inspector	Rekognition	
CloudWatch Events	EMR	IoT	S3	
CodeBuild	ElastiCache	Kinesis	SMS	
CodeDeploy	Elastic Beanstalk	Lambda	SNS	
Config	Elastic Container Service	Lex	SWF	
DMS	Elastic Transcoder	Machine Learning	SageMaker	

* Required [Cancel](#) [Next: Permissions](#)

- In the policy type search box, type "Lambda", and scroll to "AWSLambdaExecute"

- SELECT THE CHECKBOX next to the policy name (if you select the policy name itself, you will be sent elsewhere!)

Trust Permissions Review

Attach permissions policies

Choose one or more policies to attach to your new role.

[Create policy](#) [Refresh](#)

Policy name		Attachments	Description
<input type="checkbox"/>	AWSLambdaENIManagementAccess	0	Provides minimum permissions for a Lambda function to ma...
<input checked="" type="checkbox"/>	AWSLambdaExecute	0	Provides Put, Get access to S3 and full access to CloudWat...

Showing 2 results

- Select "Next Review"

- Enter a name for your role "my-lambda-role"

- Select "Create role"

Create role

1 2 3

Trust Permissions Review

Review

Provide the required information below and review this role before you create it.

Role name* my-lambda-role
Maximum 64 characters. Use alphanumeric and '+-,@-_'

Role description Allows Lambda functions to call AWS services on your behalf.
Maximum 1000 characters. Use alphanumeric and '+-,@-_'

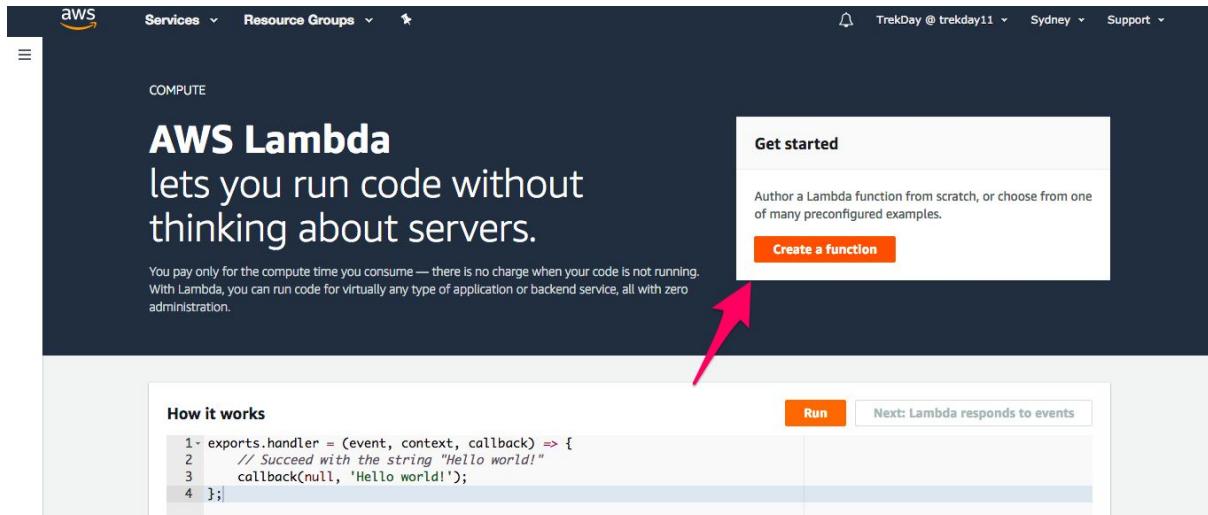
Trusted entities AWS service: lambda.amazonaws.com

