

SUMMARY:

This Lab covers Bootstrapping, CFN-INIT, Instance Roles, Parameter Store, Logging metrics with CW Agent

Contents

1. Bootstrapping wordpress directly and with CFN	1
2. CFN-INIT and CFN Creation Policies	3
3. Providing permissions and credentials to EC2 using Instance Roles'	7
4. Parameter Store	10
5. Logging and Metrics with CW Agent.....	11

1. Bootstrapping wordpress directly and with CFN

In this [DEMO] you will bootstrap two EC2 instances with wordpress and the cowsay login banner customisations.

The first, directly using User Data via the console UI, the second, using Cloudformation

TEXT

Command used to query User Data

```
curl http://169.254.169.254/latest/user-data
```

Bootstrapping script:

```
#!/bin/bash -xe

# Setpassword & DB Variables
DBName='a4lwordpress'
DBUser='a4lwordpress'
DBPassword='REPLACEME'
DBRootPassword='REPLACEME'

# System Updates
yum -y update
yum -y upgrade

# STEP 2 - Install system software - including Web and DB
yum install -y mariadb-server httpd wget cowsay
amazon-linux-extras install -y lamp-mariadb10.2-php7.2 php7.2

# STEP 3 - Web and DB Servers Online - and set to startup
systemctl enable httpd
systemctl enable mariadb
systemctl start httpd
systemctl start mariadb

# STEP 4 - Set Mariadb Root Password
mysqladmin -u root password $DBRootPassword

# STEP 5 - Install Wordpress
wget http://wordpress.org/latest.tar.gz -P /var/www/html
cd /var/www/html
tar -zxvf latest.tar.gz
cp -rvf wordpress/* .
```

```

rm -R wordpress
rm latest.tar.gz
# STEP 6 - Configure Wordpress
cp ./wp-config-sample.php ./wp-config.php
sed -i "s/'database_name_here'/'$DBName'/g" wp-config.php
sed -i "s/'username_here'/'$DBUser'/g" wp-config.php
sed -i "s/'password_here'/'$DBPassword'/g" wp-config.php
# Step 6a - permissions
usermod -a -G apache ec2-user
chown -R ec2-user:apache /var/www
chmod 2775 /var/www
find /var/www -type d -exec chmod 2775 {} \;
find /var/www -type f -exec chmod 0664 {} \;
# STEP 7 Create Wordpress DB
echo "CREATE DATABASE $DBName;" >> /tmp/db.setup
echo "CREATE USER '$DBUser'@'localhost' IDENTIFIED BY '$DBPassword';" >> /tmp/db.setup
echo "GRANT ALL ON $DBName.* TO '$DBUser'@'localhost';" >> /tmp/db.setup
echo "FLUSH PRIVILEGES;" >> /tmp/db.setup
mysql -u root --password=$DBRootPassword < /tmp/db.setup
sudo rm /tmp/db.setup
# STEP 8 COWSAY
echo "#!/bin/sh" > /etc/update-motd.d/40-cow
echo 'cowsay "Amazon Linux 2 AMI - Animals4Life"' >> /etc/update-motd.d/40-cow
chmod 755 /etc/update-motd.d/40-cow
rm /etc/update-motd.d/30-banner
update-motd

```

Creates EC2 instance with wordpress, check IP and verify site contents

The screenshot shows the AWS Management Console interface. On the left, there is a navigation menu with options like 'EC2 Dashboard', 'Events', 'Tags', 'Reports', 'Limits', 'INSTANCES', 'Instance Types', 'Launch Templates', 'Spot Requests', 'Savings Plans', 'Reserved Instances', 'Dedicated Hosts', 'Scheduled Instances', 'Capacity Reservations', 'IMAGES', and 'AMIs'. The main area displays the details of an EC2 instance named 'A4L-ManualWordpress' with Instance ID 'i-047f5aa825fb03070'. The instance is in a 'running' state, located in the 'us-east-1a' Availability Zone, and has a 't2.micro' instance type. The public DNS is 'ec2-18-209-161-171.compute-1.amazonaws.com'. The public IPv4 address is '18.209.161.171', which is highlighted with a green box. Other details include the private DNS 'ip-10-16-57-35.ec2.internal', private IPs '10.16.57.35', and the VPC ID 'vpc-07b889beb15994990 (a4l-)'. The bottom of the console shows the footer with 'Feedback', 'English (US)', and copyright information for Amazon Web Services, Inc. or its affiliates.

