

# Microcosm on AWS

---

## Table of Contents

---

1. [Introduction](#)
2. [Login](#)
3. [Deploy Microcosm using Amazon ECS](#)
  - i. [Clusters](#)
  - ii. [Task Definitions](#)
    - a. [Microcosm Task Definitions](#)
  - iii. [Services](#)
  - iv. [Setting up Jenkins for DevOps Pipeline \(Microcosm\)](#)
  - v. [Deploying test code in DevOps Pipeline](#)
    - a. [Public Git Hub](#)
    - b. [Gitlab Setup](#)
    - c. [Create Maven Deployment in Jenkins](#)
    - d. [Sonarqube Setup](#)
    - e. [Add Sonatype Nexus build step](#)
    - f. [AWS EC2 Based Code Deploy](#)
      - a. [Create IAM Role](#)
      - b. [Create EC2 Artifacts](#)
        - a. [Amazon Machine Image](#)
        - b. [Load Balancer](#)
      - c. [Create S3 Bucket](#)
      - d. [Create Codedeploy Application](#)
      - e. [Create Secret Key](#)
      - f. [Modify Jenkins Build](#)
      - g. [Build and Deploy](#)

[g. AWS ECS Based Code Deploy](#)

[4. Appendix A](#)

- [i. Billing](#)
- [ii. Add Sonatype Nexus build step](#)
- [iii. Creating Identity and Access Management \(IAM\) Users](#)
- [iv. Create Key Pair](#)
- [v. Creating a Role](#)
- [vi. Requesting Changes to Default AWS Limits](#)
- [vii. Create VPCs and user Accounts using CloudFormation Templates](#)

[5. Appendix B](#)

- [i. AWS Command Line Interface \(CLI\) and Setting up Elastic Container Registry \(ECR\)](#)
- [ii. ECS Get Started Wizard
  - \[a. Using Get Started Wizard\]\(#\)
  - \[b. Everything Created By Get Started Wizard\]\(#\)](#)
- [iii. Deleting an ECS Service and Associated Tasks](#)

[6. Appendix C](#)

- [i. Stand Alone Microcosm Template](#)

## Introduction

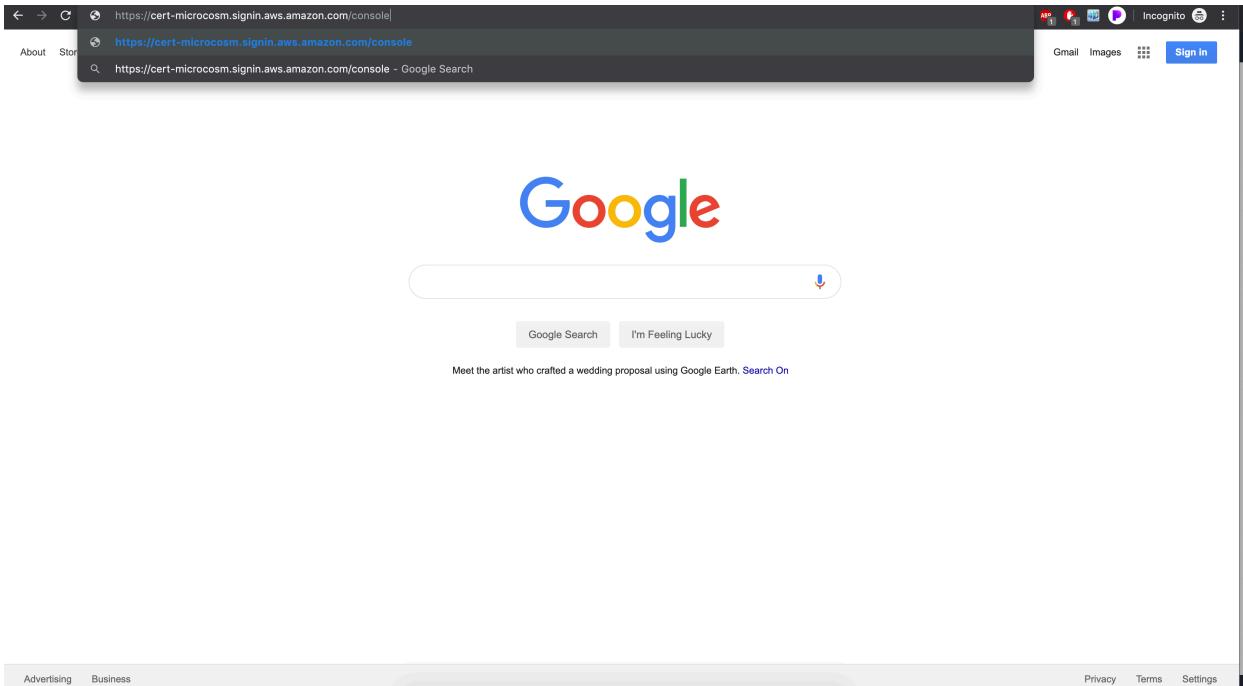
---

This version of Microcosm is intended to allow the user to stand up a complete DevSecOps pipeline comprised of serverless Docker containers using Amazon Web Service's (AWS) Elastic Container Service (ECS) Fargate instances. Instructions included below will cover the manual creation of all necessary AWS resources required to stand up the Microcosm DevSecOps pipeline, as well as two AWS CloudFormation template scripts that allow the automatic creation of all AWS resources and the pipeline using Infrastructure as Code (IaC). The CloudFormation template scripts are the recommended way to stand up a pipeline due to the fact that all resources in AWS are created within 10 minutes, and minimal prior AWS knowledge is required.

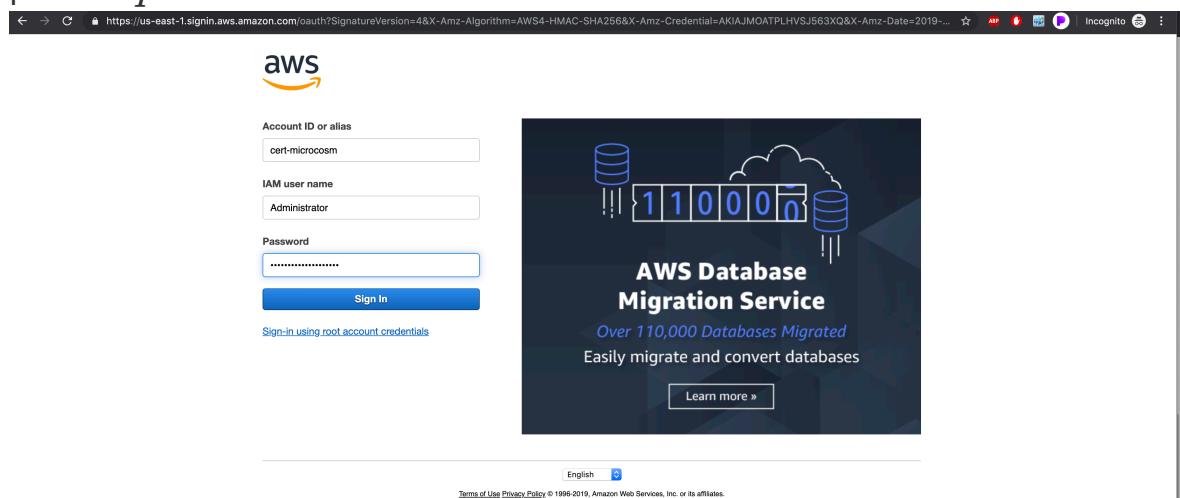
[Return to Table of Contents](#)

# Login

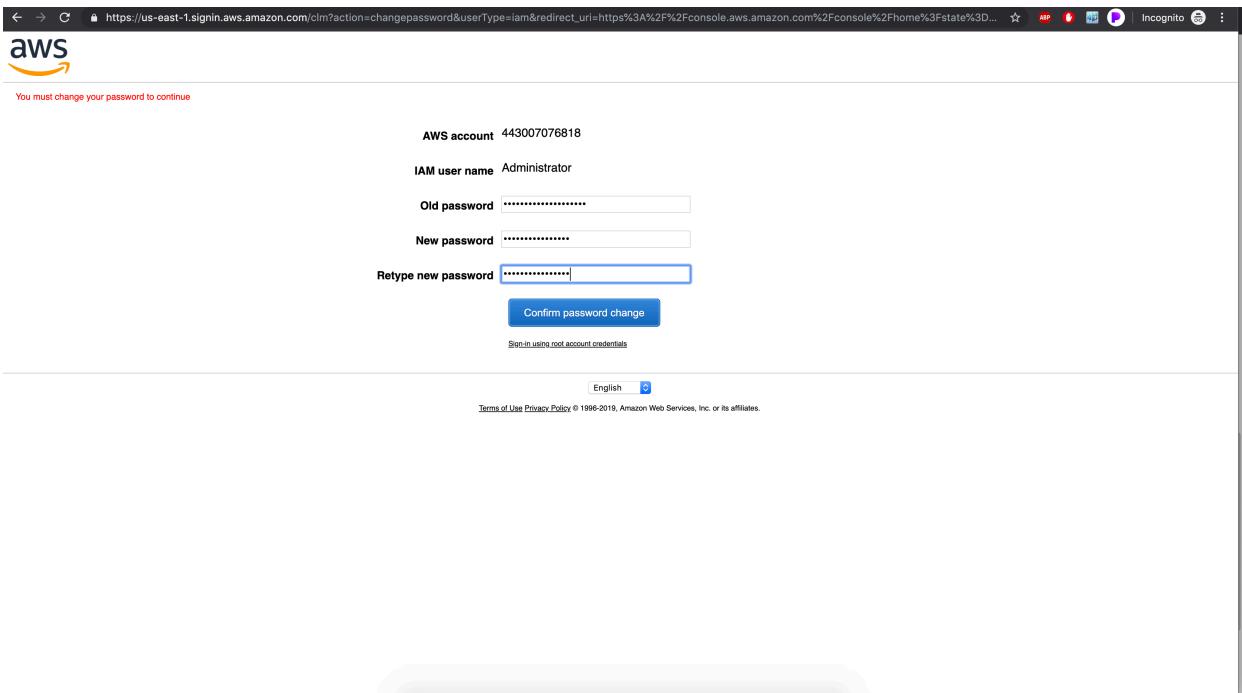
- Navigate to the URL received (eg <https://cert-microcosm.signin.aws.amazon.com/console> )



- Enter the credentials received
  - User:[stackname]-StudentUser[number]-[hash]
  - pwd: bzqUrFLxw9HFtB-49eRtf!



- Change your password to something you will remember



- After logging in and changing your password you be brought to the AWS Console, which is the nerve center of AWS. From here you can access any and all AWS services.

AWS Management Console

**AWS services**

**Find Services**  
You can enter names, keywords or acronyms.  
Example: Relational Database Service, database, RDS

All services

**Build a solution**  
Get started with simple wizards and automated workflows.

<b>Launch a virtual machine</b> With EC2 2-3 minutes 	<b>Build a web app</b> With Elastic Beanstalk 6 minutes 	<b>Build using virtual servers</b> With Lightsail 1-2 minutes 
<b>Connect an IoT device</b> With AWS IoT 5 minutes 	<b>Start a development project</b> With CodeStar 5 minutes 	<b>Register a domain</b> With Route 53 3 minutes 

**Access resources on the go**  
Access the Management Console using the AWS Console Mobile App. [Learn more](#)

**Explore AWS**

**Amazon Redshift**  
Fast, simple, cost-effective data warehouse that can extend queries to your data lake. [Learn more](#)

**Run Serverless Containers with AWS Fargate**  
AWS Fargate runs and scales your containers without having to manage servers or clusters. [Learn more](#)

**Scalable, Durable, Secure Backup & Restore with Amazon S3**  
Discover how customers are building backup & restore solutions on AWS that save money. [Learn more](#)

**AWS Marketplace**  
Find, buy, and deploy popular software products that run on AWS. [Learn more](#)

[Return to Table of Contents](#)

## Deploy Microcosm using Amazon ECS

Deploying Docker Containers in AWS is natively supported by Amazon's elastic container service. In ECS, the task definitions take the place of the docker-compose.yml file and contain the same information. These task definitions are instantiated as Services - which are grouping of instances of the task definitions. Once a service is started and the related task launches successfully, ECS manages the load balancing (if selected), failure recovery, addressing and other features. For information on the quick start wizard, see Appendix B.

## Clusters

Clusters are a grouping construct within ECS that enable Docker images to be instantiated as containers. The default cluster is created using the get started wizard and will create a VPC, Security groups, networking policies and more behind the scenes - enabling access to and for the container instances (also known as tasks). The following assumes that all enabling services are in place and only the cluster has yet to be created.

- Select ECS from the default AWS Console or the drop down services menu
- Open the cluster console and select create cluster

The screenshot shows the AWS ECS Clusters page. The left sidebar has 'Clusters' selected under 'Amazon ECS'. The main content area shows a single cluster named 'microcosm'. The table below provides details: Cluster name: microcosm, Services: 5, Running tasks: 6, Pending tasks: 0, Container instances: 0. The page is last updated on April 18, 2019 at 9:49:29 AM.

Cluster name	Services	Running tasks	Pending tasks	Container instances
microcosm	5	6	0	0

- We are going to be using AWS Fargate, so leave the default selection and click Next Step

The screenshot shows the 'Create Cluster' wizard on the AWS console. The current step is 'Step 1: Select cluster template'. There are three options available:

- Networking only**: Resources to be created: Cluster, VPC (optional), Subnets (optional). This option is highlighted and labeled 'Powered by AWS Fargate'.
- EC2 Linux + Networking**: Resources to be created: Cluster, VPC, Subnets. Auto Scaling group with Linux AMI.
- EC2 Windows + Networking**: Resources to be created: Cluster, VPC, Subnets. Auto Scaling group with Windows AMI.

At the bottom of the screen, there are buttons for '\*Required', 'Cancel', and 'Next step'.

- Enter a Cluster Name and select Create. Select create VPC. Once the cluster is successfully created, take a screenshot or write down the VPC, subnet, and security group ID's.
- Next, click on the link to your newly created security group and add the following inbound rules:
  - Type: HTTP | Protocol: TCP | Port Range: 80 | Source: 0.0.0.0/0
  - Type: customtcp | Protocol: TCP | Port Range: 9000 | Source: 0.0.0.0/0
  - Type: customtcp | Protocol: TCP | Port Range: 8080 | Source: 0.0.0.0/0
  - Type: SSH | Protocol: TCP | Port Range: 22 | Source: 0.0.0.0/0
  - Type: customtcp | Protocol: TCP | Port Range: 8081 | Source: 0.0.0.0/0
  - Type: HTTPS | Protocol: TCP | Port Range: 443 | Source: 0.0.0.0/0

Create Cluster

Configure cluster

Step 1: Select cluster template

Step 2: Configure cluster

Cluster name\* StudentCluster

Networking

Create a new VPC for your cluster to use. A VPC is an isolated portion of the AWS Cloud populated by AWS objects, such as Fargate tasks.

Create VPC  Create a new VPC for this cluster

Tags

Key Value

Add key Add value

\*Required Cancel Previous Create

- Once the cluster is created successfully select View Cluster

Launch status

Your container instances are launching, and it may take a few minutes until they are in the running state and ready to access. Usage hours on your new container instances start immediately and continue to accrue until you stop or terminate them.

Back View Cluster

ECS status - 1 of 1 complete StudentCluster

ECS cluster  
ECS Cluster StudentCluster successfully created

- Explore the cluster details as desired. In the next steps we will be using this cluster to instantiate services based on our Task definitions.

- Select Clusters from the left hand menu to see a high level view of all clusters available

## Task Definitions

Task Definitions take the place of elements within a docker-compose.yml file.

### Microcosm Task Definitions

Use the following table to create the task definitions necessary to stand up the Microcosm DevOps Pipeline **Important: for the Hubot task, replace the bracketed portion with the hubot key when generated**

Task/Service Name	Container Name	Image	Ports	Volumes
jenkins	jenkins	h1kkan/jenkins-docker:lts	8080	jenkins_home
sonarqube	sonarqube	sonarqube:lts	9000	sonarqube_config
				sonarqube_certs
				sonarqube_extensions
				sonarqube_plugins
gitlab	gitlab	gitlab/gitlab-ce	443	gitlab-config
			80	gitlab-logs
				gitlab-data
owaspZAP	owaspzap	owasp/zap2docker-stable	8080	
			8090	
nexus	sonatype_nexus	sonatype/nexus	8081	nexus-data

[Return to Table of Contents](#)

## Define a task

Repeat the Following for each desired task definition

- To define a new task (aka docker container) navigate to the ECS Console and select Task Definitions from the left menu, and press Create new Task Definition

The screenshot shows the AWS ECS Task Definitions page. On the left, there's a sidebar with links for Amazon ECS Clusters, Task Definitions, Amazon EKS Clusters, Amazon ECR Repositories, AWS Marketplace Discover software, and Subscriptions. The main area is titled "Task Definitions" and contains a table of existing task definitions. The table has two columns: "Task Definition" and "Latest revision status". The entries are:

Task Definition	Latest revision status
first-run-task-definition	ACTIVE
gitlab	ACTIVE
hubot-slack	ACTIVE
jenkins	ACTIVE
sonarqube	ACTIVE
zap2docker	ACTIVE

At the bottom, there are buttons for "Create new Task Definition", "Create new revision", and "Actions". A note says "Last updated on April 12, 2019 2:19:48 PM (0m ago)".

- Select the Fargate Type and press Next Step

The screenshot shows the "Create new Task Definition" wizard, Step 1: Select launch type compatibility. It asks to select which launch type you want your task definition to be compatible with based on where you want to launch your task. There are two options: "FARGATE" and "EC2".

**FARGATE**

- Icon: A yellow hat icon.
- Price: Price based on task size.
- Requirements: Requires network mode awsvpc.
- Notes: AWS-managed infrastructure, no Amazon EC2 instances to manage.

**EC2**

- Icon: Two orange squares icon.
- Price: Price based on resource usage.
- Requirements: Multiple network modes available.
- Notes: Self-managed infrastructure using Amazon EC2 instances.

\*Required

Cancel Next step

Feedback English (US) © 2008 - 2019, Amazon Web Services, Inc. or its affiliates. All rights reserved. Privacy Policy Terms of Use

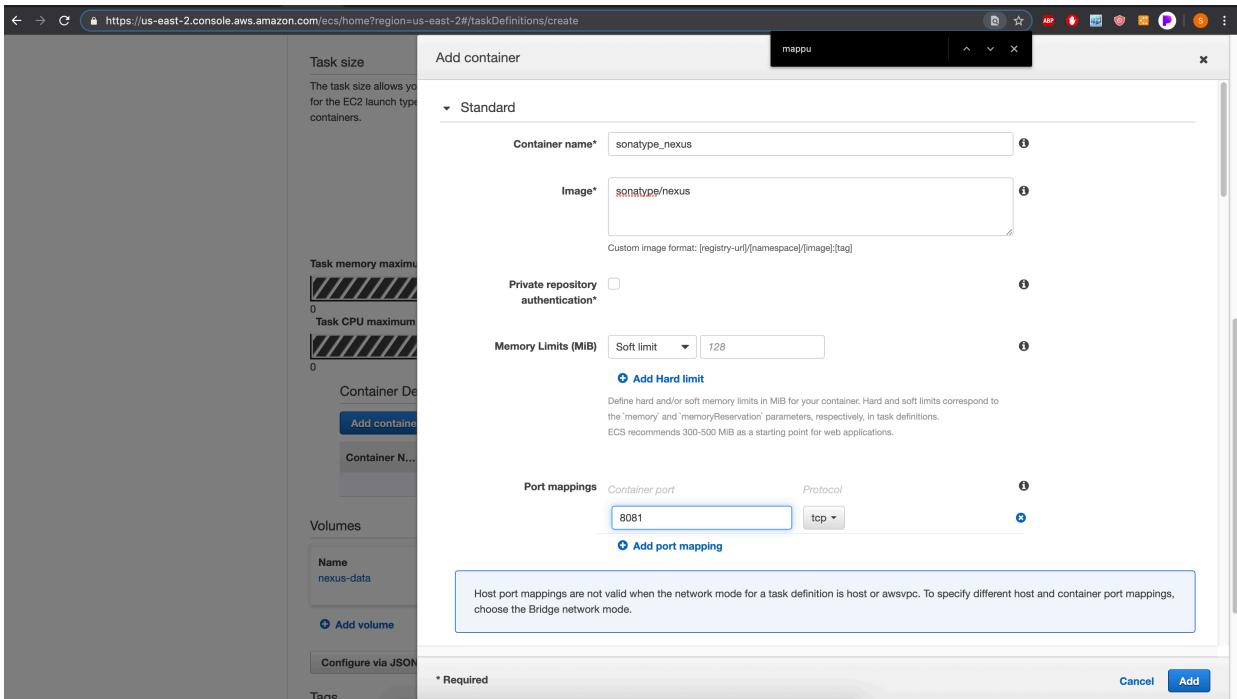
- Enter the task name, and select a Task Role, a Task Execution Role, and sizing parameters

The screenshot shows the 'Create new Task Definition' page in the AWS ECS console. The URL is <https://us-east-2.console.aws.amazon.com/ecs/home?region=us-east-2#/taskDefinitions/create>. The top navigation bar includes 'Services' and 'Resource Groups'. The search bar contains 'mapu'. The main content area is titled 'Configure task and container definitions'. It shows the 'Task Definition Name' set to 'nexus'. Under 'Requires Compatibilities', 'FARGATE' is selected. The 'Task Role' dropdown is set to 'ecsTaskExecutionRole'. The 'Network Mode' dropdown is set to 'awsvpc'. The 'Task execution IAM role' dropdown is also set to 'ecsTaskExecutionRole'. A note states: 'This role is required by tasks to pull container images and publish container logs to Amazon CloudWatch on your behalf. If you do not have the ecsTaskExecutionRole already, we can create one for you.' Below these settings is a 'Task size' section.

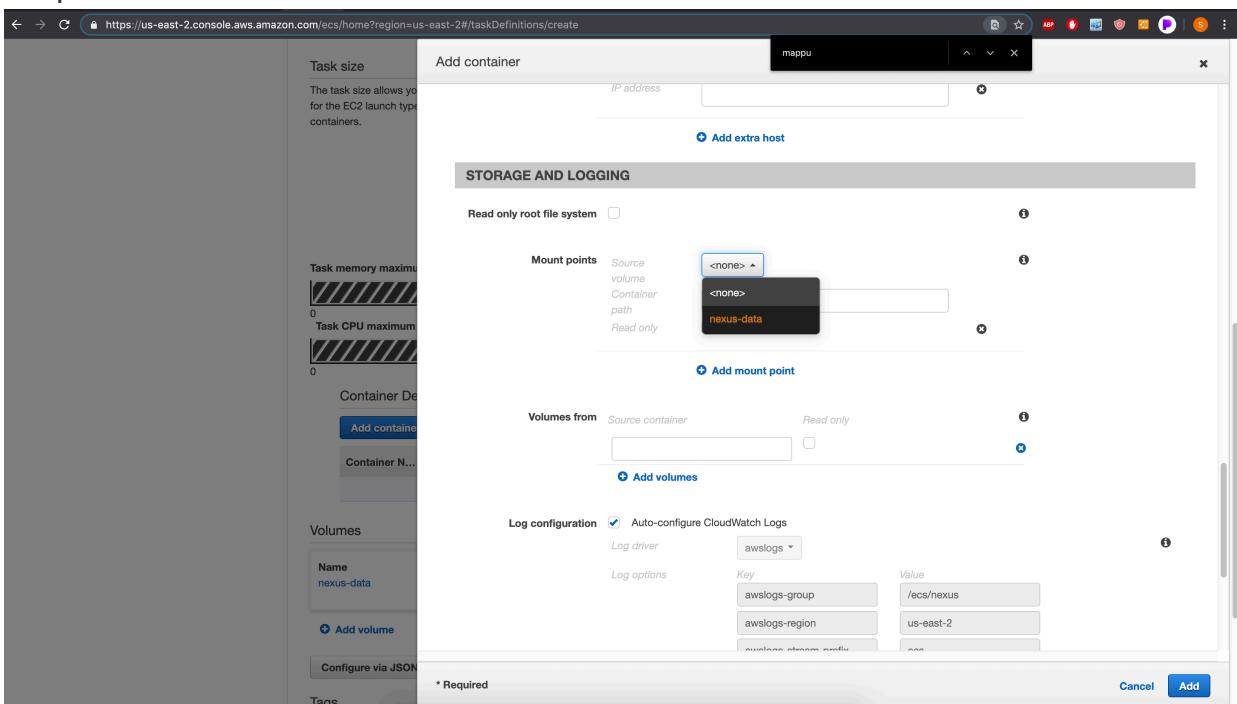
- Select Add Volume, enter the Volume Name and and press add

The screenshot shows the 'Task size' configuration section with a modal dialog titled 'Add volume' overlaid. The dialog has a 'Name' input field containing 'nexus-data'. A note below it says: 'Specify a volume driver' and 'This feature is not supported when the Fargate launch type is used.' At the bottom of the dialog are 'Cancel' and 'Add' buttons. The background shows other configuration sections like 'Container Definitions' (with 'Add container' button) and 'Volumes' (with 'Add volume' button).