Policies AWSLambdaExecute

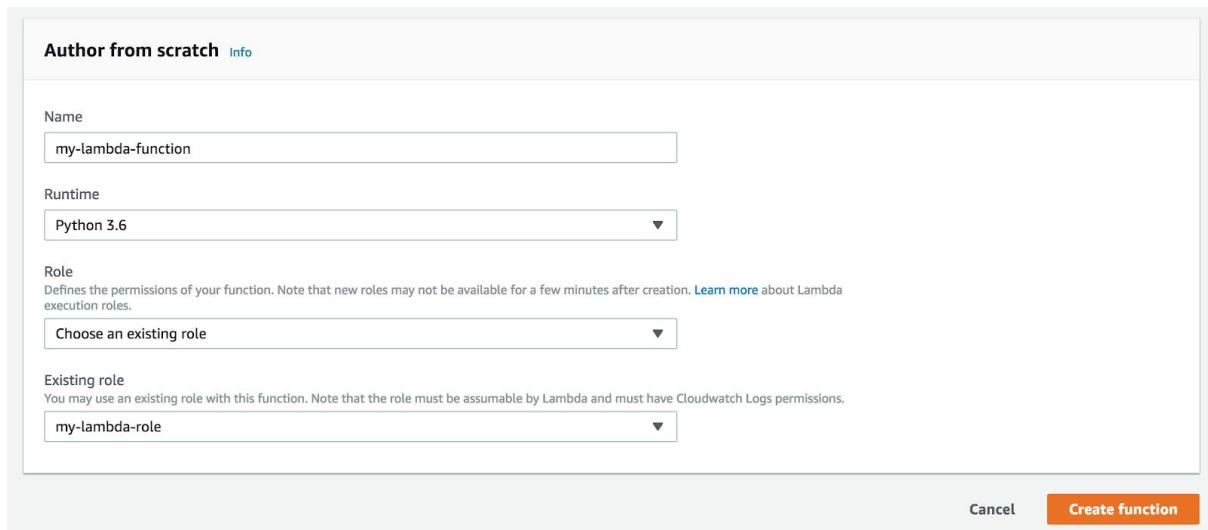
* Required Cancel Previous **Create role**

Create a Lambda Function

- Now lets create our Lambda function
- Navigate to Lambda
- Select "Create function"



- Make sure that "Author from scratch" is selected
- Enter a function name: "my-lambda-function"
- In the Runtime selection, select "Python 3.6" (*Notice the other runtimes you could use when writing your own functions.*)
- In Role, select "Choose an existing role" and then select in the "Existing role" selection to find "my-lambda-role"
- Select "Create function"



Author from scratch Info

Name
my-lambda-function

Runtime
Python 3.6

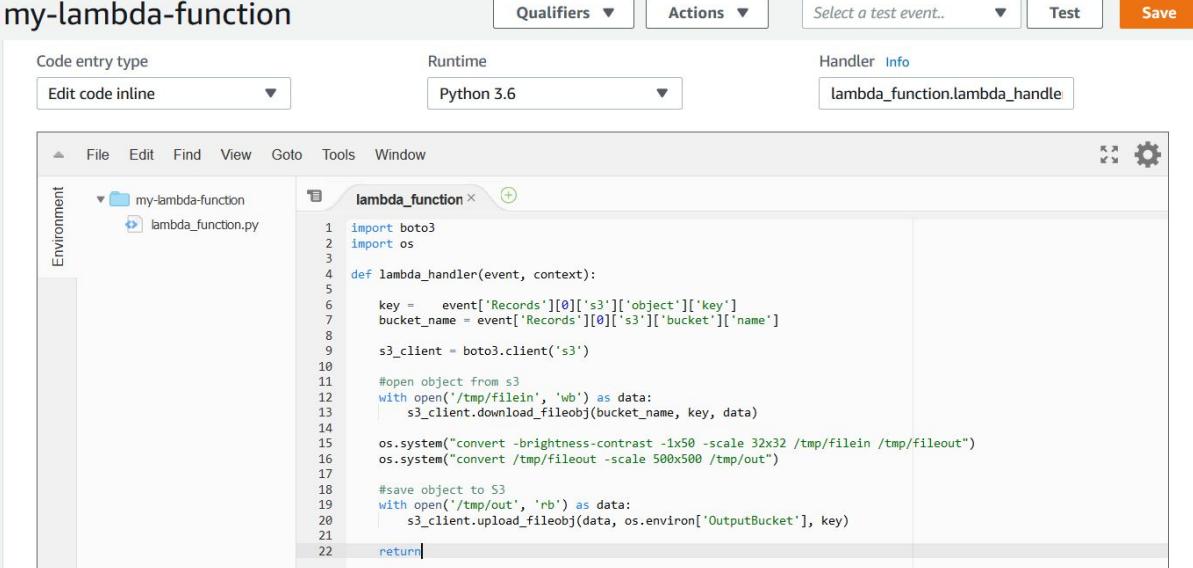
Role
Defines the permissions of your function. Note that new roles may not be available for a few minutes after creation. [Learn more](#) about Lambda execution roles.
Choose an existing role