WordPress » Installation

18.209.161.171/wp-admin/install.php

Welcome

Welcome to the famous five-minute WordPress installation process! Just fill in the information below and you'll be on your way to using the most extendable and powerful personal publishing platform in the world.

Information needed

Please provide the following information. Don't worry, you can always change these settings later.

Site Title

Username

Username can have only alphanumeric characters, spaces, underscores, hyphens, periods, and the @ symbol.

Password [Hide](#)

Strong

Important: You will need this password to log in. Please store it in a secure location.

Your Email

Double-check your email address before continuing.

Search Engine Visibility ☐ Discourage search engines from indexing this site

It is up to search engines to honor this request.

[Install WordPress](#)

2. CFN-INIT and CFN Creation Policies

In this [DEMO] Lesson you will get to experience how CFN-INIT, CFN-SIGNAL and CloudFormation Creation policies can further enhance the EC2 Bootstrapping process.

CFN Init – should create same results as the bootstrapping script:

```
Description: A4L CFN-INIT Wordpress Template
Requires A4L VPC Template to run
Parameters:
LatestAmiId:
  Description: AMI for Instance (default is latest AmaLinux2)
  Type: 'AWS::SSM::Parameter::Value<AWS::EC2::Image::Id>'
  Default: '/aws/service/ami-amazon-linux-latest/amzn2-ami-hvm-x86_64-gp2'
KeyName:
  Type: AWS::EC2::KeyPair::KeyName
```

```
    Description: "Name of an existing SSH Keypair to access the instance"
DBName:
  AllowedPattern: '[a-zA-Z][a-zA-Z0-9]*'
  ConstraintDescription: must begin with a letter and contain only alphanumeric
    characters.
  Default: a4lwordpress
  Description: The WordPress database name
  MaxLength: '64'
  MinLength: '1'
  Type: String
DBPassword:
  AllowedPattern: '[a-zA-Z0-9]*'
  ConstraintDescription: must contain only alphanumeric characters.
  Description: The WordPress database admin account password
  MaxLength: '41'
  MinLength: '8'
  NoEcho: 'true'
  Type: String
DBRootPassword:
  AllowedPattern: '[a-zA-Z0-9]*'
  ConstraintDescription: must contain only alphanumeric characters.
  Description: MySQL root password
  MaxLength: '41'
  MinLength: '8'
  NoEcho: 'true'
  Type: String
DBUser:
  AllowedPattern: '[a-zA-Z][a-zA-Z0-9]*'
  ConstraintDescription: must begin with a letter and contain only alphanumeric
    characters.
  Description: The WordPress database admin account username
  Default: a4lwordpress
  MaxLength: '16'
  MinLength: '1'
  NoEcho: 'true'
  Type: String
Resources:
  EC2Instance:
    Type: AWS::EC2::Instance
    CreationPolicy:
      ResourceSignal:
        Timeout: PT15M
    Metadata:
      AWS::CloudFormation::Init:
        configSets:
          wordpress_install:
            - install_cfn
            - software_install
            - configure_instance
            - install_wordpress
            - configure_wordpress
          install_cfn:
```

```
files:
  /etc/cfn/cfn-hup.conf:
    content: !Sub |
      [main]
      stack= ${AWS::StackId}
      region=${AWS::Region}
    group: root
    mode: '000400'
    owner: root
  /etc/cfn/hooks.d/cfn-auto-reloader.conf:
    content: !Sub |
      [cfn-auto-reloader-hook]
      triggers=post.update
      path=Resources.EC2Instance.Metadata.AWS::CloudFormation::Init
      action=/opt/aws/bin/cfn-init -v --stack ${AWS::StackName} --
resource EC2Instance --configsets wordpress_install --region ${AWS::Region}
    group: root
    mode: '000400'
    owner: root
services:
  sysvinit:
    cfn-hup:
      enabled: true
      ensureRunning: true
      files:
        - /etc/cfn/cfn-hup.conf
        - /etc/cfn/hooks.d/cfn-auto-reloader.conf
software_install:
  packages:
    yum:
      httpd: []
      mariadb-server: []
      wget: []
      cowsay: []
  commands:
    0_extra_installs_php72_lampmariadb:
      command: amazon-linux-extras install -y lamp-mariadb10.2-php7.2 php7.2
services:
  sysvinit:
    httpd:
      enabled: true
      ensureRunning: true
    mariadb:
      enabled: true
      ensureRunning: true
configure_instance:
  files:
    /etc/update-motd.d/40-cow:
      content: !Sub |
        #!/bin/sh
        cowsay "Amazon Linux 2 AMI - Animals4Life"
      group: root
```

```

    mode: '000755'
    owner: root
  commands:
    01_set_mysql_root_password:
      command: !Sub |
        mysqladmin -u root password '${DBRootPassword}'
      test: !Sub |
        ${mysql ${DBName} -u root --
password='${DBRootPassword}' >/dev/null 2>&1 </dev/null); (( $? != 0 ))
    02_remove_original_banner:
      command: rm /etc/update-motd.d/30-banner
    03_updatemotd:
      command: update-motd
  install_wordpress:
    sources:
      /var/www/html: http://wordpress.org/latest.tar.gz
    files:
      /tmp/create-wp-config:
        content: !Sub |
          #!/bin/bash -xe
          cp /var/www/html/wp-config-sample.php /var/www/html/wp-config.php
          sed -i "s/'database_name_here'/'${DBName}'/g" wp-config.php
          sed -i "s/'username_here'/'${DBUser}'/g" wp-config.php
          sed -i "s/'password_here'/'${DBPassword}'/g" wp-
config.php
        group: root
        mode: '000500'
        owner: root
      /tmp/db.setup:
        content: !Sub |
          CREATE DATABASE ${DBName};
          CREATE USER '${DBUser}'@'localhost' IDENTIFIED BY '${DBPassword}';
          GRANT ALL ON ${DBName}.* TO '${DBUser}'@'localhost';
          FLUSH PRIVILEGES;
        group: root
        mode: '000400'
        owner: root
  configure_wordpress:
    files:
      /tmp/permissionsfix:
        content: !Sub |
          usermod -a -G apache ec2-user
          chown -R ec2-user:apache /var/www
          chmod 2775 /var/www
          find /var/www -type d -exec chmod 2775 {} \;
          find /var/www -type f -exec chmod 0664 {} \;
        group: root
        mode: '000500'
        owner: root
    commands:
      01_create_database:
        command: !Sub |