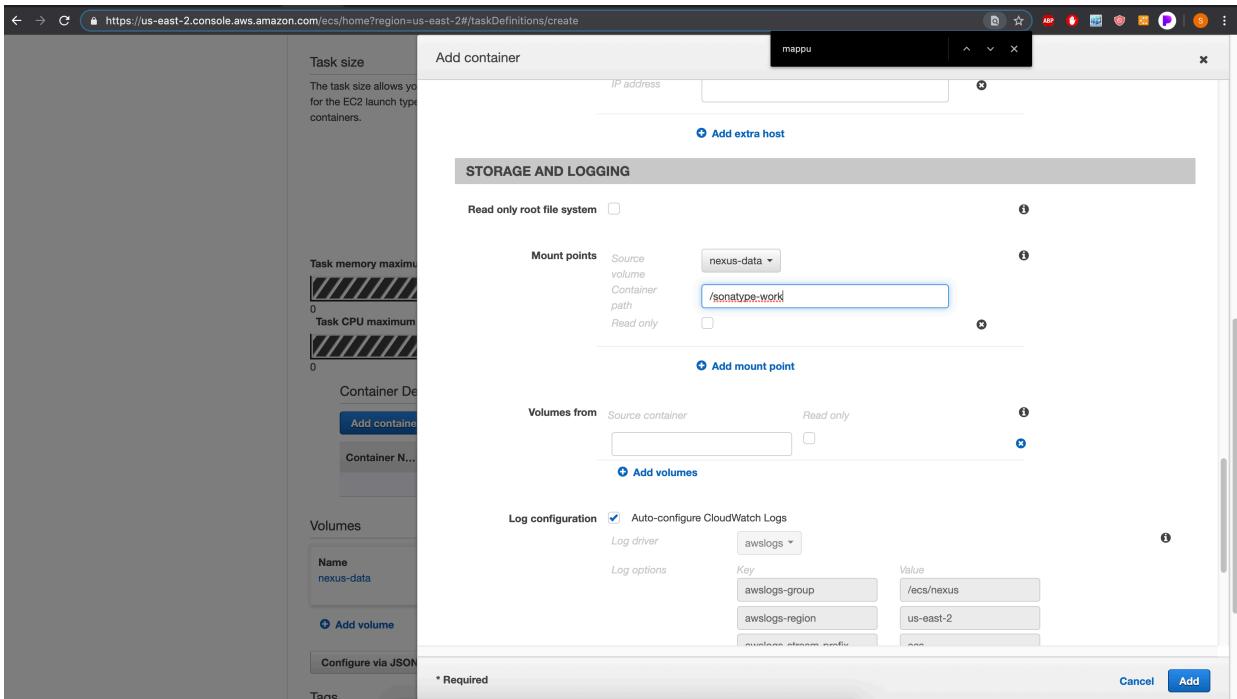
- Select Add Container.
- Enter the Container Name, Image, Port mappings number(s) (select Add port mapping to add additional ports)



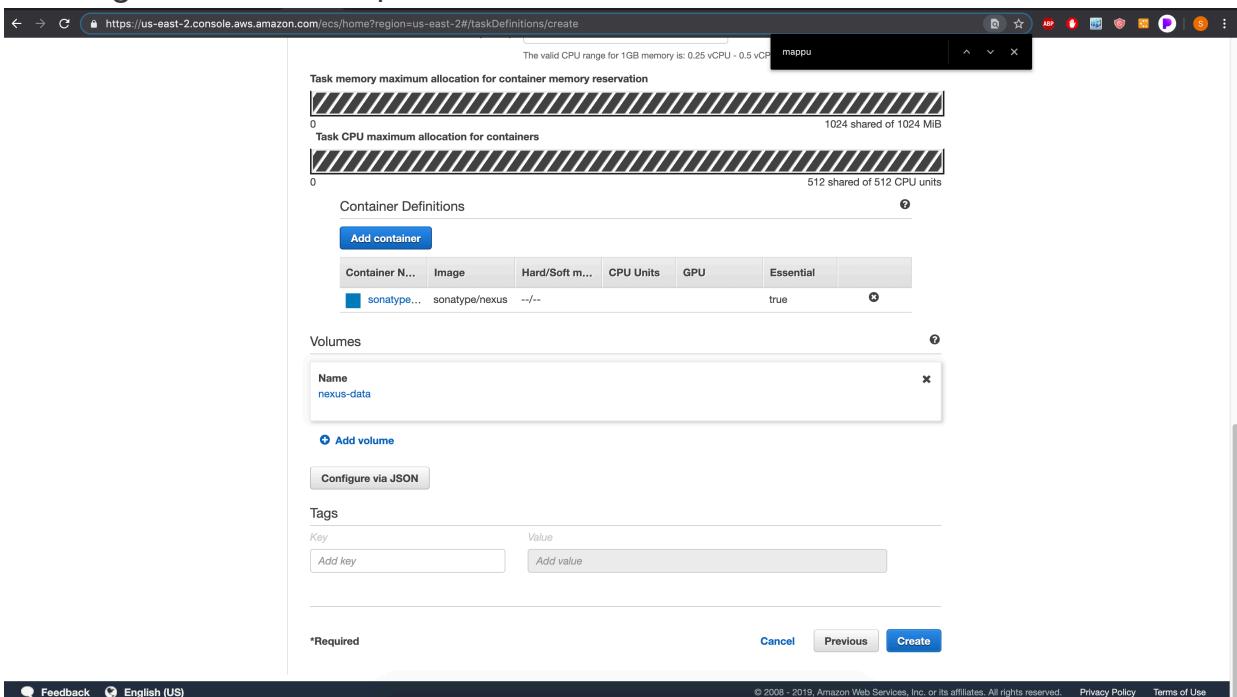
- Under Mount Points, if applicable, select a volume entered previously from the drop down



- Enter the mount point (s) (Select Add mount point if additional are required).
- Make sure Auto-configure CloudWatch Logs is selected.
- Press Add.



- Add Tags if desired and press create



- Once the Task has been created successfully, select View task definition

The screenshot shows the AWS CloudWatch Log Group creation page. At the top, it says "Create Task Definition: nexus". Below that, a green box indicates "nexus succeeded". Under "Create CloudWatch Log Group", another green box shows a checkmark and the message "CloudWatch Log Group created CloudWatch Log Group /ecs/nexus". At the bottom right, there are "Back" and "View task definition" buttons.

- Review the task definition, if desired

The screenshot shows the AWS Task Definition configuration page for "Task Definition: nexus:1". The left sidebar shows "Amazon ECS" and "Task Definitions" selected. The main area displays the task definition details:

- Task Definition Name:** nexus
- Task Role:** ecsTaskExecutionRole
- Network Mode:** awsvpc (with a note about Docker's default networking mode)
- Compatibilities:** EC2, FARGATE
- Requires compatibilities:** FARGATE
- Task execution IAM role:** A note stating this role is required for tasks to pull container images and publish logs.
- Task execution role:** ecsTaskExecutionRole
- Task size:** (with a question mark icon)

[Return to Table of Contents](#)

## Services

Services are constructs within ECS that allow for the management, grouping, load balancing, fault tolerance, etc. of tasks. When a service is created based on a task

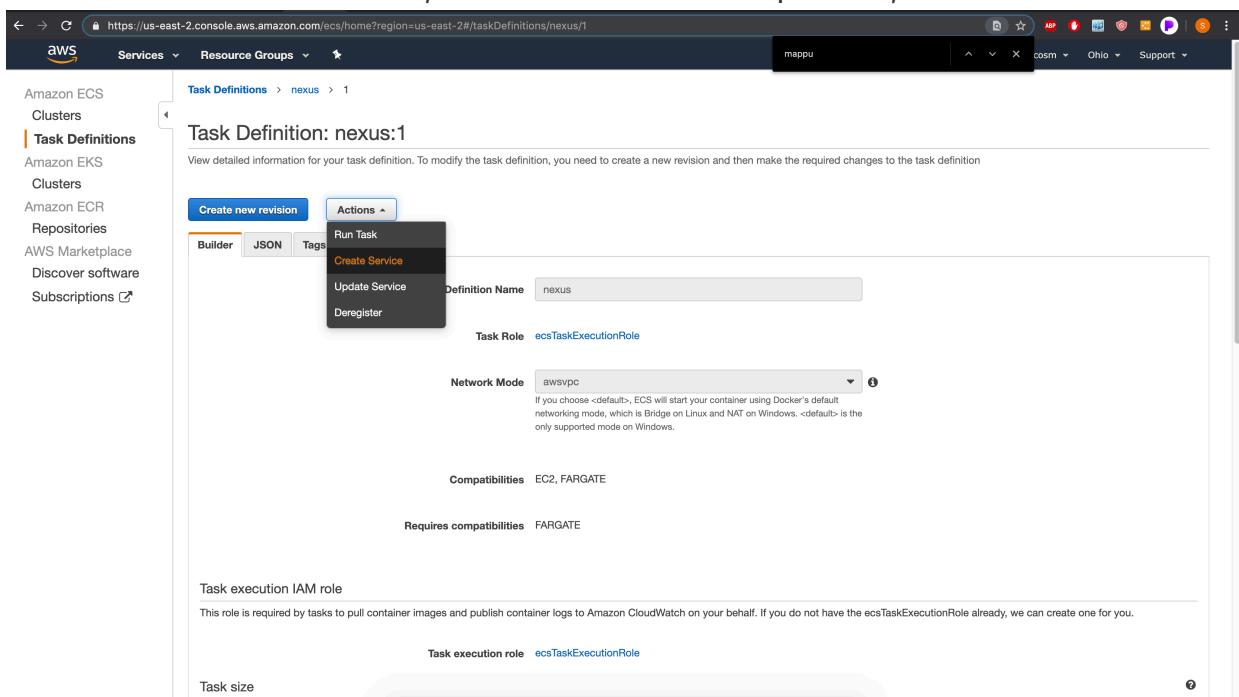
definition, ECS will instantiate tasks (containers) automatically and re-start them up if they fail. It is possible to have multiple different task definitions, of the same or different types, within a single service. Note, it is possible to have tasks not managed by a service, these have no fault tolerance nor take advantage of any other service properties. In this example we will create a service based on a task definition and let ECS create the task for us. Note, before deleting a service, you must stop all running tasks of that service and set the number of tasks to 0, else the service will restart the tasks before the service can be deleted.

#### ####Create a Service

Repeat the Following for each desired service definition

Note, if navigating to an application , eg gitlab, using the IP immediately after creation, the application may not yet be available due to initialization delays or processing. Please wait a few minutes and try again later.

- Select a task definition and, under the Actions drop down, select Create Service



- For Launch type select Fargate
- Select the cluster you created earlier
- Enter the service name from the table above (though it can be anything)
- Enter 1 for the number of tasks (we only need 1 instance of each container/task)

definition for the present practice)

- Press Next Step

The screenshot shows the 'Configure service' step of the AWS ECS Create Service wizard. The URL is <https://us-east-2.console.aws.amazon.com/ecs/home?region=us-east-2#/taskDefinitions/nexus/1/createService>. The 'Launch type' is set to FARGATE. The 'Task Definition' dropdown shows 'nexus' under 'Family' and '1' under 'Revision'. The 'Platform version' is set to LATEST. The 'Cluster' dropdown shows 'jenkinsdefault'. The 'Service name' is 'sonatype\_nexus'. The 'Service type\*' is set to 'REPLICAS'. The 'Number of tasks' is 1. The 'Minimum healthy percent' is 100, and the 'Maximum percent' is 200.

- Scroll up to the top of the page
- Select your VPC (10.0.0.0/24, also the hover text will identify it as a student VPC)
- Select a subnet from the drop down
- Within the Auto-assign public IP drop down, select ENABLED
- Under Security Groups, select Edit

Step 1: Configure service

Step 2: Configure network

Step 3: Set Auto Scaling (optional)

Step 4: Review

Configure network

VPC and security groups

Cluster VPC\* vpc-0a8f63ffd28e39ced (10.0.0/...)

Subnets\*

Security groups\*

Auto-assign public IP

Health check grace period

Load balancing

Load balancer  None

- Select an existing Security Group
- Select the group with Student VPC in the description
- Press Save
- Select Next Step

**Configure security groups**

A security group is a set of firewall rules that control the traffic for your task. On this page, you can add rules to allow specific traffic to reach your task, or you can choose to use an existing security group. [Learn more](#).

**Assigned security groups**

- Create new security group
- Select existing security group

**Existing security groups**

All existing security groups for the VPC of this cluster are listed below.

1 selected					< 0-0 >
	Security group ID	Name	Description	Actions	
<input checked="" type="checkbox"/>	sg-016e9fba44d930d92	Dynamic2-StudentVPCSecurityGro...	Student VPC Security Group	<a href="#">Copy to new</a>	
<input type="checkbox"/>	sg-050f7aa077e6edfa2	gitlab-2784	2019-04-16T15:33:53.395Z	<a href="#">Copy to new</a>	
<input type="checkbox"/>	sg-0793fec6a9c4a1c01	default	default VPC security group	<a href="#">Copy to new</a>	

**Inbound rules for selected security groups**

Security group ID	Type	Protocol	Port range	Source
sg-016e9fba44d930d92	HTTP	TCP	80	0.0.0.0/0
sg-016e9fba44d930d92	customtcp	TCP	9000	0.0.0.0/0
sg-016e9fba44d930d92	customtcp	TCP	8080	0.0.0.0/0
sg-016e9fba44d930d92	SSH	TCP	22	0.0.0.0/0
sg-016e9fba44d930d92	customtcp	TCP	50000	0.0.0.0/0
sg-016e9fba44d930d92	customtcp	TCP	8090	0.0.0.0/0
sg-016e9fba44d930d92	HTTPS	TCP	443	0.0.0.0/0

[Cancel](#) [Save](#)

- **Select Next Step**

<https://us-east-2.console.aws.amazon.com/ecs/home?region=us-east-2#/TaskDefinitions/nexus/1/createService>

**Create Service**

Step 1: Configure service  
Step 2: Configure network  
**Step 3: Set Auto Scaling (optional)**  
Step 4: Review

**Set Auto Scaling (optional)**

Automatically adjust your service's desired count up and down within a specified range in response to CloudWatch alarms. You can modify your Service Auto Scaling configuration at any time to meet the needs of your application.

**Service Auto Scaling**

- Do not adjust the service's desired count
- Configure Service Auto Scaling to adjust your service's desired count

\*Required [Cancel](#) [Previous](#) [Next step](#)

[Feedback](#) [English \(US\)](#)

© 2008 - 2019, Amazon Web Services, Inc. or its affiliates. All rights reserved. [Privacy Policy](#) [Terms of Use](#)

- Review and press Create Service

The screenshot shows the 'Create Service' step of the AWS ECS Task Definition creation process. It includes the following sections:

- Configure network:** Shows VPC ID: vpc-0a8f63ffd28e39ced, Subnets: subnet-0c12a48c3055e9708, Create new security group: sonaty-6743, and Auto assign IP: ENABLED.
- Configure service discovery:** Shows Namespace: ns-iz26fjly45cjagg, Service discovery name: sonatype\_nexus, Enable ECS task health propagation: true, and DNS record type and TTL: A 60.
- Set Auto Scaling (optional):** Shows 'not configured'.

At the bottom, there are 'Cancel', 'Previous', and 'Create Service' buttons. The 'Create Service' button is highlighted in blue.

- Once the service is successfully created, select View Service

The screenshot shows the 'View Service' step of the AWS ECS Task Definition creation process, displaying the following successful steps:

- Create security group:** Create security group succeeded sg-08677ef9f87f9e049
- Set inbound rules:** Set inbound rules succeeded sg-08677ef9f87f9e049
- Create service discovery service:** Create service discovery service succeeded arm:aws:servicediscovery:us-east-2:443007076818:service/srv-bcfmvbuuvvuxelnz created
- Create Service:** Service created. Tasks will start momentarily. View: sonatype\_nexus

Below these, it says 'Additional integrations you can connect to your ECS service'.

- Review the service's properties. Observe the creation of the new task.
- Once the task is present, select the task ID.
- View the task status, until the status is green and says Running
- Now you should be able to see both the private and public IP address of the container. This is important for the next steps

**Task : 325b875c-9e8f-4686-872a-937c986973d7**

Name	Container Id	Status	Image	CPU Units	Hard/Soft memo...	Essential
sonatype_nexus	a7474130-6432-44a7-8105-d79153c33a3a	RUNNING	sonatype/nexus	0	--/--	true

## NOTE

- If you would like, select clusters from the left menu and see the overall service and task numbers.

Cluster	Services	Running tasks	Pending tasks	CPUUtilization	MemoryUtilization	Container instances
jenkinsdefault >	6	5	1	No data	No data	0
microcosm >	0	0	0	No data	No data	0

- Selecting a cluster, then the task tab, will allow you to select a task and see its IP and other details

**Cluster : jenkinsdefault**

Status: ACTIVE

Registered container instances	0							
Pending tasks count	0 Fargate, 0 EC2							
Running tasks count	6 Fargate, 0 EC2							
Active service count	6 Fargate, 0 EC2							
Draining service count	0 Fargate, 0 EC2							
Task	Task definition	Container instan...	Last status	Desired status	Started By	Group	Launch type	Platform version
108c5928-1525-4...	hubot-slack:1	--	RUNNING	RUNNING	ecs-svc/9223370...	service:hubot	FARGATE	1.3.0
13c3e227-2949-4...	first-run-task-defi...	--	RUNNING	RUNNING	ecs-svc/9223370...	service:jenkins-de...	FARGATE	1.3.0
325b875c-9ef8-4...	nexus:1	--	RUNNING	RUNNING	ecs-svc/9223370...	service:sonatype_...	FARGATE	1.3.0
62af7548-c047-4...	zap2docker:1	--	RUNNING	RUNNING	ecs-svc/9223370...	service:owaspzap	FARGATE	1.3.0
89bd2a92-90fc-4...	sonarqube:2	--	RUNNING	RUNNING	ecs-svc/9223370...	service:sonarqube	FARGATE	1.3.0
caaa3991-accd-4...	gitlab:1	--	RUNNING	RUNNING	ecs-svc/9223370...	service:gitlab	FARGATE	1.3.0

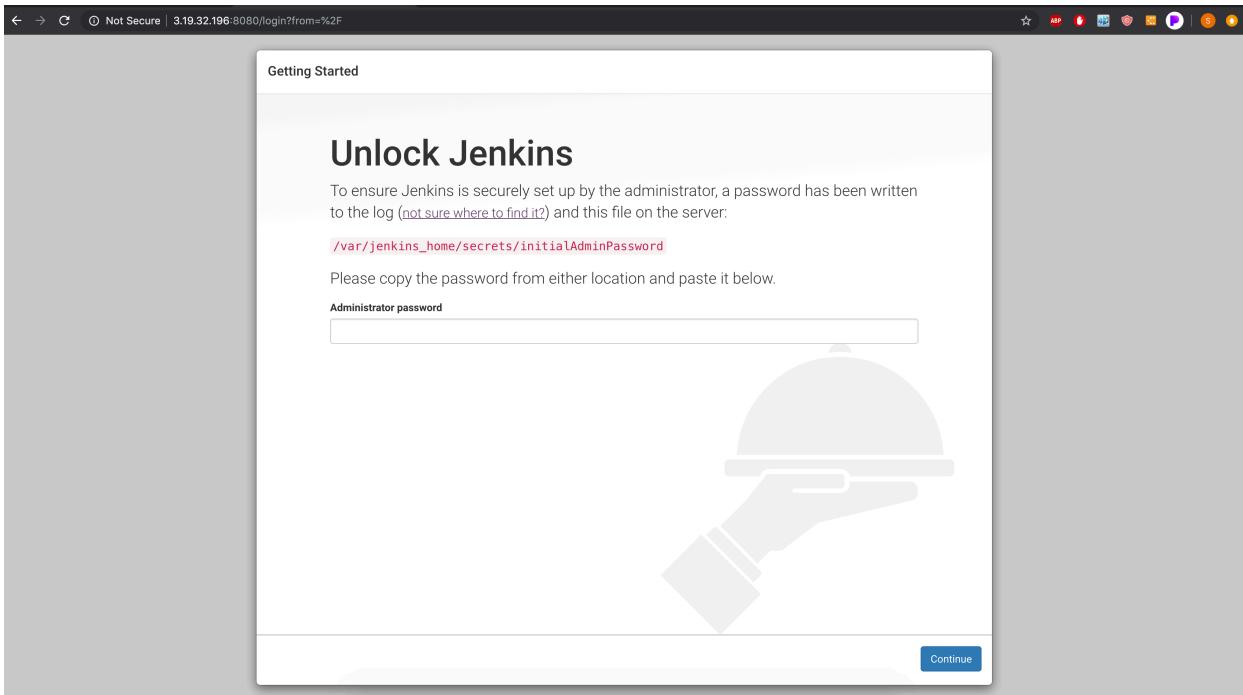
[Return to Table of Contents](#)

## Setting up Jenkins for DevOps Pipeline (Microcosm)

We will assume that the Jenkins task indicates that it is running

*Note, that without setting up persistant volume storage (not covered here), if the task (container) fails in any way or is stopped or otherwise shutdown, ALL of the settings and data entered into that container will be lost. Therefore, be careful.*

- Navigate to the IP address of the Jenkins task at port 8080 (eg 24.23.22.21:8080)
  - Remember that all task IP addresses can be found by clicking on the cluster, then the tasks tab and finally the task ID
- The first time you navigate to Jenkins, you will see the following prompt asking you to unlock Jenkins. To do this, we will first need to look in the CloudWatch logs to get a key string (only available after you navigate to Jenkins the first time).