Existing role
You may use an existing role with this function. Note that the role must be assumable by Lambda and must have Cloudwatch Logs permissions.
my-lambda-role

Cancel **Create function**

- You will now be taken to the main Lambda function page.
- Scroll down and find the code editor.

- From the resources with the session, locate 'lambda_handler.py' open up the file in a text editor. It's a pretty simple script. Copy and paste it into the Lambda code editor, replacing all the code that was there

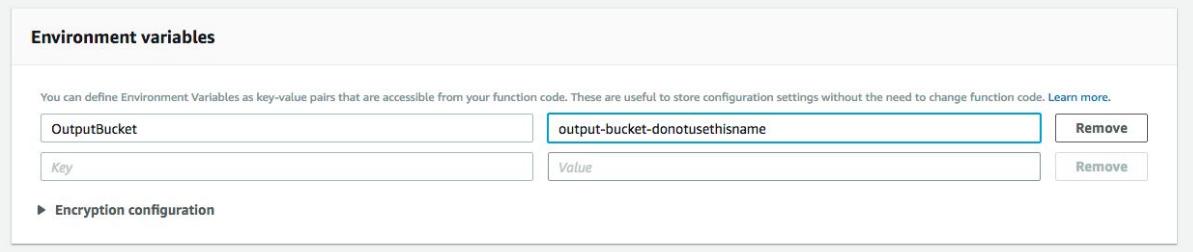


```

1 import boto3
2 import os
3
4 def lambda_handler(event, context):
5     key = event['Records'][0]['s3']['object']['key']
6     bucket_name = event['Records'][0]['s3']['bucket']['name']
7
8     s3_client = boto3.client('s3')
9
10    #open object from S3
11    with open('/tmp/filein', 'wb') as data:
12        s3_client.download_fileobj(bucket_name, key, data)
13
14    os.system("convert -brightness-contrast -1x50 -scale 32x32 /tmp/filein /tmp/fileout")
15    os.system("convert /tmp/fileout -scale 500x500 /tmp/out")
16
17    #save object to S3
18    with open('/tmp/out', 'rb') as data:
19        s3_client.upload_fileobj(data, os.environ['OutputBucket'], key)
20
21
22    return