```

```
mysql -u root --password='${DBRootPassword}' < /tmp/db.setup
test: !Sub |
    ${mysql ${DBName} -u root --
password='${DBRootPassword}' >/dev/null 2>&1 </dev/null); (( $? !=0))
02_move_wordpress:
    command: !Sub |
        cp -rvf /var/www/html/wordpress/* /var/www/html/
03_tidyup:
    command: !Sub |
        rm -R /var/www/html/wordpress
04_configure_wordpress:
    command: /tmp/create-wp-config
    cwd: /var/www/html
04_fix_permissions:
    command: /tmp/permissionsfix
Properties:
    KeyName: !Ref KeyName
    InstanceType: "t2.micro"
    ImageId: !Ref LatestAmiId
    SubnetId: !ImportValue a4l-vpc1-subnet-weba
    SecurityGroupIds:
        - !ImportValue a4l-vpc1-default-instance-sg
    Tags:
        - Key: Name
          Value: A4L-Wordpress
    UserData:
        Fn::Base64: !Sub |
            #!/bin/bash -xe
            yum -y update
            /opt/aws/bin/cfn-init -v --stack ${AWS::StackId} --resource EC2Instance --
configsets wordpress_install --region ${AWS::Region}
            /opt/aws/bin/cfn-signal -e $? --stack ${AWS::StackId} --resource EC2Instance --
region ${AWS::Region}
```

3. Providing permissions and credentials to EC2 using Instance Roles'

In this [DEMO] Lesson you will create an EC2 Instance Role, apply it to an EC2 instance and learn how to interact with the credentials this generates within the EC2 instance metadata.

1-Click Deployment

Lesson Commands


Credential Precedence


Create instance, create instance role in IAM, attach instance role to the instance:


Create role


1 2 3 4

Select type of trusted entity

**AWS service**
EC2, Lambda and others

**Another AWS account**
Belonging to you or 3rd party

**Web identity**
Cognito or any OpenID provider

**SAML 2.0 federation**
Your corporate directory

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

Choose a use case

Common use cases

EC2

Allows EC2 instances to call AWS services on your behalf.