- From the services dropdown find/select CloudWatch
- Select Logs from the column on the Left
- Select /ecs/jenkinsTSTNAME
- Sort the Log Streams by clicking on the Last Event Time column header so that the most recent log stream date is at the top
- Select the most recent Log Stream

Last Event Time	Log Stream Name Prefix
2019-04-16 13:27 UTC-6	ecs/jenkins/92c59a81-6a6a-45e0-bc7b-3932928df9a9
2019-04-16 13:24 UTC-6	ecs/jenkins/dd0696c2-acd9-4e5-b512-70c1d84df06
2019-04-16 13:18 UTC-6	ecs/jenkins/234135b5-3ad5-40a5-a8bc-2c96a5977dd3
2019-04-16 13:12 UTC-6	ecs/jenkins/cb06a3e4-8354-4392-9abc-b577bd0caef
2019-04-16 13:06 UTC-6	ecs/jenkins/4c39ca9d-512e-43e9-afe0-49f6d1b16d87
2019-04-16 13:01 UTC-6	ecs/jenkins/23d544d-46a1-4c11-bcfe-59a39fd398b3
2019-04-16 12:55 UTC-6	ecs/jenkins/1ba1470-002c-4503-8ec6-d2b7c93d48ce
2019-04-16 12:49 UTC-6	ecs/jenkins/f87338b8-e9fe-462b-99ec-00c1ca93b43
2019-04-16 12:43 UTC-6	ecs/jenkins/ef11b1d9-3bcf-4c4e-9efe-1f78adb17ed0
2019-04-16 12:38 UTC-6	ecs/jenkins/a4b79c1e-377a-4bd8-b3a4-79bcece8eae4
2019-04-16 12:36 UTC-6	ecs/jenkins/40e70ae0-edc1-4778-8a5d-e77ef096b7f5
2019-04-16 12:35 UTC-6	ecs/jenkins/300d0d420-7fc4-a441-b81f-5ab001dcde23
2019-04-16 12:34 UTC-6	ecs/jenkins/7eea7fe8-2a5d-4170-a077-66a3ba95635d
2019-04-16 12:32 UTC-6	ecs/jenkins/1927e44d-fbb8-4dbc-b942-870140cc4ced
2019-04-16 12:31 UTC-6	ecs/jenkins/12c54ba2-90db-416d-ab59-c718a7624384
2019-04-16 12:30 UTC-6	ecs/jenkins/a6329022-7683-488a-af9f-4ac0a1dc0df9
2019-04-16 12:29 UTC-6	ecs/jenkins/02182562-c89e-4287-a4df-ae690f053a2c
2019-04-16 12:27 UTC-6	ecs/jenkins/e7ff68630-8228-403f-b4dd-c44916ba2d35
2019-04-16 12:26 UTC-6	ecs/jenkins/816ed740-ea0e-40a7-9954-55bf447afe3
2019-04-16 12:25 UTC-6	ecs/jenkins/30da76c7-0dc8-4448-9e93-9ee15e385485
2019-04-16 12:10 UTC-6	ecs/jenkins/1b42226d-9eab-4bf4-824b-a13b1fe4d8d
2019-04-16 11:53 UTC-6	ecs/jenkins/7071fa1c-47f5-43ba-a40c-8a5a5a61a720
2019-04-16 10:36 UTC-6	ecs/jenkins/c272e68f-c302-4b61-8610-cee6c63c6bc3
2019-04-15 15:59 UTC-6	ecs/jenkins/0be7fe7b-eb7d-43c3-9076-9bad78e68e9f

- Scroll down until you find the line "Please use the following password to proceed"

to installation"

- Select the line below this, which contains a string of seemingly random character
- Highlight and copy that string
- Paste that string into the Jenkins Getting Started window and press Continue

The screenshot shows the AWS CloudWatch Logs interface. On the left, there's a sidebar with navigation links like CloudWatch, Dashboards, Alarms, Events, Rules, Event Buses, Logs (which is selected), Insights, Metrics, and Favorites. The main area displays a log group named 'ecs/jenkins'. A specific log stream is selected, showing log entries from April 16, 2019, at 19:28:31 UTC. The log entries include:

```

2019-04-16
19:28:31 Apr 16, 2019 7:28:31 PM org.springframework.context.support.AbstractApplicationContext prepareRefresh
19:28:31 INFO: Refreshing org.springframework.web.context.support.StaticWebApplicationContext@1b118f44; display name [Root WebApplicationContext]; startup date [Tue Apr 16 19:28:31 UTC 2019]
19:28:31 Apr 16, 2019 7:28:31 PM org.springframework.context.support.AbstractApplicationContext obtainFreshBeanFactory
19:28:31 INFO: Bean factory for application context [org.springframework.web.context.support.StaticWebApplicationContext@1b118f44]: org.springframework.beans.factory.support.DefaultListableBeanFactory
19:28:31 Apr 16, 2019 7:28:31 PM org.springframework.beans.factory.support.DefaultListableBeanFactory@4a15c682: defining beans [authenticationManager]; root of factory hierarchy
19:28:31 INFO: Pre-instantiating singletons in org.springframework.beans.factory.support.DefaultListableBeanFactory@4a15c682; defining beans [authenticationManager]; root of factory hierarchy
19:28:32 Apr 16, 2019 7:28:32 PM org.springframework.context.support.StaticWebApplicationContext@33803935: display name [Root WebApplicationContext]; startup date [Tue Apr 16 19:28:32 UTC 2019]
19:28:32 INFO: Refreshing org.springframework.web.context.support.StaticWebApplicationContext@33803935; display name [Root WebApplicationContext]; startup date [Tue Apr 16 19:28:32 UTC 2019]
19:28:32 INFO: Bean factory for application context [org.springframework.web.context.support.StaticWebApplicationContext@33803935]: org.springframework.beans.factory.support.DefaultListableBeanFactory
19:28:32 Apr 16, 2019 7:28:32 PM org.springframework.beans.factory.support.DefaultListableBeanFactory@4a15c682: defining beans [filter,legacy]; root of factory hierarchy
19:28:32 INFO: Pre-instantiating singletons in org.springframework.beans.factory.support.DefaultListableBeanFactory@4a15c682; defining beans [filter,legacy]; root of factory hierarchy
19:28:33 Apr 16, 2019 7:28:33 PM jenkins.install.SetupWizard init
19:28:33 INFO: ****
19:28:33 ****
19:28:33 ****
19:28:33 Jenkins initial setup is required. An admin user has been created and a password generated.
19:28:33 Please use the following password to proceed to installation:
19:28:33 47d0d8e8ea244d5b03ad2cd5ad78db
19:28:33 This may also be found at: /var/jenkins_home/secrets/initialAdminPassword
19:28:33 ****
19:28:33 ****
19:28:33 ****
19:28:33 -->
19:28:40 --> setting agent port for jnlp
19:28:41 --> setting agent port for jnlp... done
19:28:48 Apr 16, 2019 7:28:48 PM hudson.model.UpdateSite updateData
19:28:48 INFO: Obtained the latest update center data file for UpdateSource default

```

At the bottom of the log, there's a URL: <https://47d0d8e8ea244d5b03ad2cd5ad78db>

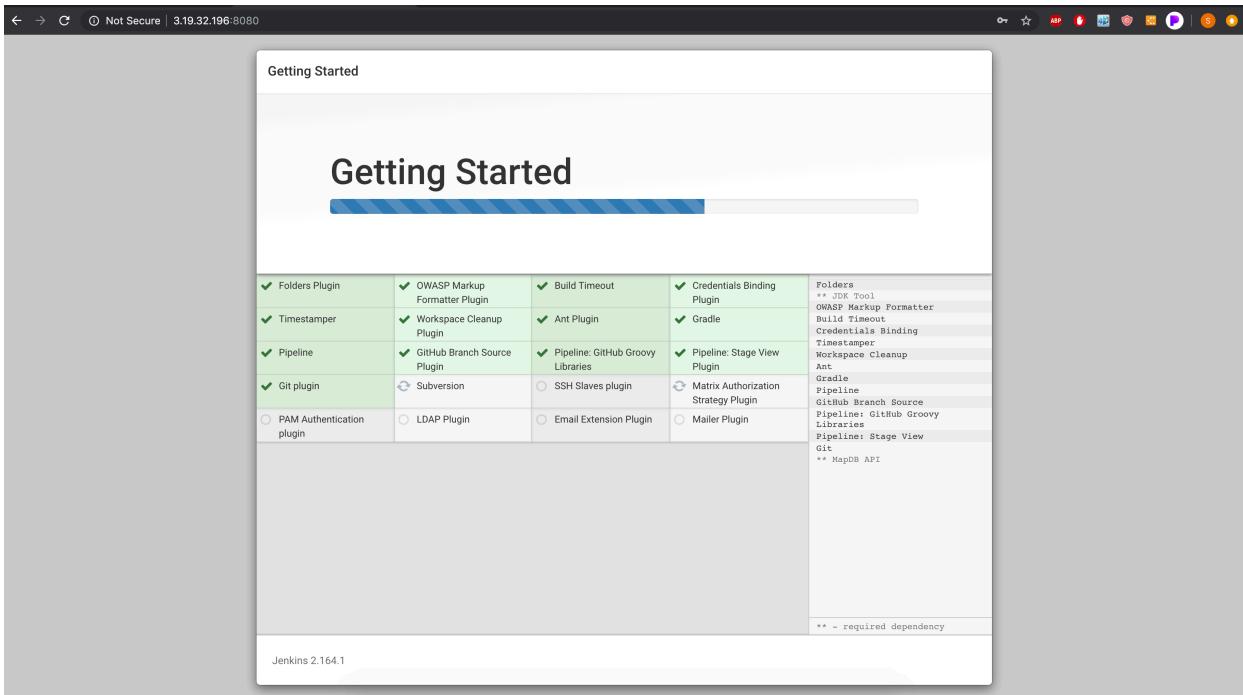
- Click Install suggested plugins

The screenshot shows the Jenkins 'Getting Started' page. The page title is 'Customize Jenkins'. It says: 'Plugins extend Jenkins with additional features to support many different needs.' There are two main buttons:

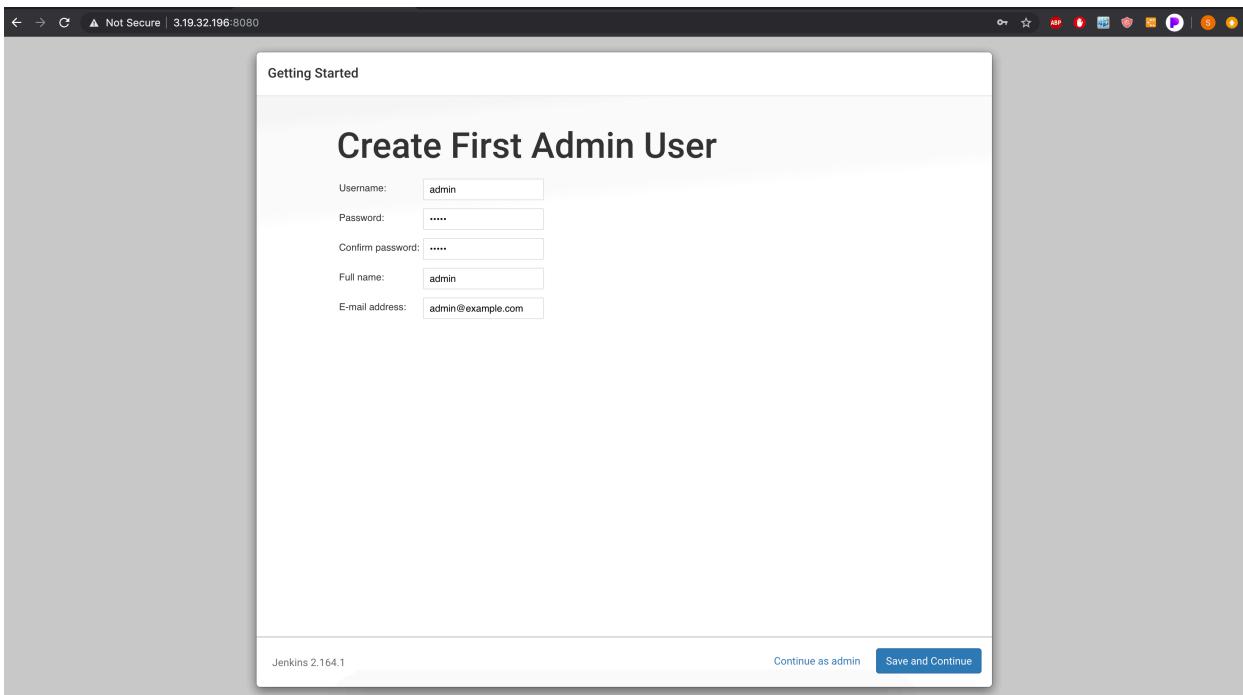
- Install suggested plugins**: A blue button with white text. Below it, a description reads: 'Install plugins the Jenkins community finds most useful.'
- Select plugins to install**: A grey button with black text. Below it, a description reads: 'Select and install plugins most suitable for your needs.'

At the bottom of the page, it says 'Jenkins 2.164.1'.

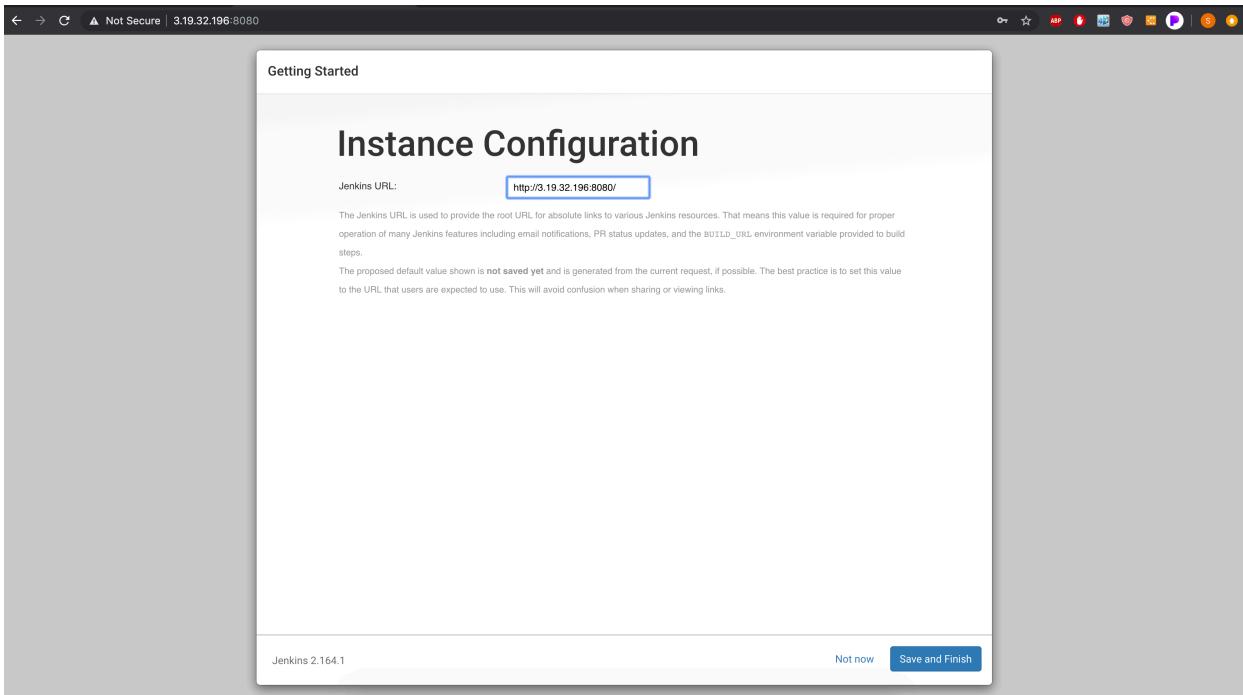
- Wait while the plugins are installed



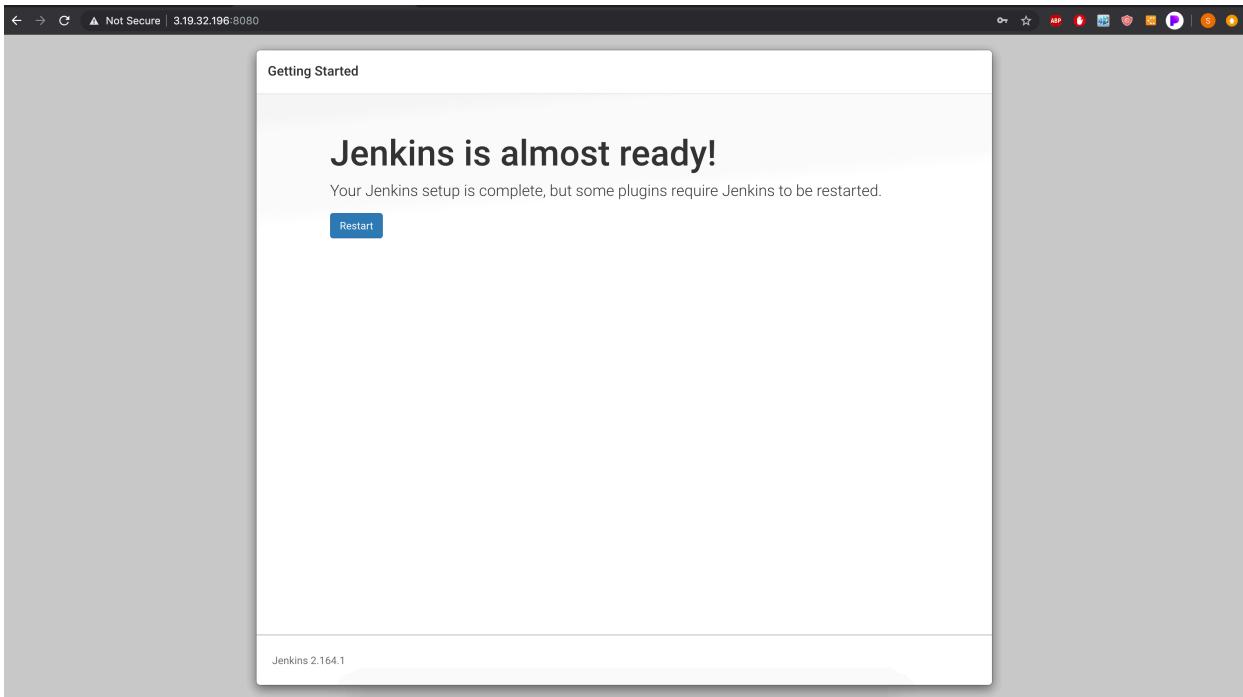
- Fill in the information for the admin user. Make sure to remember your password.
- Select Save and Continue



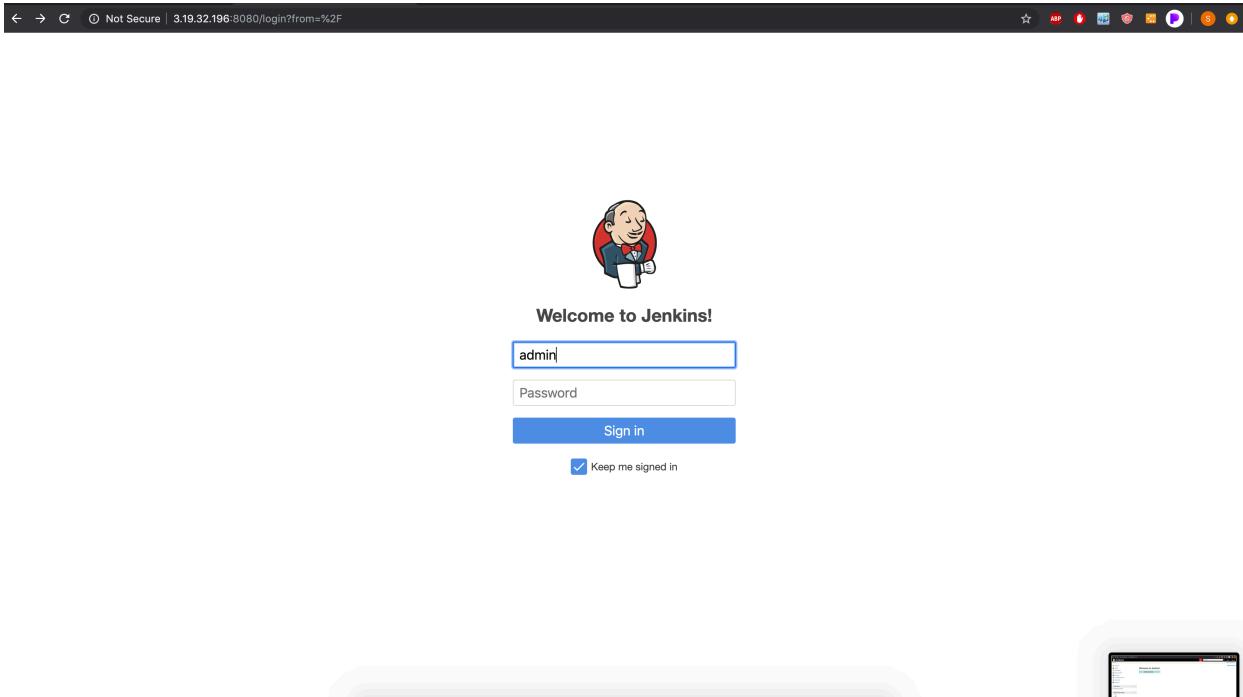
- Select Save and Finish



- Press Restart
- Wait (if waiting longer than a few minutes, refresh window or, open Jenkins in a new window)



- If, after restart, you see a login prompt, use the credentials you entered when creating the admin user previously



- From the main page, Select the warning in the corner and select Go to plugin manager
  - If no warnings are present, Select Manage Jenkins on the Left, then scroll down and select Manage Plugins

New version of Jenkins (2.164.2) is available for [download](#) ([changelog](#)). [Go to plugin manager](#) [Configure which of these warnings are shown](#)

Warnings have been published for the following currently installed components.

**Jenkins 2.164.1 core and libraries:**  
Multiple security vulnerabilities in Jenkins 2.171 and earlier, and LTS 2.164.1 and earlier

**Lockable Resources plugin 2.4:**  
XSS vulnerability

**Script Security Plugin 1.55:**  
Script Security sandbox bypass

**Pipeline: Groovy 2.64:**  
Script Security sandbox bypass

**Environment Injector Plugin 2.1.6:**  
Exposure of sensitive build variables stored by EnvInject 1.90 and earlier

- If updates are available, scroll to the bottom of the page and click select all
- Select Download Now and install after restart
- Click "Restart Jenkins when installation is complete and no jobs are running"
- at top-left menu, click "back to Dashboard"

The screenshot shows the Jenkins Plugin Manager interface. At the top, there are tabs for 'Updates' (selected), 'Available', 'Installed', and 'Advanced'. A search bar and filter button are also at the top. The main area displays a table of available plugins, sorted by name. The table has columns for Name, Version, and Installed. Plugins listed include 'Branch API', 'CloudBees Docker Hub/Registry Notification', 'Display URL API', 'Docker Commons', 'Docker Pipeline', 'Embeddable Build Status', 'Git Client', 'Lockable Resources', 'Mercurial', 'Pipeline: Build Step', 'Pipeline: Declarative', and 'Pipeline: Declarative Extension Points API'. Most plugins have a 'Install' link next to them. At the bottom of the page, there are buttons for 'Download now and install after restart' and 'Check now'.

Name	Version	Installed
<a href="#">Branch API</a>	2.4.0	2.1.2
<a href="#">CloudBees Docker Hub/Registry Notification</a>	2.4.0	2.3.0
<a href="#">Display URL API</a>	2.3.1	2.3.0
<a href="#">Docker Commons</a>	1.14	1.13
<a href="#">Docker Pipeline</a>	1.18	1.17
<a href="#">Embeddable Build Status</a>	2.0.1	2.0
<a href="#">Git Client</a>	2.7.7	2.7.6
<a href="#">Lockable Resources</a>	2.5	2.4
<a href="#">Mercurial</a>	2.6	2.5
<a href="#">Pipeline: Build Step</a>	2.9	2.8
<a href="#">Pipeline: Declarative</a>	1.3.8	1.3.6
<a href="#">Pipeline: Declarative Extension Points API</a>	1.3.8	1.3.6

- On the Plugin Manager page, select the Available Tab
  - search & select: "Git Plugin" (May be pre-installed)
  - search & select: "Pipeline Maven Integration"
  - search & select: "SonarQube Scanner"
  - search & select: "Nexus Platform"
  - search & select: "AWS CodeDeploy Plugin for Jenkins"
  - search & select: "AWS CloudWatch Logs Publisher"
  - click "install without restart" at bottom of page
  - check box next to "Restart Jenkins when installation is complete and no jobs are running."
  - at top-left menu, click "back to Dashboard"

The screenshot shows the Jenkins Plugin Manager interface. The left sidebar lists categories like 'Available' and 'Installed'. The main area displays a table of available plugins. Each plugin entry includes a checkbox, a brief description, and its version number. At the bottom, there are buttons for 'Install without restart', 'Download now and install after restart', and 'Check now'.

	Description	Version
<input type="checkbox"/> SLOCCount	For a given Parameterized Project, this plugin shows the builds sorted by the parameters used to execute the builds	1.5
<input type="checkbox"/> SonarQube Scanner	This plugin parses SLOCCount output files to produce project and build reports.	1.24
<input checked="" type="checkbox"/> Sonargraph Integration	This plugin allows an easy integration of <a href="#">SonarQube</a> , the open source platform for Continuous Inspection of code quality.	2.8.1
<input type="checkbox"/> Sonargraph	This plugin integrates <a href="#">Sonargraph</a> functionality into Jenkins, for Sonargraph versions 8 and 9	2.2.1
<input type="checkbox"/> Splunk	This plugin integrates <a href="#">Sonargraph</a> functionality into Jenkins	1.6.4
<input type="checkbox"/> Statistics Gatherer	Splunk plugin for Jenkins provides deep insights into your Jenkins master and slave infrastructure, job and build details such as console logs, status, artifacts, and an incredibly efficient way to analyze test results.	1.7.1
<input type="checkbox"/> StepCounter	Captures Statistics related to Jenkins Builds, Build Step, SCM checkouts, Jobs and Queue and sends them where you want.	2.0.3
<input type="checkbox"/> StepCounter	This plugin is Step Counter plugin.	2.0.0
<input checked="" type="checkbox"/> Summary Display	This plugin show ACI reports.	1.15
<input type="checkbox"/> TAP	This plugin publishes TAP test results.	2.2.2
<input type="checkbox"/> Task Scanner	The Task Scanner Plug-in reached end-of-life. All functionality has been integrated into the <a href="#">Warnings Next Generation Plugin</a> .	4.53
<input type="checkbox"/> Tattletale	Integration plugin for Tattletale analysis tool.	0.3
<input type="checkbox"/> Test Results Analyzer	This plugin shows history of test execution results in a tabular or graphical format.	0.35
<input type="checkbox"/> Test stability history	Displays test stabilities - i.e. the history of failed tests.	2.3
<input type="checkbox"/> Testability Explorer	Plugin for the Testability Explorer <a href="http://code.google.com/p/testability-explorer/">http://code.google.com/p/testability-explorer/</a>	0.4
<input type="checkbox"/> TestComplete support	Demo automation tools with TestComplete or TestExecute for free! <a href="#">TestComplete</a>	2.1

- Wait until the installation completes
- Jenkins will restart in the background and the UI may appear to be hung. Feel free to refresh the page after a few minutes

The screenshot shows the Jenkins Update Center. The top navigation bar includes links for 'Back to Dashboard', 'Manage Jenkins', and 'Manage Plugins'. The main content area is titled 'Installing Plugins/Upgrades' and contains two sections: 'Preparation' and 'Nexus Platform'. The 'Preparation' section lists steps: 'Checking internet connectivity', 'Checking update center connectivity', and 'Success'. The 'Nexus Platform' section lists various plugins with their status: Nexus Platform (Installing), Slack Notification (Pending), Config File Provider (Pending), Pipeline Maven Integration (Pending), HTML Publisher (Pending), Official OWASP ZAP (Pending), Summary Display (Pending), SonarQube Scanner (Pending), Extended Choice Parameter (Pending), Custom Tools (Pending), Selenium HTML report (Pending), and Restarting Jenkins (Pending). At the bottom, there are links to 'Go back to the top page' and 'Restart Jenkins when installation is complete and no jobs are running'.