```

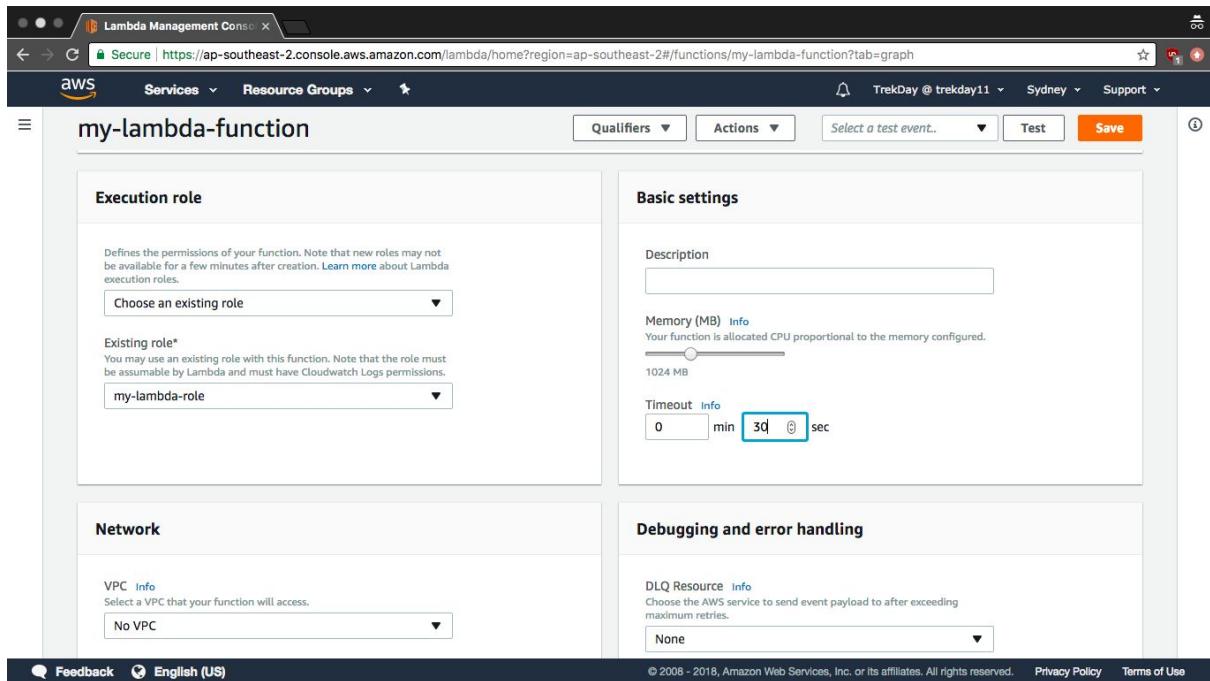
- In the "Environment variables" section enter the "Key" of "OutputBucket" and the name of YOUR output bucket "output-bucket-xxxxxx"



Key	Value	Remove
OutputBucket	output-bucket-donotusethisname	Remove
Key	Value	Remove

▶ Encryption configuration

- In the "Basic settings" section move the "Memory" slider up to "1024 MB" and the "Timeout" to "30" seconds.

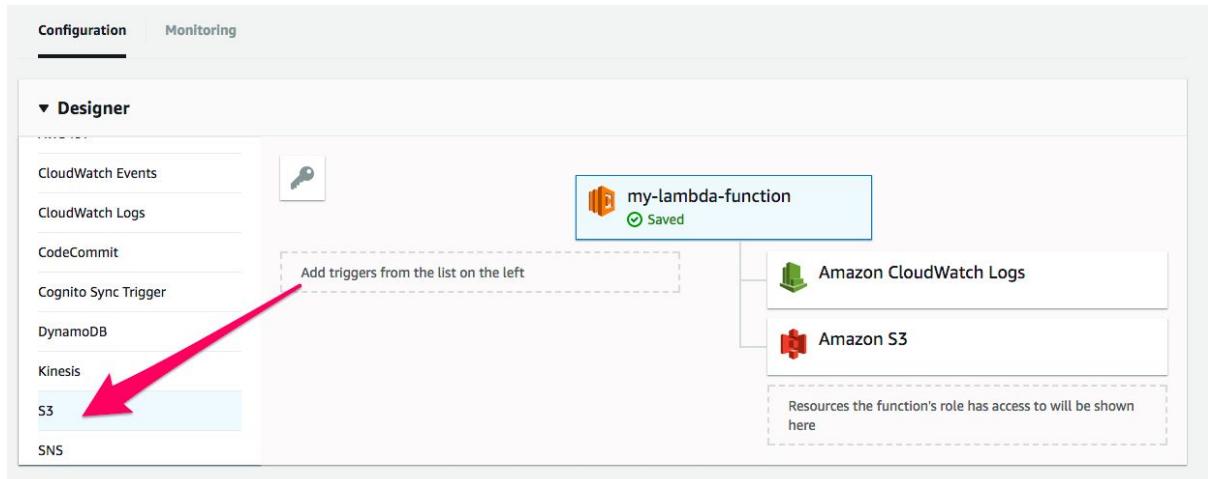


The screenshot shows the AWS Lambda Management Console for the function 'my-lambda-function'. The 'Basic settings' tab is active, displaying a memory of 1024 MB and a timeout of 30 seconds. The 'Execution role' tab shows 'my-lambda-role' selected. The 'Network' tab shows 'No VPC' selected. The 'Debugging and error handling' tab shows 'None' selected.