Lambda

Allows Lambda functions to call AWS services on your behalf.

▼ Attach permissions policies

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








Create policy



Filter policies ▼

Q s3


Showing 9 results

	Policy name ▼	Used as
<input type="checkbox"/>	 AmazonDMSRedshiftS3Role	None
<input type="checkbox"/>	 AmazonS3FullAccess	None
<input type="checkbox"/>	 AmazonS3OutpostsFullAccess	None
<input type="checkbox"/>	 AmazonS3OutpostsReadOnlyAccess	None
<input checked="" type="checkbox"/>	 AmazonS3ReadOnlyAccess	None
<input type="checkbox"/>	 IVSRecordToS3	None
<input type="checkbox"/>	 QuickSightAccessForS3StorageManagementAnalyticsReadOnly	None
<input type="checkbox"/>	s3crr_for_animals4lifeuseast1_to_animals4lifeuswest1_80cad7	None

► Set permissions boundary

Review

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



 **Role name***

Use alphanumeric and '+=, @-_' characters. Maximum 64 characters.

Role description

Maximum 1000 characters. Use alphanumeric and '+=, @-_' characters.

Trusted entities AWS service: ec2.amazonaws.com

Policies  [AmazonS3ReadOnlyAccess](#) 

Permissions boundary Permissions boundary is not set

EC2 > Instances > i-049b107605a791e55 > Modify IAM role

Modify IAM role [Info](#)

Attach an IAM role to your instance.

Instance ID

 i-049b107605a791e55 (A4L-PublicEC2)

IAM role

Select an IAM role to attach to your instance or create a new role if you haven't created any. The role you select replaces any roles that are currently attached to your instance.

Choose IAM role

Q |

No IAM Role

Choose this option to detach an IAM role

A4LInstanceRole

arn:aws:iam::945690336440:instance-profile/A4LInstanceRole



Create new IAM role 

The instance will be removed. Are you

Cancel

Save

The screenshot displays the AWS Management Console interface for an EC2 instance. At the top, the 'Instances (1/1)' header is visible, along with a search bar and a filter for 'Instance state: running'. Below this, a table lists the instance details:

Name	Instance ID	Instance state	Instance type
A4L-PublicEC2	i-049b107605a791e55	Running	t2.micro

Below the table, the 'Instance: i-049b107605a791e55 (A4L-PublicEC2)' details are shown. The 'Security' tab is selected, and the 'IAM Role' section is highlighted with a green box, showing the role 'A4LInstanceRole'. Other details include the Owner ID '945690336440' and the Security group 'sg-03697ed2db43150d0 (IAMROLEDEMO-InstanceSecurityGroup-1QFOUAXNRVW78)'.

4. Parameter Store

In this [DEMO] you get a chance to create some Parameters in the Parameter Store and interact with them via the command line - using individual parameter operations and accessing via paths.

Create parameters by path

Name

Description — *Optional*

Tier

Parameter Store offers standard and advanced parameters.

☒ **Standard**

Limit of 10,000 parameters. Parameter value size up to 4 KB. Parameter policies are not available. No additional charge.

☐ **Advanced**

Can create more than 10,000 parameters. Parameter value size up to 8 KB. Parameter policies are available. Charges apply

Type

☐ **String**

Any string value.

☐ **StringList**

Separate strings using commas.

☒ **SecureString**

Encrypt sensitive data using KMS keys from your account or another account.

KMS key source

☒ **My current account**

Use the default KMS key for this account or specify a customer-managed key for this account. [Learn more](#)

☐ **Another account**

Use a KMS key from another account [Learn more](#)

KMS Key ID

AWS Systems Manager > Parameter Store

Parameters

Settings

Parameters



<input type="checkbox"/>	Name	Tier	Type
<input type="checkbox"/>	/my-cat-app/dbpassword	Standard	SecureString
<input type="checkbox"/>	/my-cat-app/dbstring	Standard	String
<input type="checkbox"/>	/my-cat-app/dbuser	Standard	String
<input type="checkbox"/>	/my-dog-app/dbstring	Standard	String
<input type="checkbox"/>	/rate-my-lizard/dbstring	Standard	String

5. Logging and Metrics with CW Agent

In this [DEMO] lesson you will download and install the CloudWatch Agent and configure it to capture 3 log files from an EC2 instance

```
/var/log/secure
```

```
/var/log/httpd/access_log
```

```
/var/log/httpd/error_log
```


You will also configure an instance role allowing the agent to store the above config into parameter store AND allow the agent to inject the logging and metric data into CW and CW Logs.