- Return to the home page and click "Manage Jenkins"
- click "Global Tool Configuration"

The screenshot shows the Jenkins 'Manage Jenkins' page. On the left, there's a sidebar with links like 'New Item', 'People', 'Build History', 'Manage Jenkins', 'My Views', 'Lockable Resources', 'Credentials', and 'New View'. Below these are 'Build Queue' and 'Build Executor Status' sections. The main content area has a blue header bar stating 'New version of Jenkins (2.164.2) is available for download (changelog.)'. Underneath, a red box highlights 'Warnings have been published for the following currently installed components.' It lists several vulnerabilities across different plugins: Jenkins 2.164.1 core and libraries (multiple security vulnerabilities), Pipeline: Groovy 2.64 (Script Security sandbox bypass), Official OWASP ZAP Jenkins Plugin 1.1.0 (Credentials stored in plain text), Environment Injector Plugin 2.1.6 (Exposure of sensitive build variables stored by EnvInject 1.90 and earlier), Script Security Plugin 1.55 (Script Security sandbox bypass), Lockable Resources plugin 2.4 (XSS vulnerability), and Pipeline: Artifactory 2.1.0 (Pipeline: Artifactory security vulnerability). At the bottom of this section are four configuration links: 'Configure System', 'Configure Global Security', 'Configure Credentials', and 'Global Tool Configuration'. The 'Global Tool Configuration' link is underlined.

- Scroll down to the Maven: Maven Installations section and click "Add Maven"
- the form may not expand the first time. sometimes one or more page refreshes is required before this works.
- enter "petclinic" as name and make sure that Install automatically is selected
- click Apply

The screenshot shows the Jenkins 'Global Tool Configuration' page. It lists various tools: Gradle, Mercurial, SonarScanner for MSBuild, SonarQube Scanner, Ant, and Maven. The 'Maven' section is expanded, showing 'Maven installations' with an 'Add Maven' button. A sub-section for 'petclinic' is shown with a 'Name' field containing 'petclinic', a checked 'Install automatically' checkbox, and an 'Install from Apache' dropdown set to '3.6.1'. At the bottom are 'Save' and 'Apply' buttons.

- Click Apply
- Click Save

The screenshot shows the Jenkins Global Tool Configuration page. In the 'Custom tool' section, a new tool named 'ZAP\_2.6.0' is being configured. The 'Name' field contains 'ZAP\_2.6.0'. Under 'Extract \*.zip/\*tar.gz', the 'Label' field is empty, the 'Download URL for binary archive' is set to 'https://github.com/zaproxy/zaproxy/releases/download/2.6.0/ZAP\_2.6.0\_Linux.tar.gz', and the 'Subdirectory of extracted archive' is set to 'ZAP\_2.6.0'. The 'Install automatically' checkbox is checked. Buttons for 'Delete Installer' and 'Delete Custom tool' are visible. At the bottom, there are 'Save' and 'Apply' buttons.

## Slack & Hubot integration with Jenkins

[Return to Table of Contents](#)

## Deploying test code in DevOps Pipeline

We are going to use a publicly available, SEI created, test project hosted on Github to test our deployment pipeline. The following steps will guide you through that process.

### Public Github Setup

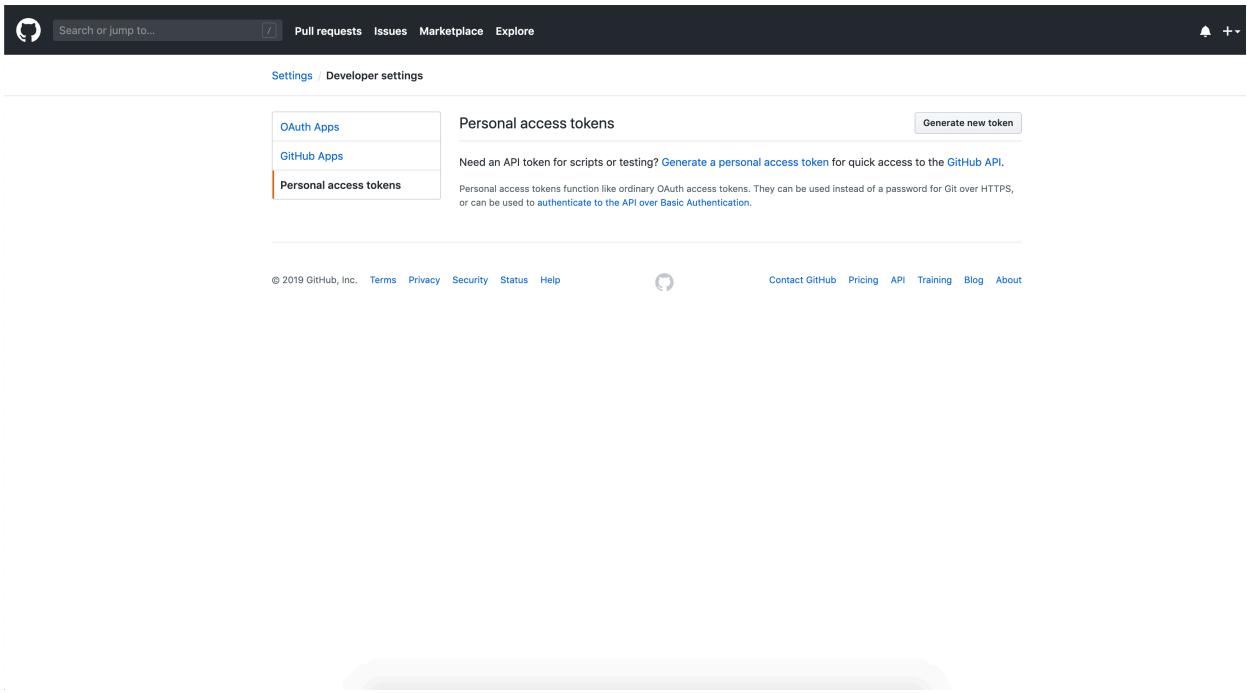
- Login to github (not your gitlab server) at <https://github.com> (or create an account if you don't have one already).
- After logging in, click on your user icon on the top right and select Settings

The screenshot shows the GitHub homepage. At the top right, there's a sidebar with links like 'Welcome to the new closer to the stuff you', 'Set your status', 'Your profile', 'Your repositories', 'Your projects', 'Your stars', 'Your gists', 'Help', 'Settings', and 'Sign out'. Below the sidebar, there are sections for 'Discover repositories' (listing 'Microsoft/AppCenter-SDK', 'Objective-C', 'lukesampson/scoop-extras', and 'PowerShell') and 'mozilla/DeepSpeech' (a TensorFlow implementation of Baidu's DeepSpeech architecture). The main content area features a large card for the 'Hello World' guide with buttons for 'Read the guide' and 'Start a project'. Below this is another card for 'Discover interesting projects and people to populate your personal news feed' with a 'Explore GitHub' button.

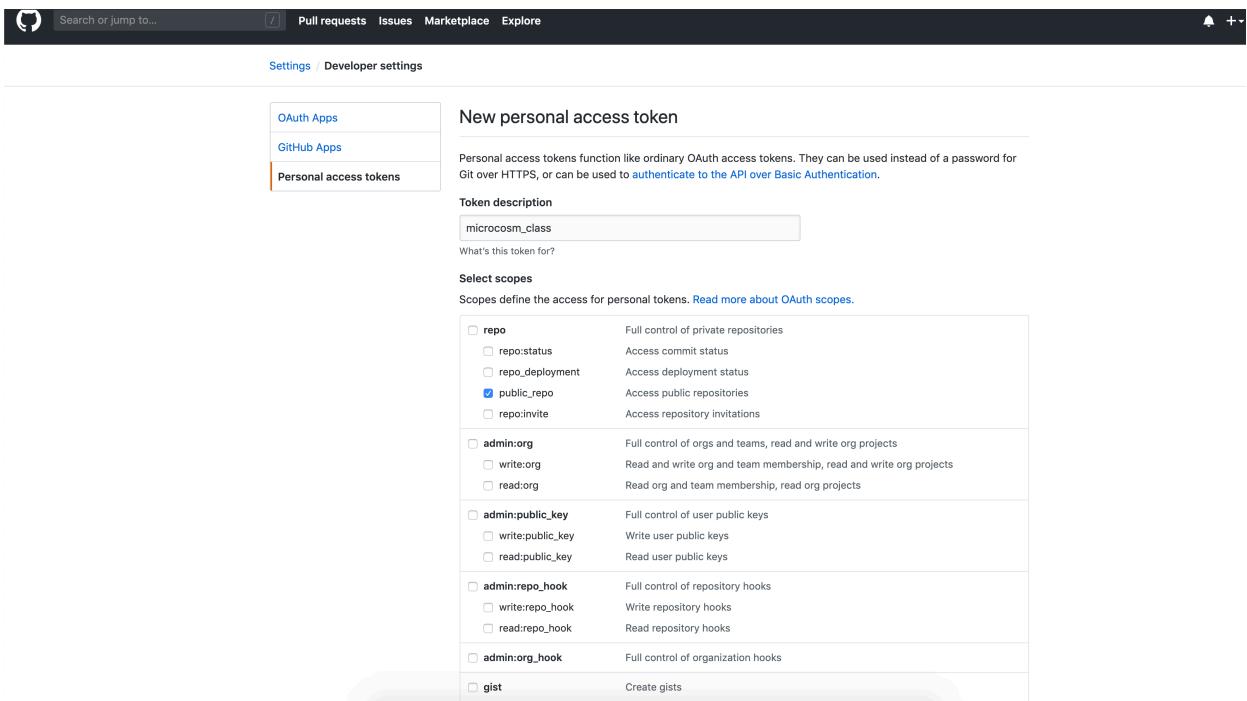
- Select Developer settings from the left menu list

The screenshot shows the 'Public profile' settings page. On the left, there's a sidebar with links: Personal settings, Profile, Account, Emails, Notifications, Billing, SSH and GPG keys, Security, Sessions, Blocked users, Repositories, Organizations, Saved replies, Applications, and Developer settings (which is highlighted). The main content area is titled 'Public profile' and contains sections for 'Name', 'Public email', 'Bio', 'URL', 'Company', and 'Location'. A note at the bottom states that all fields are optional and can be deleted at any time. A 'Update profile' button is at the bottom right.

- Select Person Access Tokens
- Click Generate new token



- Give your token a related description and select the public\_repo checkbox
- Complete the form and **copy the generated key** to the clipboard or another location for reference in the next section



- Navigate to <https://github.com/SLS-ALL/spring-petclinic>
- Click Fork on the top right of the screen

A sample Spring-based application

429 commits | 6 branches | 0 releases | 17 contributors

Branch: master | New pull request | Create new file | Upload files | Find File | Clone or download

This branch is 113 commits ahead, 330 commits behind spring-projects:master.

**kontostathis Merge pull request #2 from kingsman142/master** | Latest commit 38a3a41 on Dec 12, 2017

File	Description	Time Ago
cookbooks	removed 'bash' resource from petclinic cookbook - added in a template...	4 years ago
environments	removed 'bash' resource from petclinic cookbook - added in a template...	4 years ago
src	Set for demo - Jenkins-	2 years ago
.gitignore	added .DS_Store to .gitignore	2 years ago
springBeans	using latest versions of hibernate, spring-data, joda...	5 years ago
temp	Added empty file.	4 years ago
Vagrantfile	Commented out chef provisioning. Also, added ubuntu/trusty64 box for ...	4 years ago
deploy.yml	added firewall rule for 8080/tcp to deploy.yml	2 years ago
pom.xml	Missed a section when commenting out.	4 years ago
pom_provision_demo.xml	minimized shell scripting demo. updated readme	4 years ago
provision.sh	Parallelized server setup, added option to modify security groups, an...	a year ago
readme.md	Updated readme to indicate where to download resources to usbstick.	4 years ago
solo.json	removed 'bash' resource from petclinic cookbook - added in a template...	4 years ago
solo.rb	minimized shell scripting demo. updated readme	4 years ago

- Notice that the name at the top of the page has changed to your username and indicated the source of the fork operation

A sample Spring-based application

429 commits | 6 branches | 0 releases | 17 contributors

Branch: master | New pull request | Create new file | Upload files | Find File | Clone or download

This branch is even with SLS-ALL:master.

**kontostathis Merge pull request SLS-ALL#2 from kingsman142/master** | Latest commit 38a3a41 on Dec 12, 2017

File	Description	Time Ago
cookbooks	removed 'bash' resource from petclinic cookbook - added in a template...	4 years ago
environments	removed 'bash' resource from petclinic cookbook - added in a template...	4 years ago
src	Set for demo - Jenkins-	2 years ago
.gitignore	added .DS_Store to .gitignore	2 years ago
springBeans	using latest versions of hibernate, spring-data, joda...	5 years ago
temp	Added empty file.	4 years ago
Vagrantfile	Commented out chef provisioning. Also, added ubuntu/trusty64 box for ...	4 years ago
deploy.yml	added firewall rule for 8080/tcp to deploy.yml	2 years ago
pom.xml	Missed a section when commenting out.	4 years ago
pom_provision_demo.xml	minimized shell scripting demo. updated readme	4 years ago
provision.sh	Parallelized server setup, added option to modify security groups, an...	a year ago
readme.md	Updated readme to indicate where to download resources to usbstick.	4 years ago
solo.json	removed 'bash' resource from petclinic cookbook - added in a template...	4 years ago

- Close Github

## Gitlab Setup

- From the Task view in AWS ECS, find the public IP of your Gitlab Task
- Change your password when prompted (at least 8 characters long)

The screenshot shows the GitLab Community Edition homepage. On the right, a modal window titled "Change your password" is open. It displays a single error message: "1 error prohibited this user from being saved: Password is too short (minimum is 8 characters)". Below the error message are two input fields: "New password" containing "\*\*\*\*\*" and "Confirm new password" also containing "\*\*\*\*\*". A blue "Change your password" button is at the bottom. At the bottom of the page, there are links for "Request a new one" and "Sign in".

- Login with the username `root` and the password you just set

The screenshot shows the GitLab Community Edition login page. A red banner at the top says "Invalid Login or password.". The main form has "Sign in" and "Register" buttons. The "Username or email" field contains "root" and the "Password" field contains "\*\*\*\*\*". There are "Remember me" and "Forgot your password?" links, and a green "Sign in" button at the bottom.

- Add `spring-petclinic` project: On GitLab dashboard, click 'new project' or 'Create a Project'
- Select the Import Project tab
- Click 'import project from github'

The screenshot shows the 'New project' page on GitLab. In the top navigation bar, there are links for 'Projects', 'Groups', 'Activity', 'Milestones', 'Snippets', and a search bar. Below the navigation, there are three main options: 'Blank project', 'Create from template', and 'Import project'. Under 'Import project', there is a sub-section titled 'Import project from' with several options: 'GitLab export' (selected), 'GitHub', 'Bitbucket Cloud', 'Bitbucket Server', 'GitLab.com', 'Google Code', 'Fogbugz', 'Gitea', 'git Repo by URL', and 'Manifest file'. A tip at the bottom suggests creating a project from the command line.

- Enter personal access token (created above) and List Github Repositories
- Click 'import' next to 'spring-petclinic' to import then click Go To Project

The screenshot shows the 'Import repositories from GitHub' page. The URL in the address bar is '192.168.153.150/import/github/new'. The page displays a table with one row, showing the import status for the repository 'morley/spring-petclinic' from GitHub to GitLab. The status is 'Done' with a green circular icon. There is also a 'Go to project' button.

From GitHub	To GitLab	Status
<a href="#">morley/spring-petclinic</a>	<a href="#">root/spring-petclinic</a>	Done

- From the clone drop down on the top right, select the copy button next to the HTTP address

You won't be able to pull or push project code via SSH until you [add an SSH key](#) to your profile

The Auto DevOps pipeline has been enabled and will be used if no alternative CI configuration file is found. [More information](#)

**spring-petclinic** Project ID: 1

Add license 429 Commits 6 Branches 0 Tags 0 Bytes Files

A sample Spring-based application

master spring-petclinic / +

Merge pull request #2 from kingsman142/master ... kontostathisk authored 1 year ago

Unverified 3833416

README Add CHANGELOG Add CONTRIBUTING Auto DevOps enabled Add Kubernetes cluster

Name	Last commit	Last update
cookbooks	removed 'bash' resource from petclinic cookbook ...	4 years ago
environments	removed 'bash' resource from petclinic cookbook ...	4 years ago
src	Set for demo - Jenkins-	1 year ago
.gitignore	added .DS_Store to .gitignore	1 year ago
.springBeans	using latest versions of hibernate, spring-data, jo...	4 years ago
.temp	Added empty file.	4 years ago
Untracked files	Commented out chef resources in spec_helper.rb	2 years ago

## IMPORTANT

- If the http url does not contain a proper ip address following the `http://`, paste the copied url somewhere that allows editing and replace the number string following `http://` and preceding `/root...` with the ip of your gitlab instance. The copy this new url for use in the next step

## Create Maven Deployment in Jenkins

- On the Jenkins main page, from the left sidebar, select New Item

New Item

People

Build History

-  Project Relationship
-  Check File Fingerprint
-  Manage Jenkins
-  My Views
-  Lockable Resources
-  Credentials
-  New View

## Build Queue

No builds in the queue.

## Build Executor Status

1 Idle

2 Idle

- When prompted enter the Item Name `petclinic` and select Maven project and press OK

**Enter an item name**

» Required field

---

**Freestyle project**



This is the central feature of Jenkins. Jenkins will build your project, combining any SCM with any build system, and this can be even used for something other than software build.

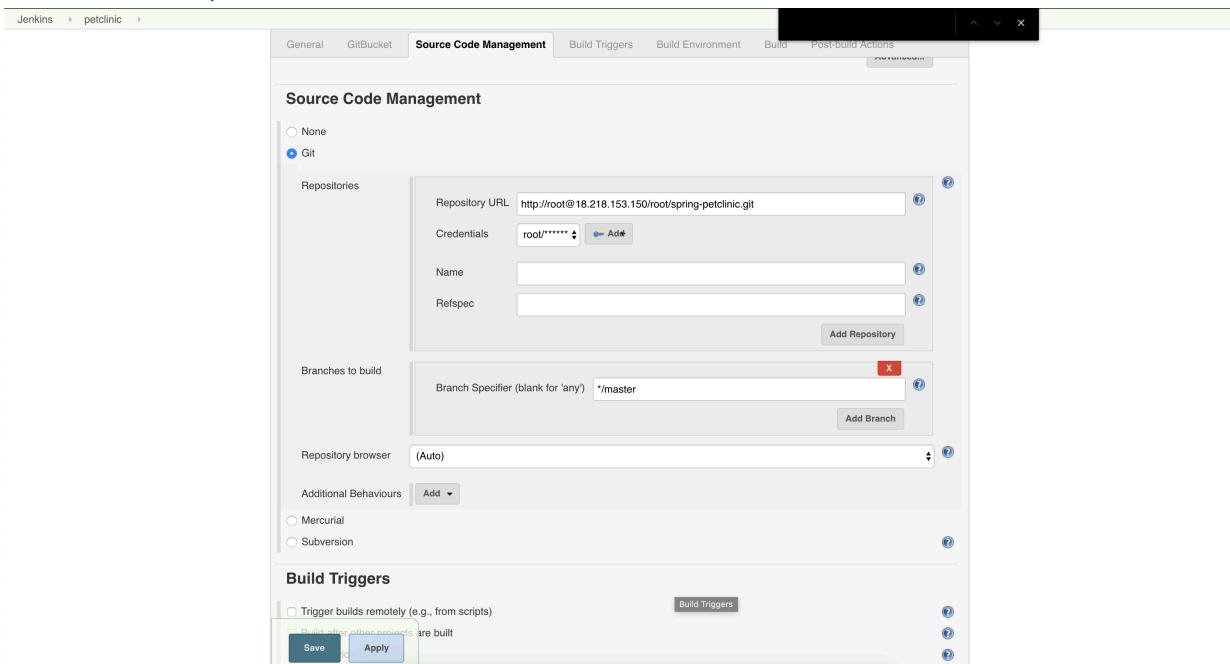
---

**Maven project**



Build a maven project. Jenkins takes advantage of your POM files and drastically reduces the configuration.

- Under Source Code Management, select 'git'
- Beside Credentials, click Add -> Jenkins
- Select "Username with password"
- Enter your GitLab credentials (see 'gitlab' VM instructions above) and click Add
- Enter repository URL: [http://\[username@gitlab VM private network IP\]/spring-petclinic.git](http://[username@gitlab VM private network IP]/spring-petclinic.git)
  - NOTE: this is the HTTP URL from the GitLab project page where 'localhost' is replaced by the 'gitlab' VM's private network IP (ex: <http://root@10.1.1.3/root/spring-petclinic.git>)
- Select appropriate credentials
- Leave the default build Root Pom pointing to pom.xml
- Select APPLY, the SAVE



- From the project page in Jenkins select Build Now to test building

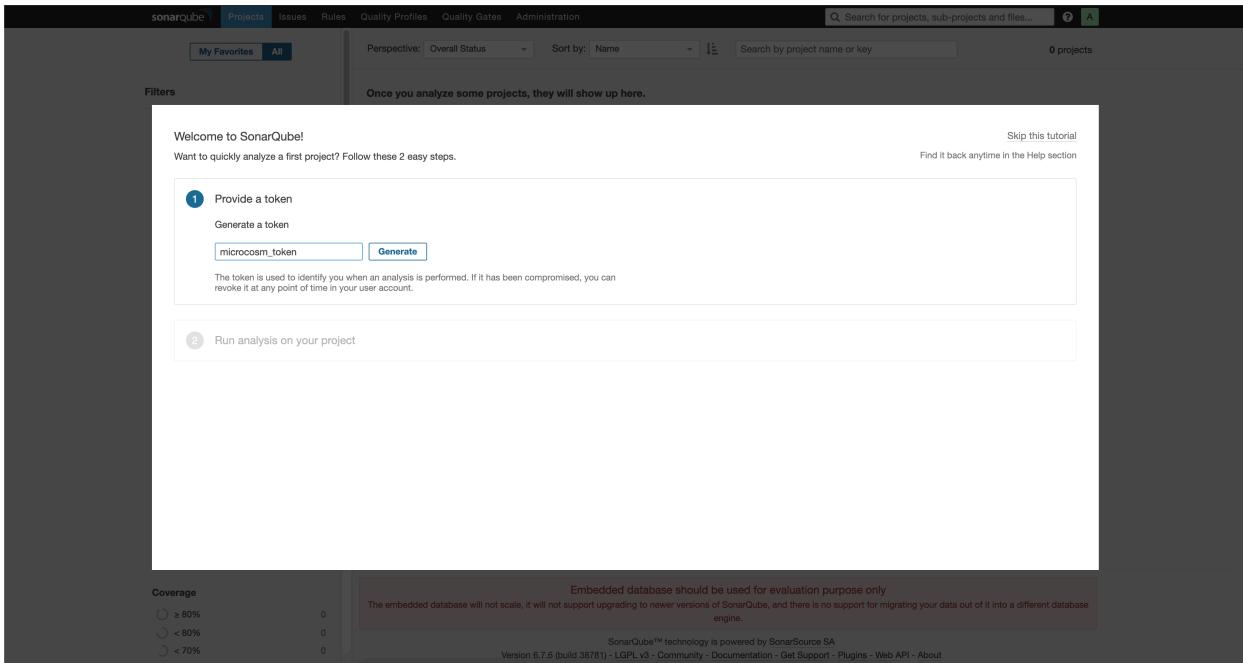
The screenshot shows the Jenkins dashboard for the Maven project 'petclinic2'. On the left, there's a sidebar with links like 'Back to Dashboard', 'Status', 'Changes', 'Workspace', 'Build Now', 'Delete Maven project', 'Configure', 'Modules', 'SonarQube', 'Rename', and 'Embeddable Build Status'. The main area is titled 'Maven project petclinic2' and shows a 'SonarQube' icon with three cards: 'Workspace', 'Recent Changes', and 'SonarQube Quality Gate'. Below that is a 'Permalinks' section with a 'Build History' table showing two builds: '#2' (Apr 17, 2019 5:22 PM) and '#1' (Apr 17, 2019 5:13 PM). At the bottom, there are 'RSS for all' and 'RSS for failures' links. The footer of the page includes a timestamp ('Page generated: Apr 17, 2019 5:23:17 PM UTC'), a 'REST API' link, and 'Jenkins ver. 2.164.1'.