- Select "Save"

Adding A Trigger

- Scroll back up to the top of the page. In the "Designer" section, scroll under "Add triggers" until you see "S3" and select it.



The screenshot shows the AWS Lambda Designer interface. The left sidebar lists various trigger types: CloudWatch Events, CloudWatch Logs, CodeCommit, Cognito Sync Trigger, DynamoDB, Kinesis, S3, and SNS. An arrow points to the 'S3' option. The main workspace shows a box for 'my-lambda-function' with a 'Saved' status. To the right, there are sections for 'Amazon CloudWatch Logs' and 'Amazon S3', with a note indicating resources the function's role has access to will be shown here.

- You should now see a section called "Configure triggers"
- In the "Bucket" selection, find and select YOUR input bucket "input-bucket-xxxxxxx"
- Under "Event type" select "Object Created (All)"

- Ensure the "Enable trigger" checkbox is ticked and select "Add" (in the bottom right)

my-lambda-function

Please select the S3 bucket that serves as the event source. The bucket must be in the same region as the function.

input-bucket-donotusethisname

Event type
Select the events that you want to trigger the Lambda function. You can optionally set up a prefix or suffix for an event. However, for each bucket, individual events cannot have multiple configurations with overlapping prefixes or suffixes that could match the same object key.

Object Created (All)

Prefix
Enter an optional prefix to limit the notifications to objects with keys that start with matching characters.
e.g. images/

Suffix
Enter an optional suffix to limit the notifications to objects with keys that end with matching characters.
e.g. jpg

Lambda will add the necessary permissions for Amazon S3 to invoke your Lambda function from this trigger. [Learn more](#) about the Lambda permissions model.

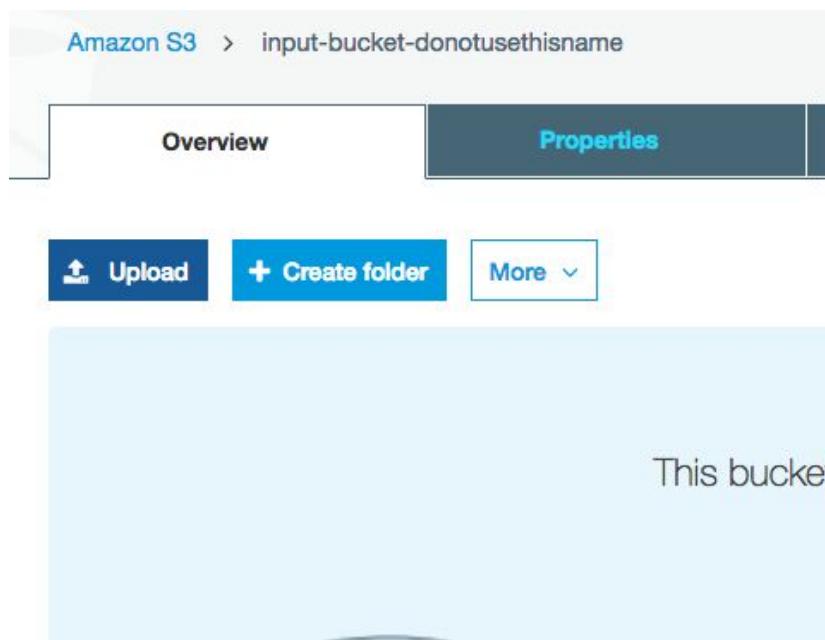
Enable trigger
Enable the trigger now, or create it in a disabled state for testing (recommended).