1-Click Deployment

Lesson Commands

Be sure to acknowledge cloudwatch on:

Capabilities

 **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.**

Cancel

Create change set

Create stack

Install agent on instance:

```

< Amazon Linux 2 AMI - Animals4Life >
-----
      ^ ^
      (oo)\_____)
      (__)|       )\/\
          ||----w |
          ||     ||

[ec2-user@ip-10-16-49-221 ~]$ wget https://s3.amazonaws.com/amazoncloudwatch-agent/amazon_linux/amd64/latest/amazon-cloudwatch-agent.rpm
--2021-01-09 01:05:50-- https://s3.amazonaws.com/amazoncloudwatch-agent/amazon_linux/amd64/latest/amazon-cloudwatch-agent.rpm
Resolving s3.amazonaws.com (s3.amazonaws.com)... 52.217.46.246
Connecting to s3.amazonaws.com (s3.amazonaws.com)|52.217.46.246|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 38761649 (37M) [application/octet-stream]
Saving to: 'amazon-cloudwatch-agent.rpm'

100%[=====] 38,761,649 99.2MB/s  in 0.4s

```

Create and attach IAM cloudwatch role for instance

Create role

1 2 3 4

Review

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

Role name*



Use alphanumeric and '+=, @-_' characters. Maximum 64 characters.

Role description

Maximum 1000 characters. Use alphanumeric and '+=, @-_' characters.

Trusted entities AWS service: ec2.amazonaws.com

Policies

-  CloudWatchAgentServerPolicy [↗](#)
-  AmazonSSMFullAccess [↗](#)

Permissions boundary Permissions boundary is not set

No tags were added.

* Required

Cancel

Previous

Create role

EC2 > Instances > i-006374942a3afe62a > Modify IAM role

Modify IAM role [Info](#)

Attach an IAM role to your instance.

Instance ID

 i-006374942a3afe62a (A4L-Wordpress)

IAM role

Select an IAM role to attach to your instance or create a new role if you haven't created any. The role you select replaces any roles that are currently attached to your instance.