## Sonarqube Setup

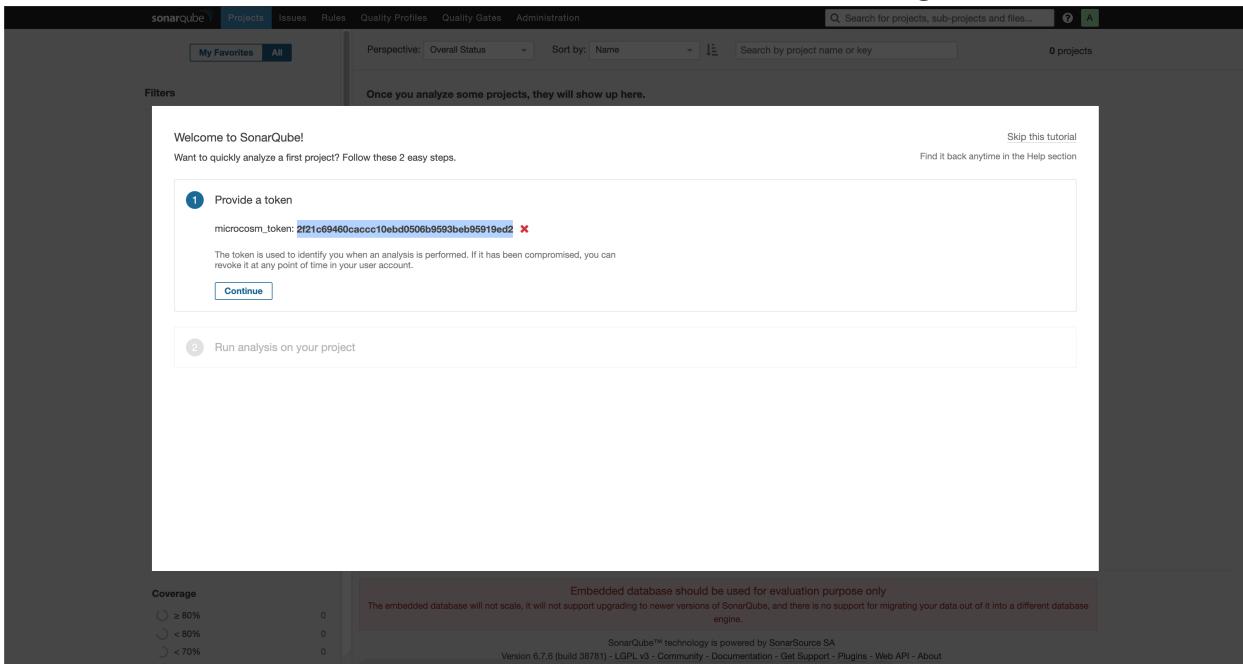
- Using the IP from the task view in AWS ECS, navigate to the sonarqube service in the browser at port 9000 (eg 44.44.44.44:9000)
- Login with the default username and password admin:admin

The screenshot shows the SonarQube interface. At the top, there's a navigation bar with 'sonarqube' logo, 'Projects', 'Issues', 'Rules', 'Quality Profiles', 'Quality Gates', a search bar, and 'Log in'. The main dashboard has a 'Continuous Code Quality' section with a 'Log in' button, a 'Read documentation' button, and a summary: '0 Projects Analyzed' with '0 Bugs', '0 Vulnerabilities', and '0 Code Smells'. Below this is a 'Multi-Language' section listing supported languages: Java, C/C++, C#, COBOL, ABAP, HTML, RPG, JavaScript, TypeScript, Objective C, XML, VB.NET, PL/SQL, T-SQL, Flex, Python, Groovy, PHP, Swift, Visual Basic, and PL/I. The 'Quality Model' section defines 'Bugs', 'Vulnerabilities', and 'Code Smells'. The 'Write Clean Code' section discusses maintaining a clean code base and fixing new issues. The 'Fix The Leak' section explains the leak paradigm and its configuration. A large watermark for 'SonarQube' is visible across the bottom of the dashboard.

- Enter a name for you token generation



- Copy the generated token and press Continue
- Select Java, then Maven when prompted.
- Press Finish Tutorial button or Skip this Tutorial in the top right corner



- Return to Jenkins
- From Manage Jenkins -> Under Configure System, scroll down to SonarQube Servers

- Select Add SonarQube
- Enter the name SonarQube
- Paste your server authentication token
- Enter the url including port of your SonarQube server
- Press APPLY, then SAVE

The screenshot shows the Jenkins Global Tool Configuration page for SonarQube installations. The 'Name' field is set to 'SonarQube'. The 'Server URL' field contains 'http://18.222.11.148:9000'. The 'Save' and 'Apply' buttons are visible at the bottom.

Setting	Value
Name	SonarQube
Server URL	http://18.222.11.148:9000

- Go to Manage Jenkins -> Global Tool Configuration
- Under SonarQube Scanner, Select Add SonarQube Scanner
- Enter "SonarQube" in the "Name" field
- Check "Install automatically"
- Choose the most recent version of SonarQube Scanner from the version dropdown
- Click Apply and Save

The screenshot shows the Jenkins Global Tool Configuration page. In the 'SonarScanner for MSBuild' section, a dropdown menu is open, listing various versions of the SonarCube Scanner. The selected version is 'SonarCube Scanner 3.3.0.1492'. Other listed versions include 3.2.0.1227, 3.1.0.1141, 3.0.3.778, 3.0.2.768, 3.0.1.733, 3.0.0.702, 2.9.0.670, 2.8, 2.7, 2.6.1, 2.6, 2.5.1, 2.5, 2.4, 2.3, 2.2.2, 2.2.1, 2.2, 2.1, 2.0, 1.4, 1.3, 1.2, 1.1, and 1.0. Buttons for 'Save', 'Apply', 'Delete Installer', and 'Delete SonarCube Scanner' are visible at the bottom.

- Return to the PetClinic configuration screen
- Scroll down to the Post Steps Section
- Click "Add build step" and select "Execute SonarQube Scanner"

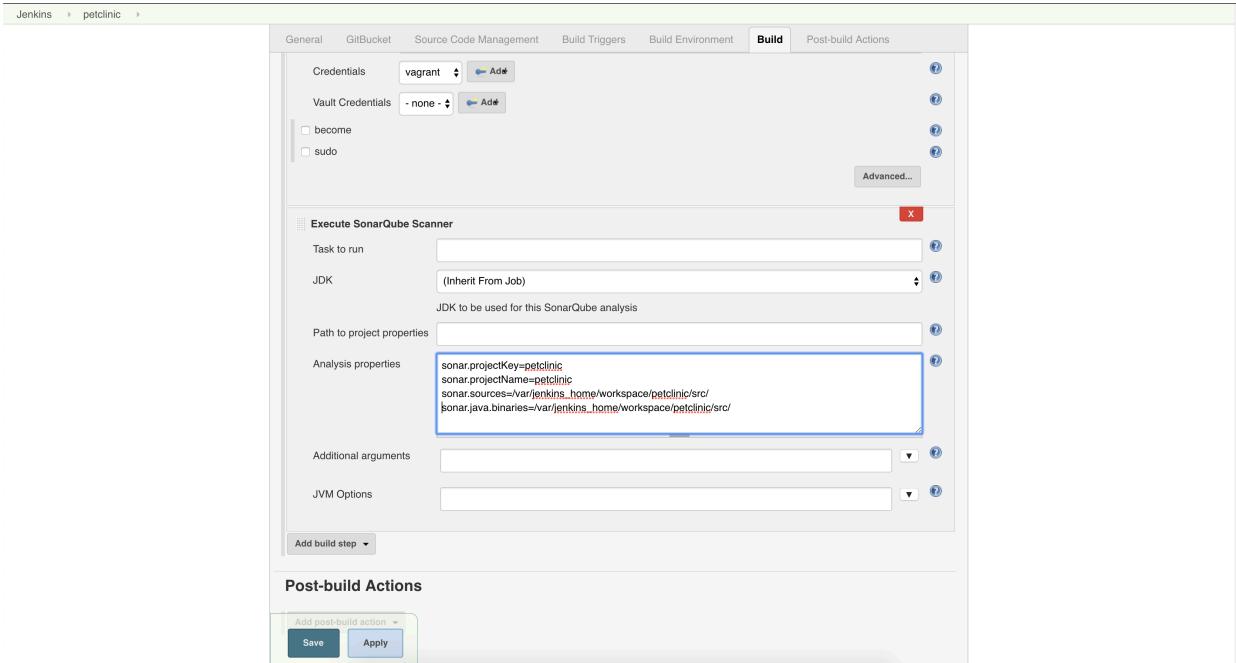
The screenshot shows the Jenkins PetClinic configuration screen under the 'Build' tab. A dropdown menu is open, showing various build steps. The 'Execute SonarQube Scanner' option is highlighted with a blue selection bar. Other options in the list include 'Conditional step (single)', 'Associate Tag (Nexus Repository Manager 3.x)', 'Conditional steps (multiple)', 'Copy artifacts from another project', 'Create Tag (Nexus Repository Manager 3.x)', 'Delete Components (Nexus Repository Manager 3.x)', 'Docker Build and Publish', 'Invoke Ansible Playbook', and 'Execute shell'. The 'Invoke Ansible Playbook' section contains fields for 'Playbook path' (set to 'deploy.yml'), 'Inventory' (set to 'File or host list' with '/etc/ansible/hosts' in the file path field), 'Host subset', 'Credentials' (set to 'vagrant'), 'Vault Credentials', and checkboxes for 'become' and 'sudo'. A 'Advanced...' button is also present.

- Under "Analysis properties" enter:

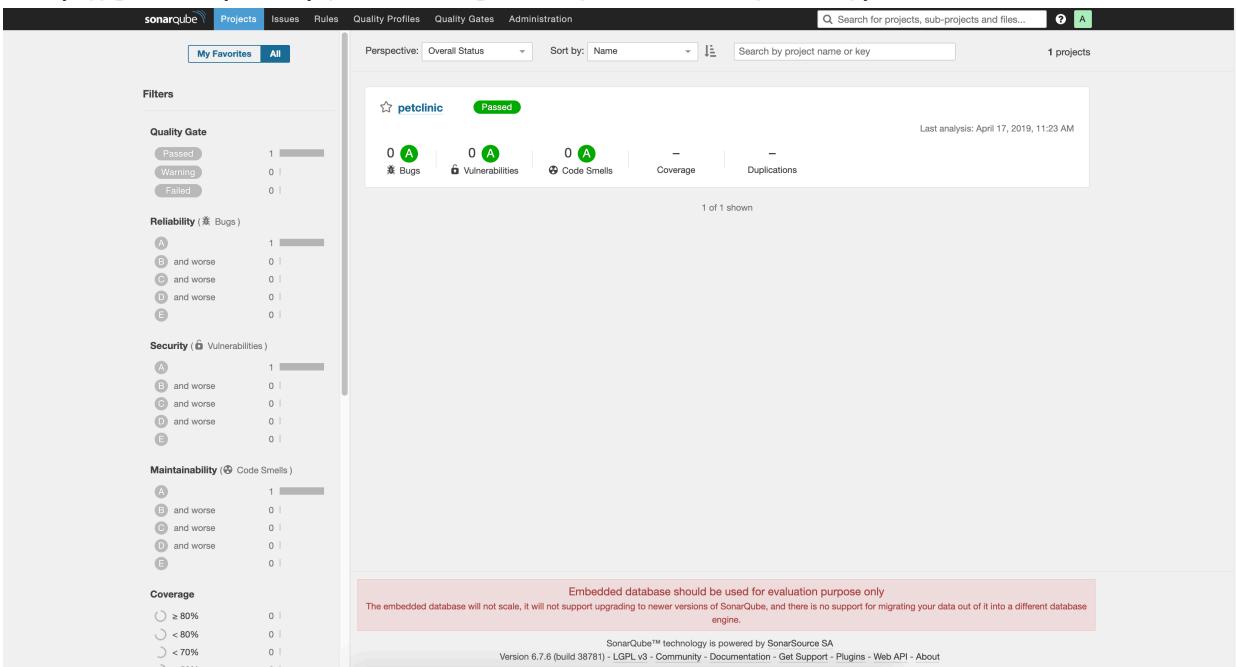
```
sonar.projectKey=petclinic
```

```
sonar.projectName=petclinic
sonar.sources=/var/jenkins_home/workspace/petclinic/src/
sonar.java.binaries=/var/jenkins_home/workspace/petclinic/src/
```

- Click Apply and Save



- On the petclinic project page select Build Now from the left sidebar
- After a successful build, the static code analysis will be available at "http://[sonarqube ip|localhost]:9000/dashboard/index/petclinic"



## AWS EC2 Based Code Deploy

Ref. [Youtube walkthrough example](#)

### Create IAM Role for EC2 Based CodeDeploy

Create a role called `CodeDeployRole`

- From the IAM Console, select Roles from the left sidebar then Create Role

The screenshot shows the AWS IAM Roles page. On the left, there is a sidebar with a 'Search IAM' input field and a navigation menu with the following items: Dashboard, Groups, Users, **Roles** (which is selected), Policies, Identity providers, Account settings, Credential report, and Encryption keys. Below the sidebar, the main content area has a title 'Roles' and a section titled 'What are IAM roles?'. It explains that IAM roles are a secure way to grant permissions to entities that you trust. Examples include IAM users in another account, application code running on an EC2 instance, AWS services, and corporate directory users. It also notes that IAM roles issue short-lived keys for secure access. Below this, there is a section titled 'Additional resources:' with links to 'IAM Roles FAQ', 'IAM Roles Documentation', 'Tutorial: Setting Up Cross Account Access', and 'Common Scenarios for Roles'. At the bottom of the main content area, there are two buttons: 'Create role' (in blue) and 'Delete role'.

- On the first page of the wizard, select AWS Service and CodeDeploy

# Create role

## Select type of trusted entity



### AWS service

EC2, Lambda and others



### Another AWS

Belonging to you

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

## Choose the service that will use this role

### EC2

Allows EC2 instances to call AWS services on your behalf.

### Lambda

Allows Lambda functions to call AWS services on your behalf.

### API Gateway

### CodeDeploy

- Select CodeDeploy as your Use Case at the bottom of the page and press Next

Select your use case

#### CodeDeploy

Allows CodeDeploy to call AWS services such as Auto Scaling on your behalf.

#### CodeDeploy - ECS

Allows CodeDeploy to read S3 objects, invoke Lambda functions, publish to SNS topics, and update ECS services on your behalf.

#### CodeDeploy for Lambda

Allows CodeDeploy to route traffic to a new version of an AWS Lambda function version on your behalf.

- Press Next

## Create role

1 2 3 4

### Attached permissions policies

The type of role that you selected requires the following policy.

Filter policies ▾			Showing 1 result
Policy name ▾	Used as	Description	
▶  AWSCodeDeployRole	None	Provides CodeDeploy service access to expand...	

- Add Tags if desired and press Next

## Create role

1 2 3 4

### Add tags (optional)

IAM tags are key-value pairs you can add to your role. Tags can include user information, such as an email address, or can be descriptive, such as a job title. You can use the tags to organize, track, or control access for this role. [Learn more](#)

Key	Value (optional)	Remove
Name	CodeDeploy Jenkins Pipeline Usage	x

- Assign your role a name (e.g. `CodeDeployRole`), then press Create Role
- For our purposes in this exercise, we are going to add additional permissions to our role. From the list of roles, select the role you created. Then, press Attach policies on the following screen.

Maximum CLI/API session duration 1 hour [Edit](#)

Permissions Trust relationships Tags (1) Access Advisor Revoke sessions

### Permissions policies (1 policy applied)

[Attach policies](#)

Policy name ▾
▶  AWSCodeDeployRole

- Search for, and select the following policies. Once all are selected press Attach Policy. (Note that there is a 10 policy limit per role)
  - `AmazonEC2FullAccess`
  - `AutoScalingFullAccess`
  - `AWSDataLifecycleManagerServiceRole`

- ElasticLoadBalancingFullAccess
- AmazonS3FullAccess
- CloudWatchLogsFullAccess
- By way of auditing the use of the role, and to verify that you haven't assigned more access than is necessary, selecting the Access Advisor tab after selecting the desired role, will display what access has actually been used by the role.

[Roles](#) > [CodeDeployServiceRole](#)

## Summary

Role ARN	arn:aws:iam::443007076818:role/CodeDeployServiceRole <a href="#">Edit</a>
Role description	Allows CodeDeploy to read S3 objects, invoke Lambda functions, publish to SNS topics, and update ECS services on your behalf.
Instance Profile ARNs	<a href="#">Edit</a>
Path	/
Creation time	2019-04-23 13:29 MDT
Maximum CLI/API session duration	1 hour <a href="#">Edit</a>

[Permissions](#) [Trust relationships](#) [Tags](#) [Access Advisor](#) [Revoke sessions](#)

Access advisor shows the service permissions granted to this role and when those services were last accessed. You can use this information to revise your policies. [Learn more](#)

Note: Recent activity usually appears within 4 hours. Data is stored for a maximum of 365 days, depending when your region began supporting this feature. [Learn more](#)

Filter: <a href="#">No filter</a> <a href="#">Search</a>		
Service Name	Policies Granting Permissions	Last Accessed
Elastic Load Balancing	AutoScalingFullAccess and 5 more	Today
Amazon EC2	AutoScalingFullAccess and 5 more	Today
Amazon Elastic Container Service	AdministratorAccess and 2 more	Yesterday
Amazon S3	AdministratorAccess and 2 more	Yesterday
AWS Lambda	AdministratorAccess and 2 more	Yesterday
Alexa for Business	AdministratorAccess	Not accessed in the tracking period
AWS Accounts	AdministratorAccess	Not accessed in the tracking period
AWS Certificate Manager	AdministratorAccess	Not accessed in the tracking period

## Create EC2 Artifacts

To deploy our built code, we need a place to deploy it. To that end, we are going to create a load balanced deployment with two machine instances so that 1. traffic can be evenly distributed to our deployment and 2. so that when deploying our product, we have less down time as our instances are updated in sequence.

### Amazon Machine Image (AMI)

- From the EC2 Console, select Instances from the left sidebar. Then Select Launch Instance.
- Given that our application is a Java Web application, in search bar, search for Tomcat. From the AWS Marketplace select the Tomcat Certified by Bitnami

image.

#### Step 1: Choose an Amazon Machine Image (AMI)

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

- Press Continue on the pricing details screen (notice that the instance is free to license)

Instance Type	Software	EC2	Total
t2.micro	\$0.00	\$0.012	<b>\$0.012/hr</b>
t2.small	\$0.00	\$0.023	<b>\$0.023/hr</b>
t2.medium	\$0.00	\$0.046	<b>\$0.046/hr</b>
t2.large	\$0.00	\$0.093	<b>\$0.093/hr</b>
t2.xlarge	\$0.00	\$0.186	<b>\$0.186/hr</b>
t2.2xlarge	\$0.00	\$0.371	<b>\$0.371/hr</b>
i3.micro	\$0.00	\$0.01	<b>\$0.01/hr</b>
i3.small	\$0.00	\$0.021	<b>\$0.021/hr</b>
i3.medium	\$0.00	\$0.042	<b>\$0.042/hr</b>
i3.large	\$0.00	\$0.083	<b>\$0.083/hr</b>
i3.xlarge	\$0.00	\$0.166	<b>\$0.166/hr</b>
m3.2xlarge	\$0.00	\$0.333	<b>\$0.333/hr</b>
m4.large	\$0.00	\$0.10	<b>\$0.10/hr</b>
m4.xlarge	\$0.00	\$0.20	<b>\$0.20/hr</b>
m4.2xlarge	\$0.00	\$0.40	<b>\$0.40/hr</b>
m4.4xlarge	\$0.00	\$0.80	<b>\$0.80/hr</b>
m4.10xlarge	\$0.00	\$2.00	<b>\$2.00/hr</b>
m4.16xlarge	\$0.00	\$3.20	<b>\$3.20/hr</b>
c5.large	\$0.00	\$0.085	<b>\$0.085/hr</b>
c5.xlarge	\$0.00	\$0.17	<b>\$0.17/hr</b>
c5.2xlarge	\$0.00	\$0.34	<b>\$0.34/hr</b>

- In Step 2, select the size of the VM desired - the smaller the instance, the cheaper it is to run. For this exercise, select t2.micro or t2.small.

## 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 v [Learn more](#) about instance types and how they can meet your computing needs.

Filter by: All instance types ▾ Current generation ▾ Show/Hide Columns

Currently selected: t2.large (Variable ECUs, 2 vCPUs, 2.3 GHz, Intel Broadwell E5-2686v4, 8 GiB memory, EBS only)

	Family	Type	vCPUs ⓘ	Memory (GiB)
<input type="checkbox"/>	General purpose	t2.nano	1	0.5
<input type="checkbox"/>	General purpose	t2.micro Free tier eligible	1	1
<input type="checkbox"/>	General purpose	t2.small	1	2
<input type="checkbox"/>	General purpose	t2.medium	2	4
<input checked="" type="checkbox"/>	General purpose	t2.large	2	8
<input type="checkbox"/>	General purpose	t2.xlarge	4	16
<input type="checkbox"/>	General purpose	t2.2xlarge	8	32

- In Step 3, do the following
  - Number of Instance: 2
  - Network: Select the VPC that matches your VPC ID captured after creating a cluster
  - Subnet: Select either subnet
  - Auto-Assign public IP: Select Enable (or leave the default if it is Enable)
  - IAM Role: Select EC2PlusS3RoleTST
    - Under Advanced Detail, copy and paste the contents of the `addCodeDeploytoEC2Image.sh` file in the optional text box. Make sure the As Text radio button is selected. The purpose of this script is to add the libraries and services necessary to the Ubuntu based VM to be able to deploy to the machine in an automated fashion. Note that the script uses the AWS EAST-2 region (Ohio), change this portion of the S3 url for the region you are using. Additionally, this script installs the services necessary to forward logs to CloudWatch to be used for monitoring and debugging deployments, and program runs. [ref.](#)
- Press Next

**Step 3: Configure Instance Details**

**Subnet**: subnet-04d355bb02dc06049 | tst1 | us-east-2a  
248 IP Addresses available [Create new subnet](#)

**Auto-assign Public IP**: Use subnet setting (Enable)

**Placement group**: Add instance to placement group

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

**IAM role**: EC2\_Plus\_S3 [Create new IAM role](#)

**Shutdown behavior**: Stop

**Enable termination protection**: Protect against accidental termination

**Monitoring**: Enable CloudWatch detailed monitoring  
Additional charges apply.

**Tenancy**: Shared - Run a shared hardware instance  
Additional charges will apply for dedicated tenancy.

**Elastic Inference**: Add an Elastic Inference accelerator  
Additional charges apply.

**T2/T3 Unlimited**: Enable  
Additional charges may apply

**Network interfaces**

Device	Network Interface	Subnet	Primary IP	Secondary IP addresses	IPv6 IPs
eth0	New network interface	subnet-04d355bb02dc06049	Auto-assign	Add IP	Add IP

[Add Device](#)

**Advanced Details**

**User data**: As text  As file  Input is already base64 encoded  
(Optional)

[Feedback](#) [English \(US\)](#)

- Press Next on Step 4, using the default storage.
- In Step 5, add a tag with the Key = Name, and Value = Code\_Deploy\_Instances

[1. Choose AMI](#) [2. Choose Instance Type](#) [3. Configure Instance](#) [4. Add Storage](#) [5. Add Tags](#) [6. Configure Security Group](#) [7. Review](#)

#### Step 5: Add Tags

A tag consists of a case-sensitive key-value pair. For example, you could define a tag with key = Name and value = 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)	Instances	Volumes
Name		Code_Deploy_Instances		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
<a href="#">Add another tag</a> (Up to 50 tags maximum)					

- Step 6, Select an Existing Security Group, then select your student Security group.
- Select Review and Launch.