Cancel Add

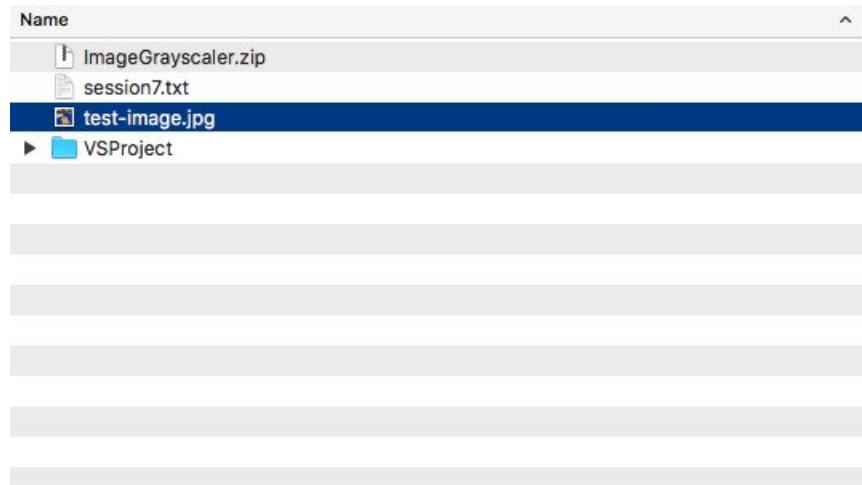
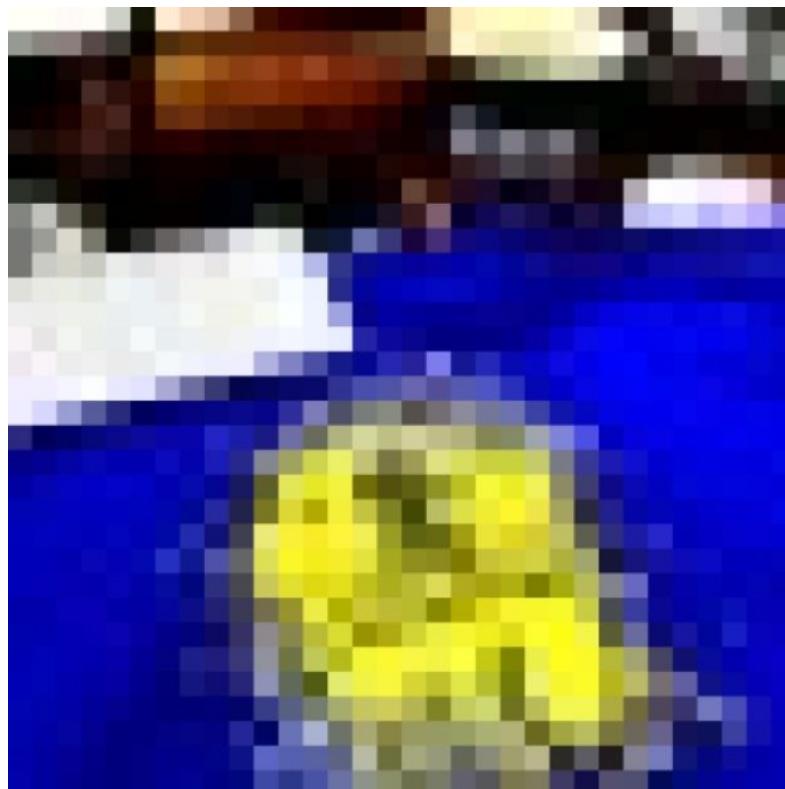
- Select "Save"

- Now it's time to test:

- Navigate to S3, and select YOUR 'input-bucket-xxxx'
- Select 'Upload'



- Within the resources for this lesson you will find a test image. Feel free to use a JPG image of your own to upload to the bucket.

**Result!**

Pixelated fun! :)

If you are running into issues...

- Check that your role has the appropriate rights
- Make sure you are uploading a .JPG file in your input-bucket
- Make sure that you have placed the OutputBucket Key and Value in the Environment Variables and not in Tags

Extra Challenge

While we wait for everyone to complete this session....

- Let's get AWS to send us an email when the image has been processed!
- We will use SNS

END