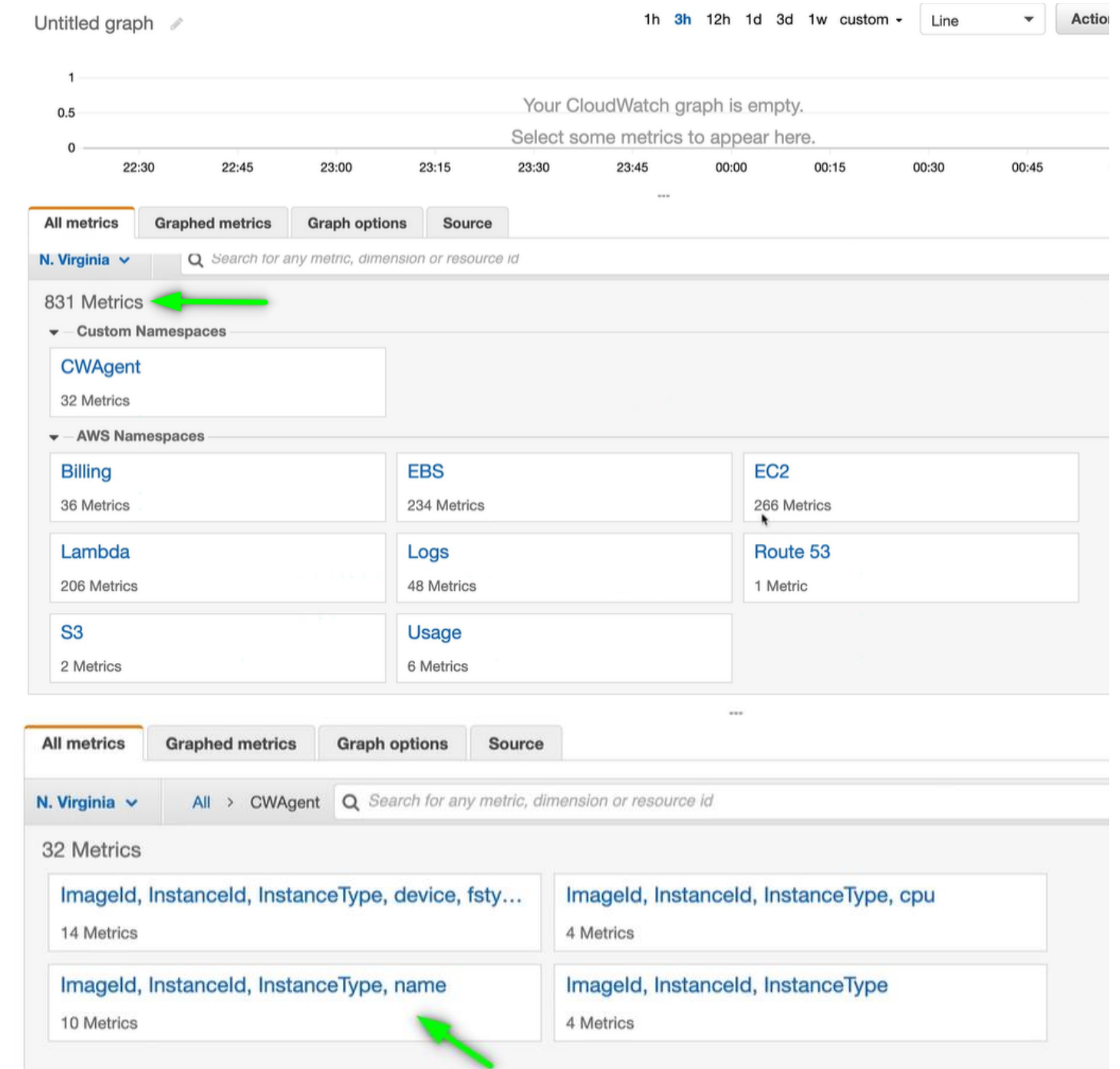


Create new IAM role [↗](#)

Cancel

Save

Can interact with wordpress instance and now have access to metrics:



All metrics

Graphed metrics

Graph options

Source

N. Virginia

All > CWAgent > ImageId, InstanceId, InstanceType, name

Search for any metric, dimension or resource id

Graph

	Instance Name (10)	ImageId	InstanceId	InstanceType	name	Metric Name
<input type="checkbox"/>	A4L-Wordpress	ami-0be2609ba883822ec	i-006374942a3afe62a	t2.micro	xvda1	diskio_read_bytes
<input type="checkbox"/>	A4L-Wordpress	ami-0be2609ba883822ec	i-006374942a3afe62a	t2.micro	xvda1	diskio_io_time
<input type="checkbox"/>	A4L-Wordpress	ami-0be2609ba883822ec	i-006374942a3afe62a	t2.micro	xvda1	diskio_reads
<input type="checkbox"/>	A4L-Wordpress	ami-0be2609ba883822ec	i-006374942a3afe62a	t2.micro	xvda1	diskio_writes
<input type="checkbox"/>	A4L-Wordpress	ami-0be2609ba883822ec	i-006374942a3afe62a	t2.micro	xvda1	diskio_write_bytes
<input checked="" type="checkbox"/>	A4L-Wordpress	ami-0be2609ba883822ec	i-006374942a3afe62a	t2.micro	xvda	diskio_writes
<input type="checkbox"/>	A4L-Wordpress	ami-0be2609ba883822ec	i-006374942a3afe62a	t2.micro	xvda	diskio_write_bytes
<input type="checkbox"/>	A4L-Wordpress	ami-0be2609ba883822ec	i-006374942a3afe62a	t2.micro	xvda	diskio_reads
<input type="checkbox"/>	A4L-Wordpress	ami-0be2609ba883822ec	i-006374942a3afe62a	t2.micro	xvda	diskio_read_bytes
<input type="checkbox"/>	A4L-Wordpress	ami-0be2609ba883822ec	i-006374942a3afe62a	t2.micro	xvda	diskio_io_time