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

### Step 6: Configure Security Group

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

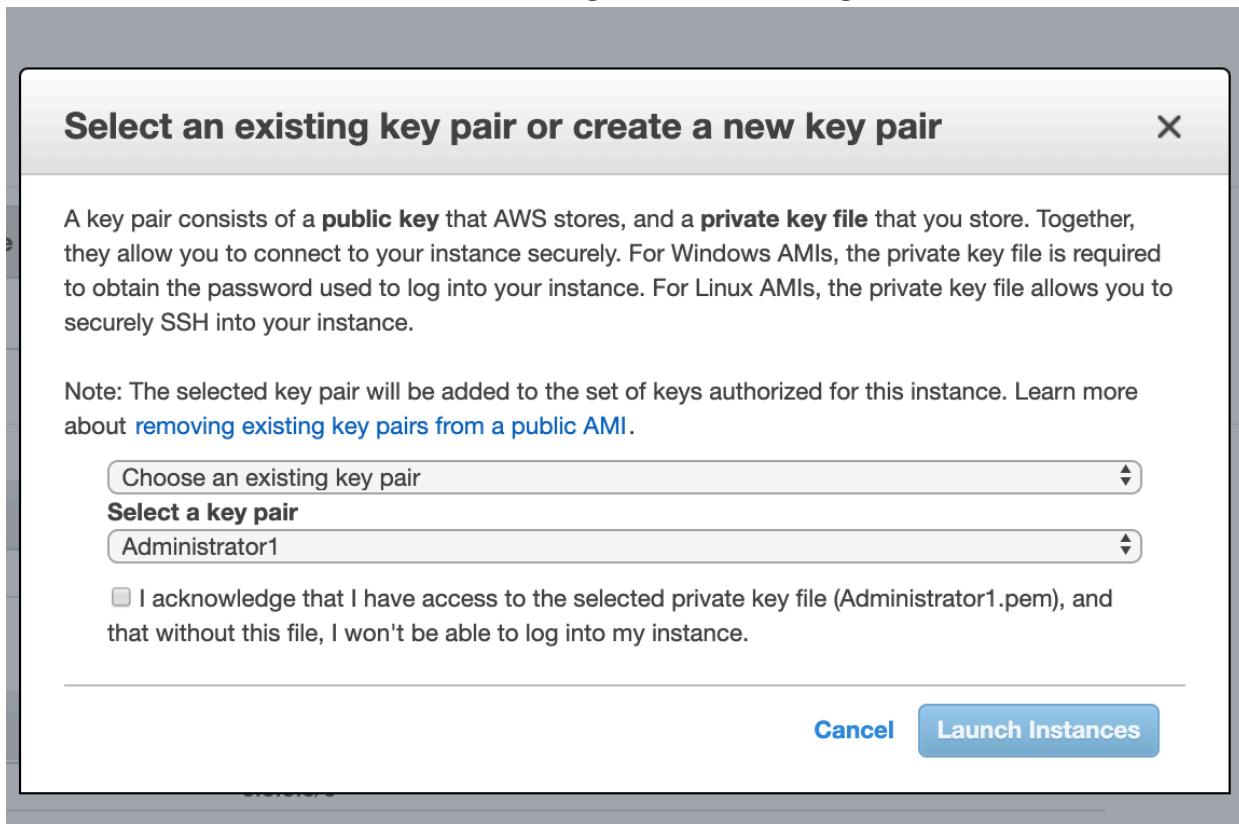
Assign a security group:  Create a new security group  
 Select an existing security group

Security Group ID	Name	Description
sg-0e1a263967d2471f2	default	default VPC security group

Inbound rules for sg-0e1a263967d2471f2 (Selected security groups: sg-0e1a263967d2471f2)

Type	Protocol	Port Range	Source
Custom TCP Rule	TCP	8087	0.0.0.0/0
Custom TCP Rule	TCP	8087	::/0
HTTP	TCP	80	0.0.0.0/0
HTTP	TCP	80	::/0
Custom TCP Rule	TCP	8080	0.0.0.0/0
Custom TCP Rule	TCP	8080	::/0
SSH	TCP	22	0.0.0.0/0
HTTPS	TCP	443	0.0.0.0/0
HTTPS	TCP	443	::/0

- When prompted to choose an SSH key, select an existing pair if you already have one, otherwise follow the wizard to create one (make sure to save the private key in a safe and memorable location).
- Press Launch Instances after selecting the acknowledgement box



- Review the instances created and make note of the IP addresses

Name	Instance ID	Instance Type	Availability Zone	Instance State	Status Checks	Alarm Status	Public DNS (IPv4)	IPv4 Public IP	IPv6 IPs	Key Name	Monitoring	Launch Time	Security Groups
Code_Deplo...	i-03a155360d9fb37b5	t2.large	us-east-2a	running	Initializing	None	ec2-18-218-184-15.eca...	18.218.184.15	-	Administrator1	disabled	April 26, 2019 at 9:14:08 AM...	default
Code_Deplo...	i-0e2a50cb45d6bd604	t2.large	us-east-2a	running	Initializing	None	ec2-18-222-113-20.eca...	18.222.113.20	-	Administrator1	disabled	April 26, 2019 at 9:14:08 AM...	default

## Load Balancer

- From the left sidebar of the EC2 Console, select Load Balancers. Press Create Load Balancer.
- Step 1, give your load balancer a name and select your VPC and select a subnet from each availability zone present (there must be at least 2 AZs for the load balancer to work)
- Press Next

1. Configure Load Balancer    2. Configure Security Settings    3. Configure Security Groups    4. Configure Routing    5. Register Targets    6. Review

### Step 1: Configure Load Balancer

#### Basic Configuration

To configure your load balancer, provide a name, select a scheme, specify one or more listeners, and select a network. The default configuration is an Internet-facing load balancer in the

Name	<input type="text" value="ec2betclinic"/>	<small>⚠ Load balancer names must only contain alphanumeric characters or hyphens, and not start with a hyphen.</small>
Scheme	<input checked="" type="radio"/> internet-facing <input type="radio"/> internal	
IP address type	<input type="text" value="ipv4"/>	

#### Listeners

A listener is a process that checks for connection requests, using the protocol and port that you configured.

Load Balancer Protocol	Load Balancer Port
<input type="text" value="HTTP"/>	<input type="text" value="8080"/>
<input type="button" value="Add listener"/>	

#### Availability Zones

Specify the Availability Zones to enable for your load balancer. The load balancer routes traffic to the targets in these Availability Zones only. You can specify only one subnet per Availability Zone.

VPC	<input type="text" value="vpc-0eb311ea035a4eb9c (10.0.0.0/16)   tst"/>
Availability Zones	<input checked="" type="checkbox"/> us-east-2a <input type="text" value="subnet-04d355bb02dc06049 (tst1)"/>
IPv4 address	<small>Assigned by AWS</small>
<input checked="" type="checkbox"/> us-east-2c	<input type="text" value="subnet-0c9b55ecae32d1073 (tst2)"/>
IPv4 address	<small>Assigned by AWS</small>

▶ Tags

- Press Next for step 2

1. Configure Load Balancer    2. Configure Security Settings    3. Configure Security Groups    4. Configure Routing    5. Register Targets    6. Review

## Step 2: Configure Security Settings



**Improve your load balancer's security. Your load balancer is not using any secure listener.**

If your traffic to the load balancer needs to be secure, use the HTTPS protocol for your front-end connection. You can go back to the first step to change your load balancer's port or protocol.

- Step 3, select an existing security, and select your Student security group

1. Configure Load Balancer    2. Configure Security Settings    3. Configure Security Groups    4. Configure Routing    5. Register Targets    6. Review

### Step 3: Configure Security Groups

A security group is a set of firewall rules that control the traffic to your load balancer. On this page, you can add rules to allow specific traffic to reach your load balancer. First, decide whether to create a new security group or select an existing one.

Assign a security group:

- Create a new security group
- Select an existing security group

Security Group ID	Name	Description
sg-0e1a263967d2471f	default	default VPC security group

- Step 4, Create a new Target Group
  - Name: create a memorable name (e.g. StudentNameCodeDeployTG80)
  - Port: 80
  - Path: /
  - Advanced Health Check Settings:
    - Healthy threshold: 2
    - Unhealthy threshold: 10
    - Timeout: 20
- Press Next

[1. Configure Load Balancer](#)[2. Configure Security Settings](#)[3. Configure Security Groups](#)[4. Configure Routing](#)

## Step 4: Configure Routing

Your load balancer routes requests to the targets in this target group using the protocol and port that you specify.

### Target group

**Target group** ⓘ[New target group](#) ▾**Name** ⓘ

TGcodedeployec2petclinic

⚠ TargetGroup name cannot contain characters that are not letters, numbers, or underscores.

**Target type** Instance IP Lambda function**Protocol** ⓘ[HTTP](#) ▾**Port** ⓘ

8080

### Health checks

**Protocol** ⓘ[HTTP](#) ▾**Path** ⓘ

/petclinic

#### ▼ Advanced health check settings

**Port** ⓘ traffic port override**Healthy threshold** ⓘ

5

**Unhealthy threshold** ⓘ

2

**Timeout** ⓘ

5

seconds

**Interval** ⓘ

30

seconds

**Success codes** ⓘ

200

- Step 5, Select both instances of your AMI, press Add to Registered on port 80
- Press Next

[1. Configure Load Balancer](#) [2. Configure Security Settings](#) [3. Configure Security Groups](#) [4. Configure Routing](#) [5. Register Targets](#) [6. Review](#)

### Step 5: Register Targets

Register targets with your target group. If you register a target in an enabled Availability Zone, the load balancer starts routing requests to the targets as soon as the registration process completes and the target passes the initial health checks.

#### Registered targets

To deregister instances, select one or more registered instances and then click Remove.

Instances						
To register additional instances, select one or more running instances, specify a port, and then click Add. The default port is the port specified for the target group. If the instance is already registered on the specified port, you must specify a different port.						
<a href="#">Add to registered</a> on port 8080						
Instance	Name	Port	State	Security groups	Zone	
i-03a155360dafb37b5	Code_Deploy_Instances	8080	running	default	us-east-2a	
i-0e2a50cb45d6bd604	Code_Deploy_Instances	8080	running	default	us-east-2a	

Instances						
To register additional instances, select one or more running instances, specify a port, and then click Add. The default port is the port specified for the target group. If the instance is already registered on the specified port, you must specify a different port.						
<a href="#">Add to registered</a> on port 8080						
Instance	Name	State	Security groups	Zone	Subnet ID	Subnet CIDR
i-03a155360dafb37b5	Code_Deploy_Instances	running	default	us-east-2a	subnet-04d355bb02dc06049	10.0.0.0/24
i-0e2a50cb45d6bd604	Code_Deploy_Instances	running	default	us-east-2a	subnet-04d355bb02dc06049	10.0.0.0/24

- Step 6, Review and Press Create

[1. Configure Load Balancer](#) [2. Configure Security Settings](#) [3. Configure Security Groups](#) [4. Configure Routing](#) [5. Register Targets](#) [6. Review](#)

### Step 6: Review

Please review the load balancer details before continuing

#### ▼ Load balancer

**Name** ec2petclinic  
**Scheme** internet-facing  
**Listeners** Port:8080 - Protocol:HTTP  
**IP address type** ipv4  
**VPC** vpc-0eb311ea035a4eb9c (tst)  
**Subnets** subnet-04d355bb02dc06049 (tst1), subnet-0c9b55ecae32d1073 (tst2)  
**Tags**

#### ▼ Security groups

**Security groups** sg-0e1a263967d2471f2

#### ▼ Routing

**Target group** New target group  
**Target group name** TGcodedeployec2petclinic  
**Port** 8080  
**Target type** instance  
**Protocol** HTTP  
**Health check protocol** HTTP  
**Path** /petclinic  
**Health check port** traffic port  
**Healthy threshold** 5  
**Unhealthy threshold** 2  
**Timeout** 5  
**Interval** 30  
**Success codes** 200

#### ▼ Targets

**Instances** i-03a155360dafb37b5 (Code\_Deploy\_Instances):8080, i-0e2a50cb45d6bd604 (Code\_Deploy\_Instances):8080

## Create S3 Bucket

- From the Services menu at the top left, navigate to the S3 Console
- Select Create Bucket

Amazon S3 Block Public Access lets you enforce a "no public access" policy for your accounts & buckets. [Learn more »](#)

S3 buckets

S3 buckets		
<input type="text"/> Search for buckets		
<a href="#">+ Create bucket</a> <span><a href="#">Edit public access settings</a></span> <span><a href="#">Empty</a></span> <span><a href="#">Delete</a></span>		
<input type="checkbox"/>	Bucket name ▾	Access ⓘ ▾
<input type="checkbox"/>	cf-templates-1a5vqotkxzda7-us-east-2	Objects can be public
<input type="checkbox"/>	petclinicdeploy	Objects can be public
		Region ▾
		US East (Ohio)
		US East (Ohio)

- Choose a name for your bucket and your region (we are using EAST-2 (OHIO)) and proceed through the wizard (Remember the bucket name as it will be used in the Jenkins step later)
- When presented with the "Public access settings for this bucket," de-select all check boxes and change the manage option to Grant.

Create bucket

1 Name and region    2 Configure options    3 Set permissions    4 Review

Note: You can grant access to specific users after you create the bucket.

Public access settings for this bucket

Use the Amazon S3 block public access settings to enforce that buckets don't allow public access to data. You can also configure the Amazon S3 block public access settings at the account level. [Learn more](#)

**Manage public access control lists (ACLS) for this bucket**

Block new public ACLs and uploading public objects (Recommended) ⓘ  
 Remove public access granted through public ACLs (Recommended) ⓘ

**Manage public bucket policies for this bucket**

Block new public bucket policies (Recommended) ⓘ  
 Block public and cross-account access if bucket has public policies (Recommended) ⓘ

**Manage system permissions**

Grant Amazon S3 Log Delivery group write access to this bucket

## Create Codedeploy Application

In order to automate the deployment of the project, the project, an "Application" description needs to be created within AWS to be referenced by Jenkins

- From the services menu at the top of the page select CodeDeploy

- Press Create Application



- Enter a name for your application and select EC2/On-premises, then press create application

The screenshot shows the 'Create application' configuration page. At the top, a breadcrumb navigation shows 'Developer Tools > CodeDeploy > Applications > Create application'. The main title is 'Create application'. Below it is a section titled 'Application configuration'. The first field is 'Application name', with a placeholder 'Enter an application name' and a text input box containing 'petclinic'. Below this is a note '100 character limit'. The next section is 'Compute platform', with a placeholder 'Choose a compute platform' and a dropdown menu currently set to 'EC2/On-premises'. At the bottom right, there are two buttons: 'Cancel' and a prominent orange 'Create application' button.

- Enter a name for your Deployment Group and choose the Service Role created earlier
- Select In-place Deployment, then scroll down

Application

Application  
petclinic

Compute type  
EC2/On-premises

Deployment group name

Enter a deployment group name  
petclinicDepGrp  
100 character limit

Service role

Choose a service role  
Select a service role with CodeDeploy permissions that grants AWS CodeDeploy access to your target instances.  
CodeDeployServiceRole ▾

Deployment type

Choose how to deploy your application

In-place  
Updates the instances in the deployment group with the latest application revisions. During a deployment, each instance will be briefly taken offline for its update

Blue/green  
Replaces the instances in the deployment group with new instances and deploys the latest application revision to them. After instances in the replacement environment are registered with a load balancer, instances from the original environment are deregistered and can be terminated.

- Select Amazon EC2 Instances under Environment Configuration
- From the Key dropdown select Name, then from the Value dropdown select the Code\_Deploy\_Instances (the tag created when the EC2 Instances were created earlier)
- Select any of the Deployment Configurations (since our application is non-critical, selecting AllAtOnce is fine here)

## Environment configuration

Select any combination of Amazon EC2 Auto Scaling groups, Amazon EC2 instances, and on-premises instances to add to this deployment

Amazon EC2 Auto Scaling groups

Amazon EC2 instances

2 unique matched instances. [Click here for details](#)

You can add up to three groups of tags for EC2 instances to this deployment group.

**One tag group:** Any instance identified by the tag group will be deployed to.

**Multiple tag groups:** Only instances identified by all the tag groups will be deployed to.

Tag group 1

Key

Value - optional

<input type="text"/> Name	X	<input type="text"/> Code_Deploy_Instances	X	<button>Remove tag</button>
---------------------------	---	--------------------------------------------	---	-----------------------------

Add tag

+ Add tag group

On-premises instances

### Matching instances

2 unique matched instances. [Click here for details](#)

## Deployment settings

### Deployment configuration

Choose from a list of default and custom deployment configurations. A deployment configuration is a set of rules that determines how fast an application will be deployed and the success or failure conditions for a deployment.

CodeDeployDefault.OneAtATime



or

Create deployment configuration

- Enable Load Balancing and select Application Load Balancer, then the Load Balancer created previously

**Load balancer**

Select a load balancer to manage incoming traffic during the deployment process. The load balancer blocks traffic from each instance while it's being deployed to and allows traffic to it again after the deployment succeeds.

Enable load balancing

Application Load Balancer or Network Load Balancer

Classic Load Balancer

Choose a load balancer

TGcodedeployec2petclinic ▾

▼ Advanced - optional

- Under Advanced, scroll down to Rollbacks.
  - De-select Disable Rollbacks
  - Select both Rollback options
- Press Create Deployment Group

**Rollbacks**

Enable deployment rollbacks for this deployment group

Roll back when a deployment fails

Roll back when alarm thresholds are met

Disable rollbacks

Cancel **Create deployment group**

- Leave the Deployment Group Summary page open for reference in the [Modify Jenkins Build Section](#)

The screenshot shows the AWS CodeDeploy console. On the left, there's a navigation sidebar with links like 'Source', 'Build', 'Deploy', 'Applications' (which is selected), 'Deployment configurations', 'On-premises instances', and 'Pipeline'. Below that are 'Feedback' and 'Return to the old experience' buttons. The main area has a green header bar with a success icon and the text 'Success Deployment group created'. Below this, the breadcrumb trail shows 'Developer Tools > CodeDeploy > Applications > petclinic > petclinicDepGrp'. The main content area is titled 'petclinicDepGrp' and contains two sections: 'Deployment group details' and 'Environment configuration: Amazon EC2 instances'. In 'Deployment group details', fields include 'Deployment group name: petclinicDepGrp', 'Application name: petclinic', 'Compute platform: EC2/On-premises', 'Deployment type: In-place', 'Service role ARN: arn:aws:iam::443007076818:role/CodeDeployServiceRole', and 'Deployment configuration: CodeDeployDefault.OneAtATime'. In 'Environment configuration', there's a table with one row: 'Name: Code\_Deploy\_Instances'. At the bottom right are 'Edit', 'Delete', and 'Create deployment' buttons.

## Create Secret Key

A Secret key is necessary for authenticating your Jenkins server with AWS. Where possible, the use of roles or PKI is preferred for authentication, however, in this instance the use of Secret keys is acceptable.

- Navigate to the IAM Concole

### Welcome to Identity and Access Management

IAM users sign-in link:

<https://cert-microcosm.signin.aws.amazon.com/console>

### IAM Resources

Users: 2

Roles: 17

Groups: 2

Identity Providers: 1

Customer Managed Policies: 2

### Security Status

- Select your user

[Add user](#)

[Delete user](#)

Find users by username or access key

User name ▾

Groups

Administrator

Administrators

- Select the Security Credentials Tab

Users > Administrator

## Summary

User ARN arn:aws:iam::443007076818:user/Administrator 

Path /

Creation time 2019-04-10 10:04 MDT

**Permissions**

Groups (1)

Tags (1)

Security credentials

Access Advisor

- Select Create Access Key

Search IAM

Users > Administrator

Dashboard

Groups

**Users**

Roles

Policies

Identity providers

Account settings

Credential report

Encryption keys

## Summary

User ARN arn:aws:iam::443007076818:user/Administrator 

Path /

Creation time 2019-04-10 10:04 MDT

Permissions

Groups (1)

Tags (1)

Security credentials

Access Advisor

### Sign-in credentials

#### Summary

- Console sign-in link: <https://cert-microcosm.signin.aws.amazon.com/console>

#### Console password

Enabled (last signed in Today) | [Manage](#)

#### Assigned MFA device

Not assigned | [Manage](#)

#### Signing certificates

None 

### Access keys

Use access keys to make secure REST or HTTP Query protocol requests to AWS service APIs. For your protection, you should never share your rotation. [Learn more](#)

[Create access key](#)

Access key ID	Created	Last used
AKIAWOJKHRHJCDYDAKNE	2019-04-10 10:04 MDT	2019-04-26 20:32 MDT with codedeploy in us-east-2

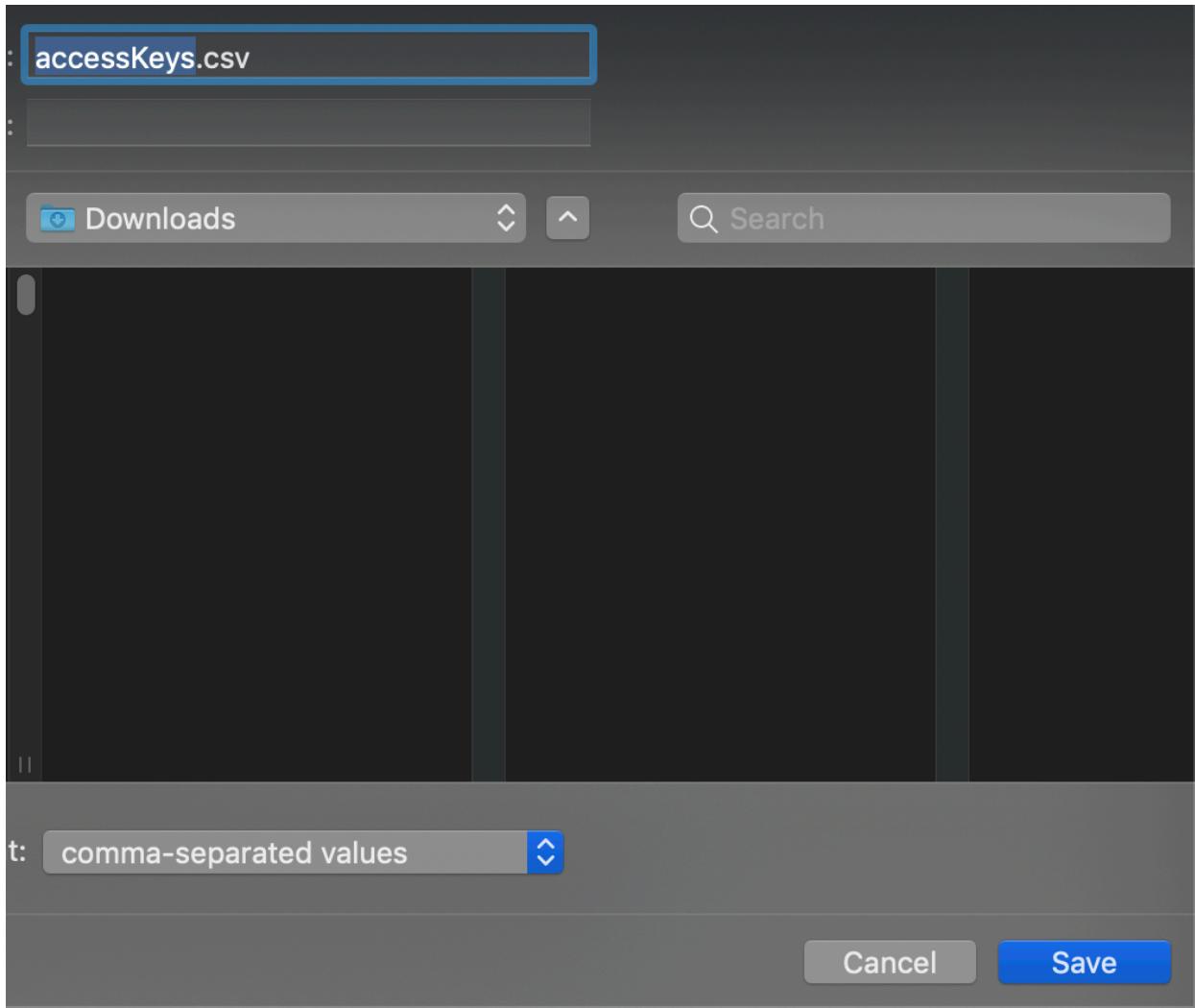
- You can unhide the access key to copy it and save it somewhere safe or press Download .csv file

The screenshot shows the AWS IAM 'Create access key' dialog box. It displays a 'Success' message: 'This is the **only** time that the secret access keys can be viewed or downloaded. You cannot recover them later. However, you can create new access keys at any time.' Below this, there is a 'Download .csv file' button. A table shows the generated keys:

Access key ID	Secret access key
AKIAWOJKHRHJASUVR35N	7ar8nUSG1C2gstsL9x3H6qBf46ta202Man6/wRXF <a href="#">Hide</a>

At the bottom right of the dialog is a 'Close' button.

- Save the CSV file somewhere you'll remember and that is secure



- Opening the CSV file, you'll see your KeyID and Secret Key. You will need this in the Jenkins Build section Below.

### accessKeys

Access key ID	Secret access key
AKIAW0JKHRHJASUYR35N	7ar8nUSG1C2gstL9x3H6qBf46ta202Man6/wRXF

### Modify Jenkins Build

- Navigate to your product page, select configure, and scroll all the way down to Post-Build Actions
- From your Deployment group page, copy and paste the Application Name, Deployment Group, and Deployment Config
- Select the region to deploy your code in

- Enter the name of the S3 bucket created previously and a folder name in which to store your deployments
- Enter the following strings for Include Files and Exclude Files (this grabs from the Jenkins workspace for your project, including source files. These files are Zipped, post-build, uploaded the given S3 bucket on AWS). Note, the .yml and .yaml files are used to give deployment instructions to AWS CodeDeploy, along with the scripts in the scripts folder, and the war file is build object created from your code. All of these files can be explored in your git repository, except the war file, or in your S3 bucket after a successful build. The exclude string excludes all files not called out in the include string.
  - \*\*/\*.html, \*\*/\*.yml, \*\*/\*.yaml, \*\*/scripts/\*.\*,\*\*/target/\*.war
  - /,/src/,/target
- Select Use Access/Secret Keys
- Enter the key information from the CSV file created previously
- Press Apply and Save

**Post-build Actions**

**Deploy an application to AWS CodeDeploy**

AWS CodeDeploy Application Name	petclinic
AWS CodeDeploy Deployment Group	petclinicDepGrp
AWS CodeDeploy Deployment Config	CodeDeployDefault.AllAtOnce
AWS Region	US_EAST_2
S3 Bucket	petclinicdeploy
S3 Prefix	target
Subdirectory	
Include Files	**/*.yml,**/*.html,**/*.yaml,**/scripts/*.*,**/target/*.war
Exclude Files	/src/target
Proxy Host	
Proxy Port	0
Version File	
Appspec.yml per Deployment Group	<input type="checkbox"/>
<input type="radio"/> Register Revision <input type="radio"/> Deploy Revision <input checked="" type="radio"/> Use Access/Secret keys	
<small>If these keys are left blank, the plugin will attempt to use credentials from the default provider chain. That is: Environment Variables, Java System properties, credentials profile file, and finally, EC2 Instance profile.</small>	
Save	Apply
AWS Secret Key	
AKIAWOJKHRHJCDYDAKNE	
.....	

## Build and Deploy

- From your Project Page in Jenkins select Build

Jenkins ➔ petclinic ➔

---

 Back to Dashboard

 Status

 Changes

 Workspace

-  Build Now
-  Delete Maven project
-  Configure
-  Modules
-  Favorite
-  SonarQube
-  Open Blue Ocean
-  Rename
-  Embeddable Build Status

☀

## Build History

trend =

X

●

#43

Apr 26, 2019 3:35 PMX

- Post build, you can navigate to the target folder in your S3 bucket to see the package uploaded

The screenshot shows the Amazon S3 console interface. At the top, the navigation path is displayed as: **Amazon S3** > **petclinicdeploy** > **target**. Below this, a large box is labeled **Overview**. A search bar contains the placeholder text: **Q Type a prefix and press Enter to search. Press ESC**. Below the search bar are three buttons: **Upload**, **Create folder**, and **Download**. To the right of the download button is a small icon. The main content area displays two items in a list:

- Name ▾**
- #43-3400450200434229806.zip**

**petclinic.war**

- Return to the CodeDeploy Console and Select Deployments
- Select the topmost Deployment and view the deployment progress

The screenshot shows the AWS CodeDeploy console with the following details:

- Deployment Status:** Shows 0 of 2 instances updated, with one instance in progress.
- Deployment Details:** Application: petclinic, Deployment ID: d-9GMQAH25Y, Deployment configuration: custom1, Deployment group: petclinicDepGrp, Status: In progress, Initiated by: user.
- Revision Details:** Revision location: s3://petclinicdeploy/target/#\$4-8254645117549694950.zip, Revision created: 1 minute ago, Description: Application revision registered via Jenkins.
- Deployment Lifecycle Events:** Two instances are listed:
 

Instance ID	Duration	Status	Most recent event	Events	Start time	End time
i-03a155360dafb37b5	-	In progress	BeforeBlockTraffic	View events	Apr 26, 2019 11:11 AM	-
i-0e2a50cb45d6bd604	-	In progress	BeforeBlockTraffic	View events	Apr 26, 2019 11:11 AM	-

- Under Instances at the bottom of the page, select View Events to see the progress and steps of the deployment

The screenshot shows the AWS CodeDeploy console with the following details:

- Deployment Details:** Application: petclinic, Deployment ID: d-9GMQAH25Y, Deployment configuration: custom1, Deployment group: petclinicDepGrp, Status: In progress, Initiated by: user.
- Revision Details:** Revision location: s3://petclinicdeploy/target/#\$4-8254645117549694950.zip, Revision created: 2 minutes ago, Description: Application revision registered via Jenkins.
- Events Table:**

Event	Duration	Status	Error code	Start time	End time
BeforeBlockTraffic	less than one second	Succeeded	-	Apr 26, 2019 11:11 AM	Apr 26, 2019 11:11 AM
BlockTraffic	-	Pending	-	-	-
AfterBlockTraffic	-	Pending	-	-	-
ApplicationStop	-	Pending	-	-	-
DownloadBundle	-	Pending	-	-	-
BeforeInstall	-	Pending	-	-	-
Install	-	Pending	-	-	-
AfterInstall	-	Pending	-	-	-
ApplicationStart	-	Pending	-	-	-
ValidateService	-	Pending	-	-	-
BeforeAllowTraffic	-	Pending	-	-	-
AllowTraffic	-	Pending	-	-	-
AfterAllowTraffic	-	Pending	-	-	-

Application <a href="#">petclinic</a>	Deployment ID d-FRO04J65Y																																																								
Deployment configuration <a href="#">CodeDeployDefault.AllAtOnce</a>	Deployment group <a href="#">petclinicDepGrp</a>																																																								
<b>Revision details</b>																																																									
Revision location <a href="s3://petclinicdeploy/target/#62-2472083856417875777.zip?eTag=5c72ef1adb2c717b3611effbef97645">s3://petclinicdeploy/target/#62-2472083856417875777.zip? eTag=5c72ef1adb2c717b3611effbef97645</a>	Revision created 33 minutes ago																																																								
<table border="1"> <thead> <tr> <th>Event</th><th>Duration</th><th>Status</th><th>Error code</th></tr> </thead> <tbody> <tr><td>BeforeBlockTraffic</td><td>less than one second</td><td><span style="color: green;">✔ Succeeded</span></td><td>-</td></tr> <tr><td>BlockTraffic</td><td>22 seconds</td><td><span style="color: green;">✔ Succeeded</span></td><td>-</td></tr> <tr><td>AfterBlockTraffic</td><td>less than one second</td><td><span style="color: green;">✔ Succeeded</span></td><td>-</td></tr> <tr><td>ApplicationStop</td><td>less than one second</td><td><span style="color: green;">✔ Succeeded</span></td><td>-</td></tr> <tr><td>DownloadBundle</td><td>less than one second</td><td><span style="color: green;">✔ Succeeded</span></td><td>-</td></tr> <tr><td>BeforeInstall</td><td>less than one second</td><td><span style="color: green;">✔ Succeeded</span></td><td>-</td></tr> <tr><td>Install</td><td>less than one second</td><td><span style="color: green;">✔ Succeeded</span></td><td>-</td></tr> <tr><td>AfterInstall</td><td>less than one second</td><td><span style="color: green;">✔ Succeeded</span></td><td>-</td></tr> <tr><td>ApplicationStart</td><td>less than one second</td><td><span style="color: green;">✔ Succeeded</span></td><td>-</td></tr> <tr><td>ValidateService</td><td>less than one second</td><td><span style="color: green;">✔ Succeeded</span></td><td>-</td></tr> <tr><td>BeforeAllowTraffic</td><td>less than one second</td><td><span style="color: green;">✔ Succeeded</span></td><td>-</td></tr> <tr><td>AllowTraffic</td><td>29 minutes 21 seconds</td><td><span style="color: green;">✔ Succeeded</span></td><td>-</td></tr> <tr><td>AfterAllowTraffic</td><td>less than one second</td><td><span style="color: green;">✔ Succeeded</span></td><td>-</td></tr> </tbody> </table>		Event	Duration	Status	Error code	BeforeBlockTraffic	less than one second	<span style="color: green;">✔ Succeeded</span>	-	BlockTraffic	22 seconds	<span style="color: green;">✔ Succeeded</span>	-	AfterBlockTraffic	less than one second	<span style="color: green;">✔ Succeeded</span>	-	ApplicationStop	less than one second	<span style="color: green;">✔ Succeeded</span>	-	DownloadBundle	less than one second	<span style="color: green;">✔ Succeeded</span>	-	BeforeInstall	less than one second	<span style="color: green;">✔ Succeeded</span>	-	Install	less than one second	<span style="color: green;">✔ Succeeded</span>	-	AfterInstall	less than one second	<span style="color: green;">✔ Succeeded</span>	-	ApplicationStart	less than one second	<span style="color: green;">✔ Succeeded</span>	-	ValidateService	less than one second	<span style="color: green;">✔ Succeeded</span>	-	BeforeAllowTraffic	less than one second	<span style="color: green;">✔ Succeeded</span>	-	AllowTraffic	29 minutes 21 seconds	<span style="color: green;">✔ Succeeded</span>	-	AfterAllowTraffic	less than one second	<span style="color: green;">✔ Succeeded</span>	-
Event	Duration	Status	Error code																																																						
BeforeBlockTraffic	less than one second	<span style="color: green;">✔ Succeeded</span>	-																																																						
BlockTraffic	22 seconds	<span style="color: green;">✔ Succeeded</span>	-																																																						
AfterBlockTraffic	less than one second	<span style="color: green;">✔ Succeeded</span>	-																																																						
ApplicationStop	less than one second	<span style="color: green;">✔ Succeeded</span>	-																																																						
DownloadBundle	less than one second	<span style="color: green;">✔ Succeeded</span>	-																																																						
BeforeInstall	less than one second	<span style="color: green;">✔ Succeeded</span>	-																																																						
Install	less than one second	<span style="color: green;">✔ Succeeded</span>	-																																																						
AfterInstall	less than one second	<span style="color: green;">✔ Succeeded</span>	-																																																						
ApplicationStart	less than one second	<span style="color: green;">✔ Succeeded</span>	-																																																						
ValidateService	less than one second	<span style="color: green;">✔ Succeeded</span>	-																																																						
BeforeAllowTraffic	less than one second	<span style="color: green;">✔ Succeeded</span>	-																																																						
AllowTraffic	29 minutes 21 seconds	<span style="color: green;">✔ Succeeded</span>	-																																																						
AfterAllowTraffic	less than one second	<span style="color: green;">✔ Succeeded</span>	-																																																						

- Once the Deployment is successful, view your deployed project - Navigate to the EC2 Instance IP at port 80 to view the Tomcat management console, and navigate to the [EC2 Instance Ip]/petclinic to view the deployed Java Project

Developer Tools > CodeDeploy > Deployments > d-61VZJA35Y

### d-61VZJA35Y

**Deployment status**

Installing application on your instances 2 of 2 instances updated Succeeded

**Deployment details**

Application petclinic	Deployment ID d-61VZJA35Y	Status <span style="color: green;">Succeeded</span>
Deployment configuration CodeDeployDefault.AllAtOnce	Deployment group petclinicDepGrp	Initiated by user

**Revision details**

Revision location s3://petclinicdeploy/target/#64-4773894709153267946.zip eTag=fea9c4099245a26e9f71d169746d58cb	Revision created 1 minute ago	Description Application revision registered via Jenkins
-----------------------------------------------------------------------------------------------------------------------	----------------------------------	------------------------------------------------------------

**Deployment lifecycle events**

Instance ID	Duration	Status	Most recent event	Events	Start time	End time
i-08dbe30c18a614261	59 seconds	<span style="color: green;">Succeeded</span>	AfterAllowTraffic	<a href="#">View events</a>	Apr 26, 2019 3:51 PM	Apr 26, 2019 3:52 PM
i-0fcfb0d38fdf28923	1 minute 0 seconds	<span style="color: green;">Succeeded</span>	AfterAllowTraffic	<a href="#">View events</a>	Apr 26, 2019 3:51 PM	Apr 26, 2019 3:52 PM

Pet Clinic A Spring Framework Demonstration

Spring

Home Find owners Who-am-I Error Help



[Return to Table of Contents](#)

## AWS ECS Based Code Deploy

### Additional Task Definitions

Task/Service Name	Container Name	Image	Ports	Volumes	Mount P
petclinic	tomcat8	tomcat:8-jre8	8080	/webapps	[TODO path to folder /opt/tomcat/webapps]
			80		

## Create EC2 ECS AIM

- Include codedeploy and logging scripts
- Not using NAT if possible

## Create Petclinic ECS Service

- Create folder on EC2 image that maps to apache webapps folder
- Create EC2 based task definition
- Set up routing for image?
- Create 2 ecs tasks within petclinic service

## Set up CodeDeploy Application

### Create Build Step in Jenkins

To DO

Create Cluster

Create Task definition

tomcat:8-jre8 name petclinic Port 8080 Mem 1 gb Cpu .5

Initialize Service

Create Role

Create Load Balancer

Create CodeDeploy -> Deploy -> Application

[http://\[petclinic IP\]:8080/petclinic](http://[petclinic IP]:8080/petclinic)

Create S3 bucket Jenkins plugin - s3 publisher manage jenkins - configure system - Amazon s3 profiles - add ... (can check the role box if using an ec2 instance of Jenkins and assigned it a role with S3 Full Access) Configure build - Post build actions - Publish artifacts to S3 Bucket Source \*\*/target/\*.war Exclude /target destination bucket: petclinicdeploy

object key if the path within the bucket eg target/petclinic.war

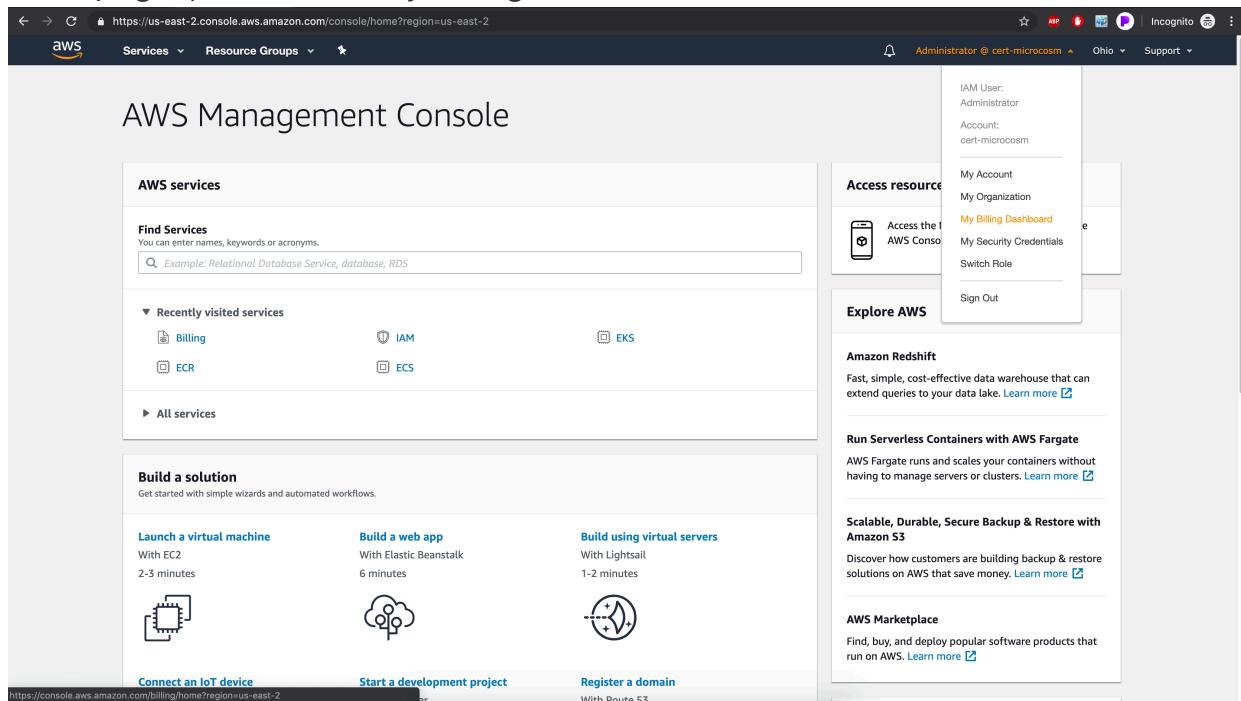
[Return to Table of Contents](#)

## APPENDIX A

---

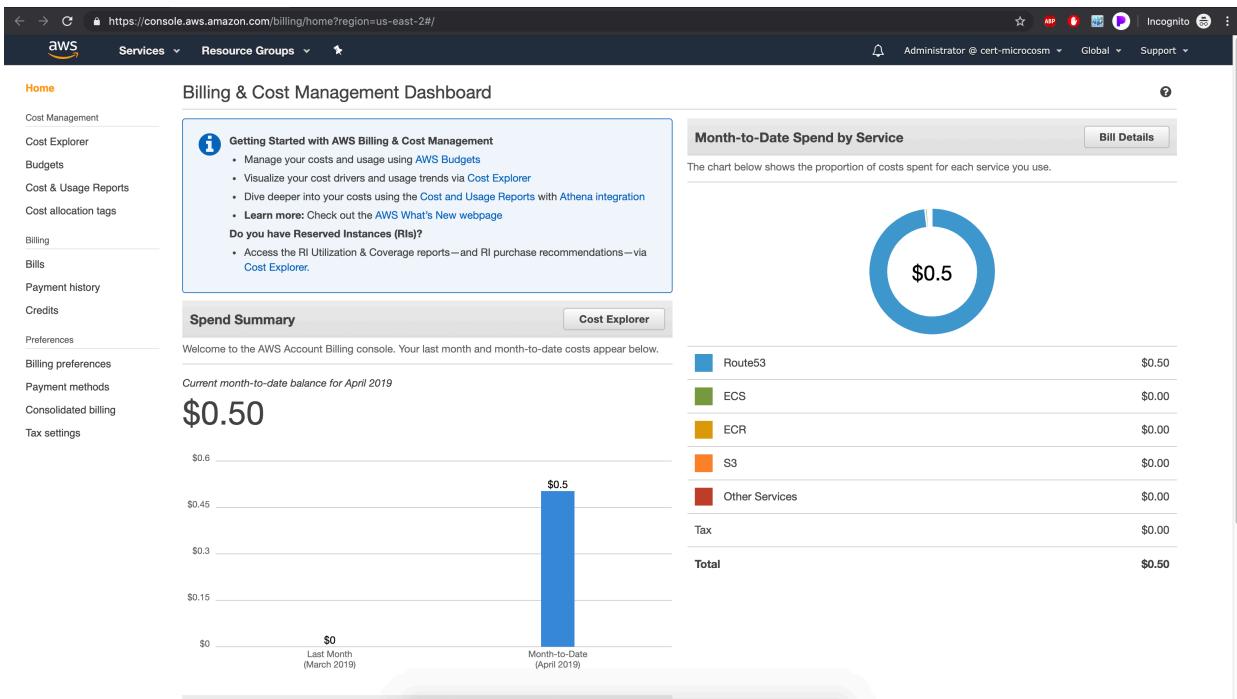
### Billing

- To view billing information, select your username dropdown from the top right of most pages, and select My Billing dashboard



The screenshot shows the AWS Management Console homepage. At the top, there's a navigation bar with links for Services, Resource Groups, and a user dropdown set to "Administrator @ cert-microcosm". Below the navigation is a search bar and a "Find Services" section where users can enter names, keywords or acronyms. A "Recently visited services" section lists Billing, IAM, EKS, ECR, and ECS. A "Build a solution" section offers options like Launch a virtual machine (With EC2, 2-3 minutes), Build a web app (With Elastic Beanstalk, 6 minutes), Build using virtual servers (With Lightsail, 1-2 minutes), Connect an IoT device, Start a development project, and Register a domain. On the right side, a sidebar menu is open under "Access resource", showing options like "Access the AWS Console", "My Account", "My Organization", "My Billing Dashboard" (which is highlighted in orange), "My Security Credentials", "Switch Role", and "Sign Out". Other sections visible include "Amazon Redshift", "Run Serverless Containers with AWS Fargate", "Scalable, Durable, Secure Backup & Restore with Amazon S3", and "AWS Marketplace".

- Explore your billing information, current charges and projections



[Return to Table of Contents](#)

## Add Sonatype Nexus build step

Use the following instructions to configure Jenkins to connect to Nexus Repository Manager:

- Select Manage Jenkins from the Dashboard's left-navigation menu
- Select Configure System from the list of configuration options
- In the Sonatype Nexus section, click the Add Nexus Repository Manager Server dropdown menu and then select Nexus Repository Manager 2.x Server.

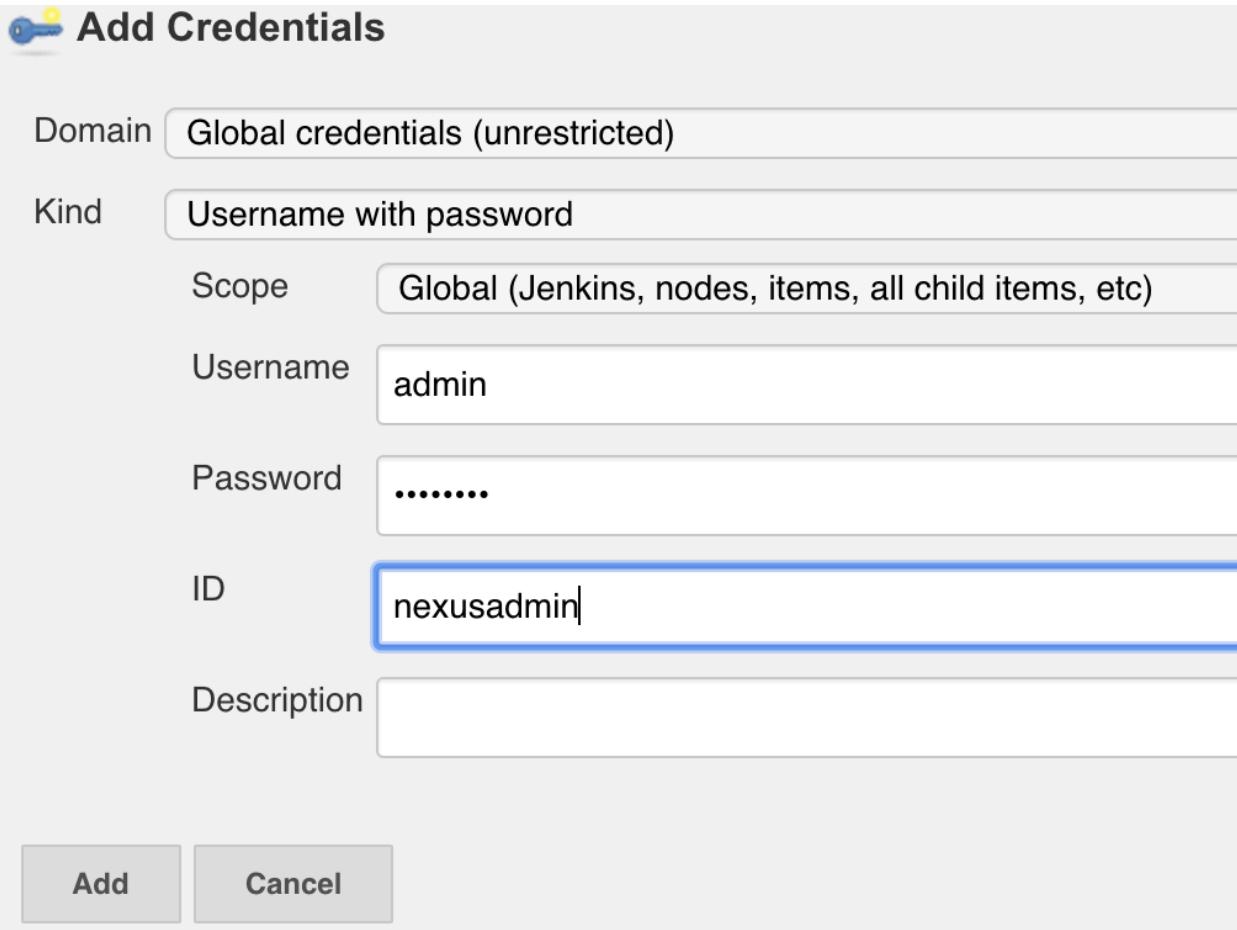
**Sonatype Nexus**

Nexus Repository Manager Servers

Add Nexus Repository Manager Server ▾

- Nexus Repository Manager 2.x Server
- Nexus Repository Manager 3.x Server

- Credentials: Select the Add button to enter your Nexus Repository Manager username and password (defaults = admin/admin123) using the Jenkins Provider Credentials: Jenkins modal window.



The screenshot shows the 'Add Credentials' dialog box. It has several input fields and buttons. The 'Domain' field is set to 'Global credentials (unrestricted)'. The 'Kind' field is set to 'Username with password'. The 'Scope' field is set to 'Global (Jenkins, nodes, items, all child items, etc)'. The 'Username' field contains 'admin'. The 'Password' field contains '.....'. The 'ID' field contains 'nexusadmin' and is highlighted with a blue selection bar. The 'Description' field is empty. At the bottom are two buttons: 'Add' and 'Cancel'.

Domain	Global credentials (unrestricted)
Kind	Username with password
Scope	Global (Jenkins, nodes, items, all child items, etc)
Username	admin
Password	.....
ID	nexusadmin
Description	

**Add**    **Cancel**

- Enter the following:
  - Display Name: Name of the server you want shown when selecting Nexus Repository Manager instances for build jobs
  - Server ID: A unique ID used to reference Nexus Repository Manager in Build Pipeline scripts. It should be alphanumeric without spaces (eg make up a string like hello1234567890bobmyfriend)
  - Server URL: Location of your Nexus Repository Manager server (ex: [http://\[NEXUS\\_IP\\_ADDRESS\]:8081/nexus](http://[NEXUS_IP_ADDRESS]:8081/nexus))
  - Select your Nexus Repository Manager username and password from the Credentials dropdown list
- Click the Test Connection button
- After a successful connection to Nexus Repository Manager, click the Apply, then Save buttons

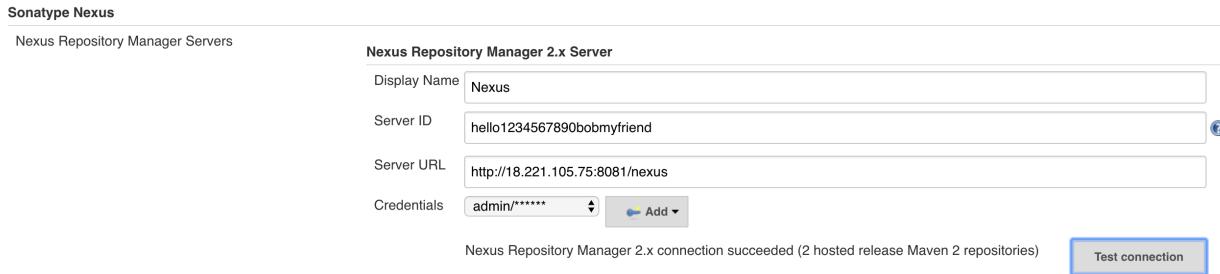
Sonatype Nexus

Nexus Repository Manager Servers

Nexus Repository Manager 2.x Server

Display Name	Nexus
Server ID	hello1234567890bobmyfriend
Server URL	http://18.221.105.75:8081/nexus
Credentials	admin/******** <input type="button" value="Add"/>

Nexus Repository Manager 2.x connection succeeded (2 hosted release Maven 2 repositories)



- Return to the PetClinic build configuration page
- Select the Post Steps Tab and click Add post-build step
- Select Nexus Repository Manager Publisher
  - Enter the following parameters:
    - Nexus Instance: Enter "Nexus"
    - Nexus Repository: Select the "Releases" repository
    - Packages: Select packages to publish to Nexus Repository Manager during your freestyle build. For this example, use the Add Package dropdown to select a Maven Package
      - For Group enter: "petclinic-main"
      - For Artifact enter: "petclinic.war"
      - For Version enter: 2.3
      - For Packaging enter: "war"
      - Click "Add Artifact Path" and choose "Maven Artifact"
      - For Filepath enter:  
"/var/jenkins\_home/workspace/petclinic/target/petclinic.war"
- Click Apply and Save

Nexus Repository Manager Publisher

Nexus Instance: Nexus

Nexus Repository: Releases

Tag:

Packages

Group	petclinic-main
Artifact	petclinic.war
Version	2.3
Packaging	war

Artifacts

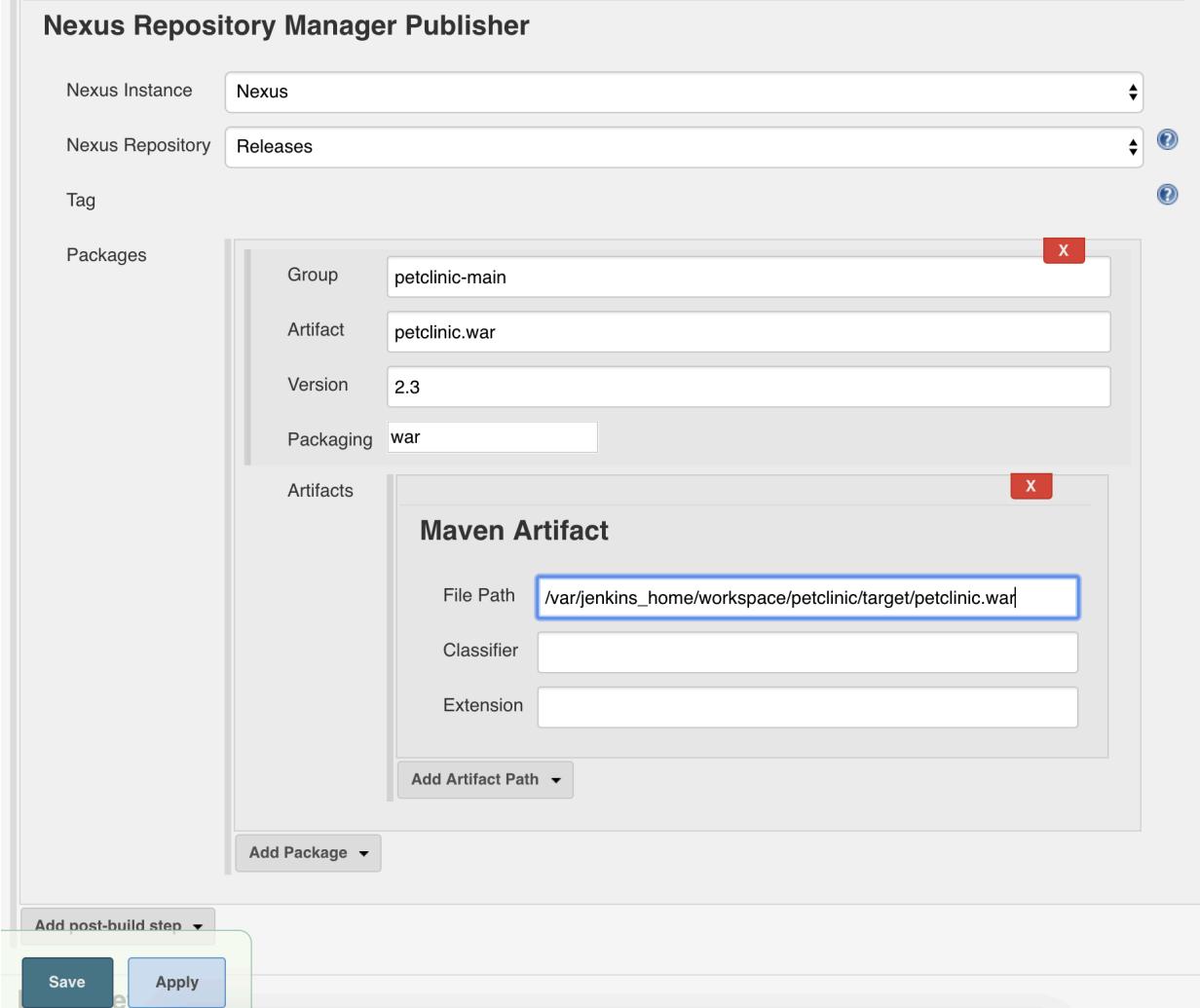
Maven Artifact

File Path	/var/jenkins_home/workspace/petclinic/target/petclinic.war
Classifier	
Extension	

Add Package ▾

Add post-build step ▾

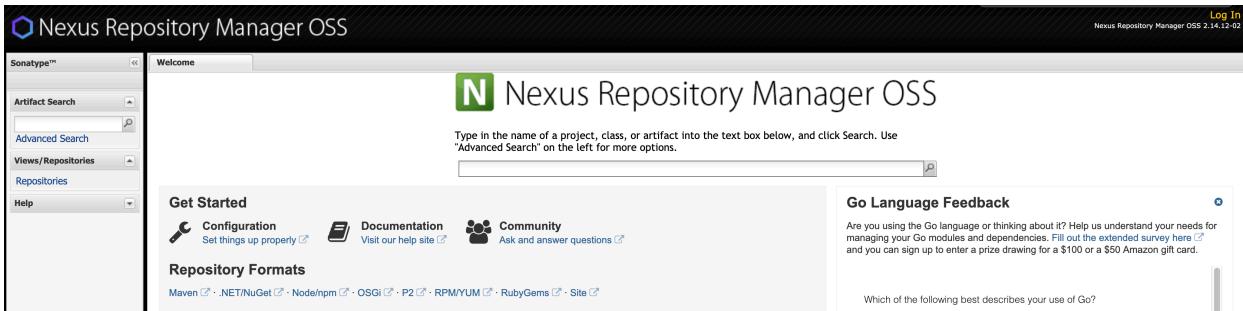
**Save** **Apply**



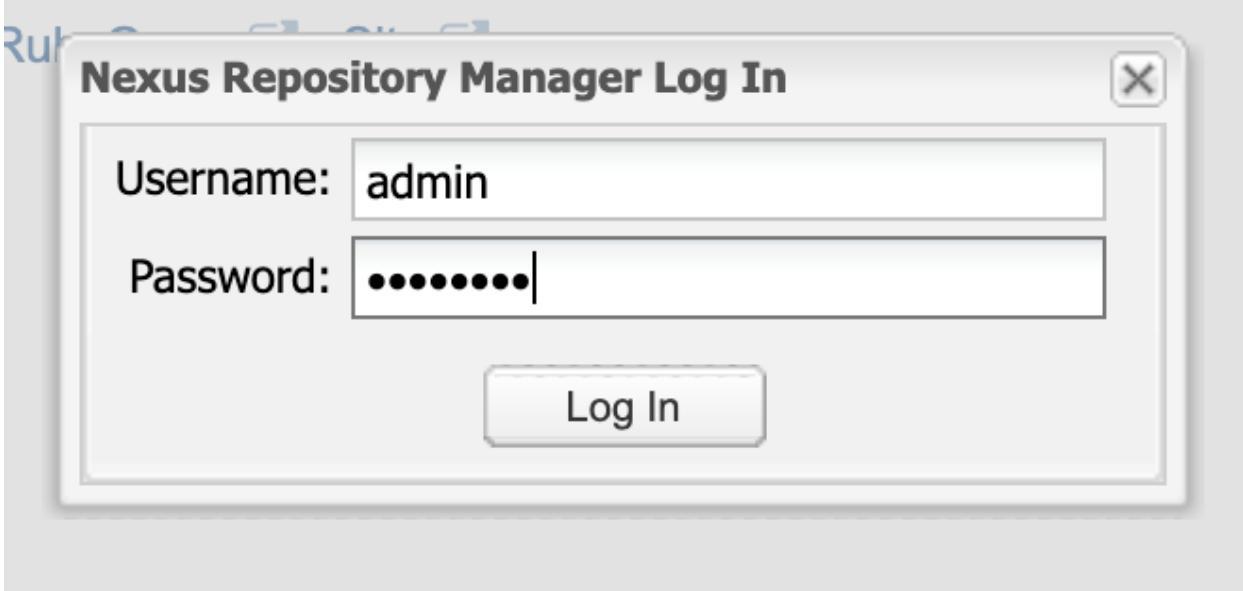
- After a successful Jenkins build, if you look in the build console output, you should see, near the bottom of the page, a message indicating success, similar to:

```
INFO: -----  
INFO: EXECUTION SUCCESS  
INFO: -----  
INFO: Total time: 22.909s  
INFO: Final Memory: 14M/348M  
INFO: -----  
Uploading Maven asset with groupId: petclinic-main artifactId: petclinic.war version: 2.3 To repository: releases  
Successfully Uploaded Maven Assets
```

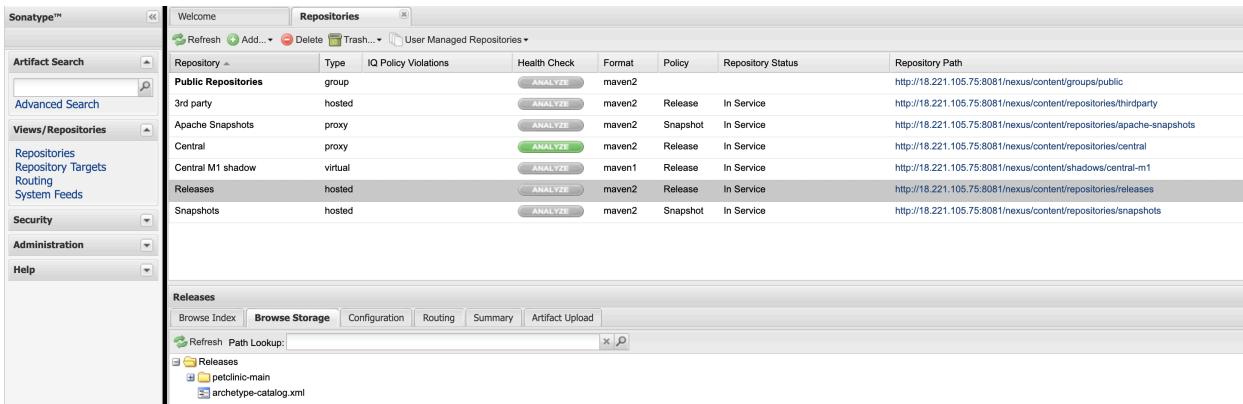
- To verify navigate to the Nexus Repository manager web UI ([http://\[NEXUS\\_IP\\_ADDRESS\]:8081/nexus](http://[NEXUS_IP_ADDRESS]:8081/nexus))



- Login using the default credentials admin:admin123



- View your selected packages in the under the "Releases" repository



[Return to Table of Contents](#)

## Creating Identity and Access Management (IAM) Users

- From the Services drop down on the top left, select IAM to get to the IAM

## Dashboard

Welcome to Identity and Access Management

IAM users sign-in link:  
https://cert-microcosm.signin.aws.amazon.com/console

**IAM Resources**

- Users: 1
- Groups: 1
- Roles: 9
- Identity Providers: 1
- Customer Managed Policies: 0

**Security Status**

4 out of 4 complete.

- Activate MFA on your root account
- Create individual IAM users
- Use groups to assign permissions
- Apply an IAM password policy

**Feature Spotlight**

Introduction to AWS IAM

**Additional Information**

- IAM best practices
- IAM documentation
- Web Identity Federation Playground
- Policy Simulator
- Videos, IAM release history and additional resources

- Select Users on the Left
- Click Add User

Add user Delete user

User name	Groups	Access key age	Password age	Last activity	MFA
smorley	admin	5 days	5 days	5 days	Not enabled

- Enter the username and custom password
- Check Programmatic Access if this user will need CLI or API access
- Check AWS management Console Access
- Check require password reset to require user to create a secret password

- Click Next: Permissions

User name\*

Add another user

Select AWS access type

Select how these users will access AWS. Access keys and autogenerated passwords are provided in the last step. [Learn more](#)

Access type\*  Programmatic access  
Enables an **access key ID** and **secret access key** for the AWS API, CLI, SDK, and other development tools.

AWS Management Console access  
Enables a **password** that allows users to sign-in to the AWS Management Console.

Console password\*  Autogenerated password  
 Custom password  
  
 Show password

Require password reset  User must create a new password at next sign-in

\* Required

Cancel **Next: Permissions**

- Click add user to Group (or Copy Permissions from an existing user, if applicable)
- If the desired group does not yet exist, select Create Group

Add user to group

Create group Refresh

Group	Attached policies
admin	AdministratorAccess

Showing 1 result

Set permissions boundary

Cancel Previous **Next: Tags**

- Name your Group
- Select the permissions appropriate for the group
  - To create a user capable of any actions within AWS, filter on Administrator,

and select AdministratorAccess

- Select Create Group

Policy name	Type	Used as	Description
<input checked="" type="checkbox"/> AdministratorAccess	Job function	Permissions policy (2)	Provides full access to AWS services and resources.
<input type="checkbox"/> AmazonAPIGatewayAdministrator	AWS managed	None	Provides full access to create/edit/delete APIs in Amazon API Gateway via the AWS Manag...
<input type="checkbox"/> AWSAppSyncAdministrator	AWS managed	None	Provides administrative access to the AppSync service, though not enough to access via th...
<input type="checkbox"/> AWSCloud9Administrator	AWS managed	None	Provides administrator access to AWS Cloud9.
<input type="checkbox"/> AWSSSODirectoryAdministrator	AWS managed	None	Administrator access for SSO Directory
<input type="checkbox"/> AWSSOMMasterAccountAdministrator	AWS managed	None	Provides access within AWS SSO to manage AWS Organizations master and member acco...
<input type="checkbox"/> AWSSOMMemberAccountAdministrator	AWS managed	None	Provides access within AWS SSO to manage AWS Organizations member accounts and clo...
<input type="checkbox"/> DatabaseAdministrator	Job function	None	Grants full access permissions to AWS services and actions required to set up and configur...
<input type="checkbox"/> NetworkAdministrator	Job function	None	Grants full access permissions to AWS services and actions required to set up and configur...
<input type="checkbox"/> SystemAdministrator	Job function	None	Grants full access permissions necessary for resources required for application and develop...

- Select the created group
- Press Next:Tags

Group	Attached policies
<input checked="" type="checkbox"/> Administrators	AdministratorAccess
<input type="checkbox"/> admin	AdministratorAccess

- Add tags if desired
- Click Review

The screenshot shows the 'Add user' wizard at step 3, titled 'Add tags (optional)'. It includes a note about IAM tags being key-value pairs. A table allows adding tags with 'Key' and 'Value (optional)' columns. One tag 'administrators' is listed. A link to 'Learn more' is present. Navigation buttons at the bottom include 'Cancel', 'Previous', and 'Next: Review'.

- After reviewing, click Create User

The screenshot shows the 'Review' step of the 'Add user' wizard. It displays the user details: User name (Administrator), AWS access type (Programmatic access and AWS Management Console access), Console password type (Custom), Require password reset (Yes), and Permissions boundary (Permissions boundary is not set). Below this is a 'Permissions summary' section showing the user will be added to the 'Administrators' group. There's also a 'Tags' section where a tag 'administrators' is listed with an empty value. Navigation buttons at the bottom include 'Cancel', 'Previous', and 'Create user'.

- On the Success Page:
  - Unhide the Secret Access Key column
  - Copy off and save this key. It cannot be recovered at a later step, so NOW is your only chance. This key is essential for use with the AWS CLI. If you don not copy this key now, you'll need to generate a new key or create a new user later.**

- Press close

**Success**

You successfully created the users shown below. You can view and download user security credentials. You can also email users instructions for signing in to the AWS Management Console. This is the last time these credentials will be available to download. However, you can create new credentials at any time.

Users with AWS Management Console access can sign-in at: <https://cert-microcosm.signin.aws.amazon.com/console>

User	Access key ID	Secret access key	Email login instructions
Administrator	AKIAW0JKHRHJCDYDAKNE	ZNHgDgrWBqnBR+K2AxGn csuI7m2SiLH0Pjkl4yqit5 Hide	<a href="#">Send email</a> <a href="#">Hide</a>

[Download .csv](#)

[Close](#)

- Selecting Users from the left side of the IAM Console, allows you to select a user. From this view, you can copy the user's login URL, edit user information, create keys, etc.

Users > Administrator

**Summary**

User ARN: arn:aws:iam::443007076818:user/Administrator

Path: /

Creation time: 2019-04-10 10:04 MDT

[Delete user](#) [?](#)

**Sign-in credentials**

Console sign-in link: <https://cert-microcosm.signin.aws.amazon.com/console>

Console password: Enabled (last signed in Today) | [Manage](#)

Assigned MFA device: Not assigned | [Manage](#)

Signing certificates: None [?](#)

**Access keys**

Use access keys to make secure REST or HTTP Query protocol requests to AWS service APIs. For your protection, you should never share your secret keys with anyone. As a best practice, we recommend frequent key rotation. [Learn more](#)

[Create access key](#)

Access key ID	Created	Last used	Status
AKIAW0JKHRHJCDYDAKNE	2019-04-10 10:04 MDT	2019-04-15 16:13 MDT with servicediscovery in us-east-2	Active   <a href="#">Make inactive</a> <a href="#">X</a>

**SSH keys for AWS CodeCommit**

Use SSH public keys to authenticate access to AWS CodeCommit repositories. [Learn more](#)

[Upload SSH public key](#)

[Return to Table of Contents](#)

## Create Key Pair

Key pairs are used for SSH and other authentication with AWS Amazon Machine Image instances (AMIs).

- From the EC2 Service Dashboard, select Key Pairs under Network & security on the left
- Select Create Key Pair

The screenshot shows the AWS EC2 Service Dashboard. The left sidebar is collapsed, and the main content area is titled 'Key Pairs'. At the top, there are three buttons: 'Create Key Pair' (highlighted in blue), 'Import Key Pair', and 'Delete'. Below these buttons is a search bar with the placeholder 'Filter by attributes or search by keyword'. A message states 'You do not have any Key Pairs in this region.' followed by 'Click the "Create Key Pair" button to create your first Key Pair.' A large blue 'Create Key Pair' button is centered below the message. The bottom of the page includes standard AWS footer links: 'Feedback', 'English (US)', '© 2008–2019, Amazon Web Services, Inc. or its affiliates. All rights reserved.', 'Privacy Policy', and 'Terms of Use'.

- Enter a name for the key pair and press Create
- Upon clicking Create a file will be downloaded to the users computer containing the private key as a .pem file. Do not lose this file or your key pair will be useless.

The screenshot shows the AWS EC2 dashboard with the 'Key Pairs' section selected. A modal window titled 'Create Key Pair' is open, prompting for a key pair name, which is set to 'Administrator1'. Below the input field are 'Cancel' and 'Create' buttons.

- View key pair details

The screenshot shows the AWS EC2 dashboard with the 'Key Pairs' section selected. A table lists one key pair named 'Administrator1' with its corresponding Fingerprint: 'fa:c0:90:4b:a7:bf:82:9f:3d:13:1c:73:dd:83:eb:b4:bd:74:f2:26'.

[Return to Table of Contents](#)

## Creating Roles

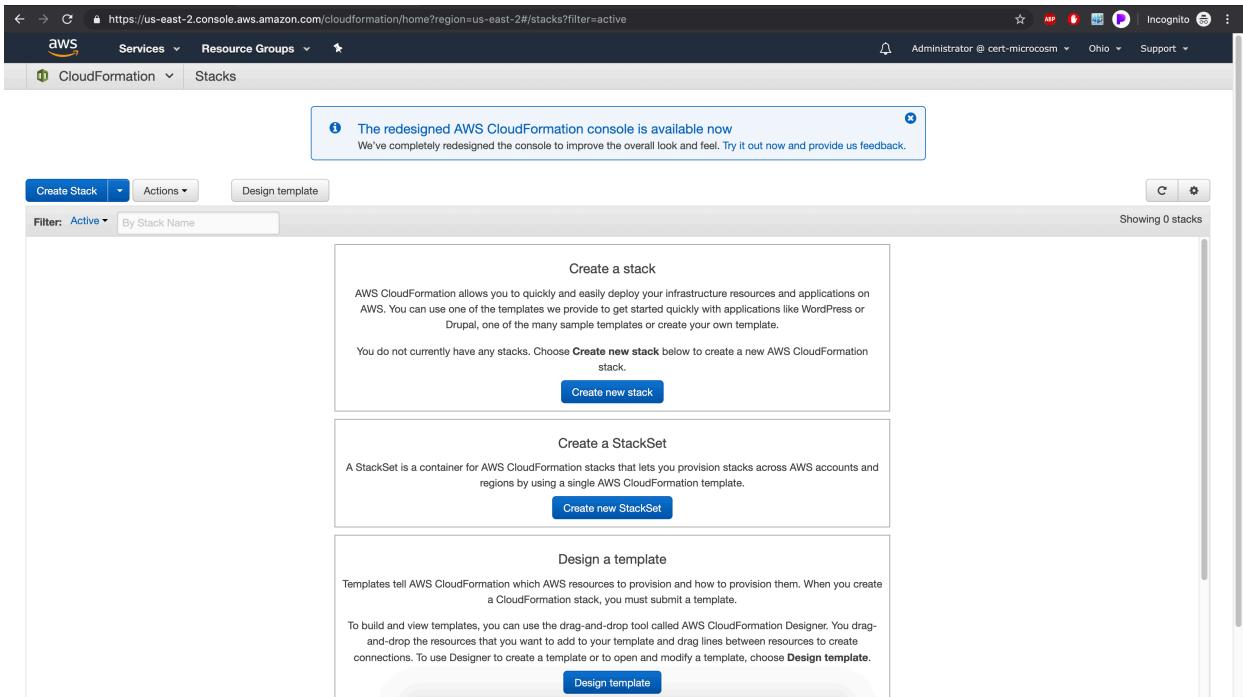
See the instructions in the [Create IAM Role for EC2 Based CodeDeploy](#) section.

[Return to Table of Contents](#)

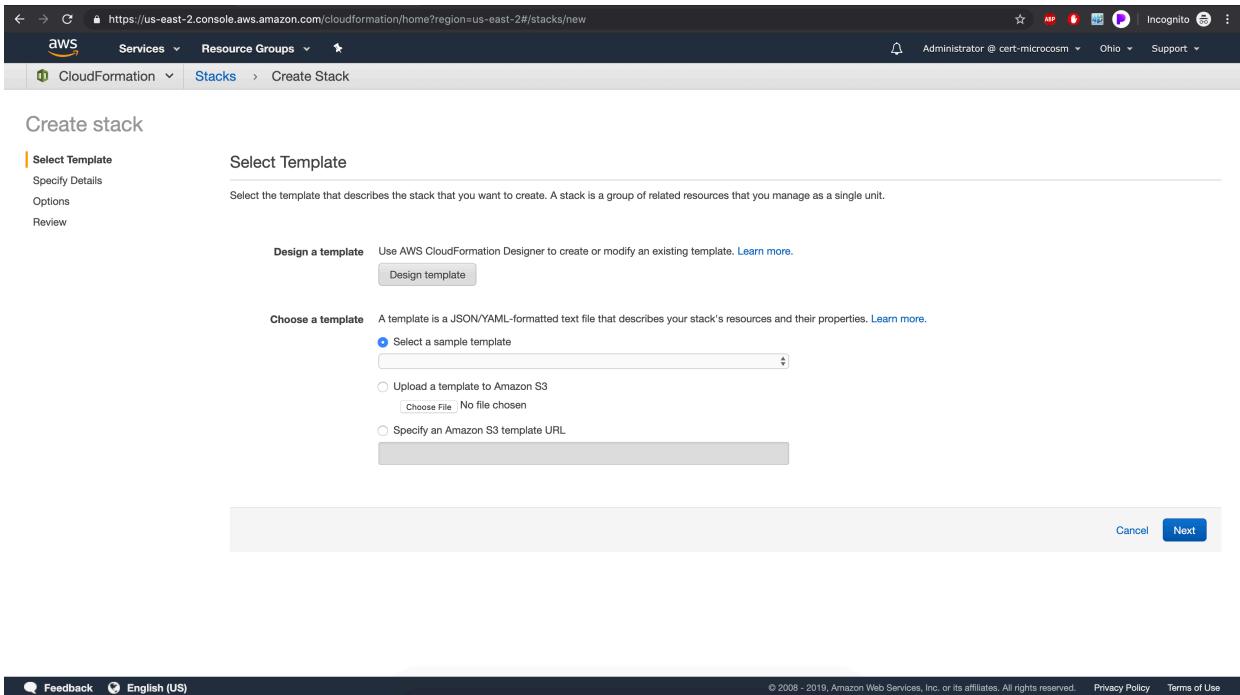
## Create VPCs and user Accounts using CloudFormation Templates

This Section covers using AWS CloudFormation to deploy templates. The templates used here include instructions to deploy lambda functions, VPCs, security groups, internet gateways, group policies, IAM users, routing tables, etc.

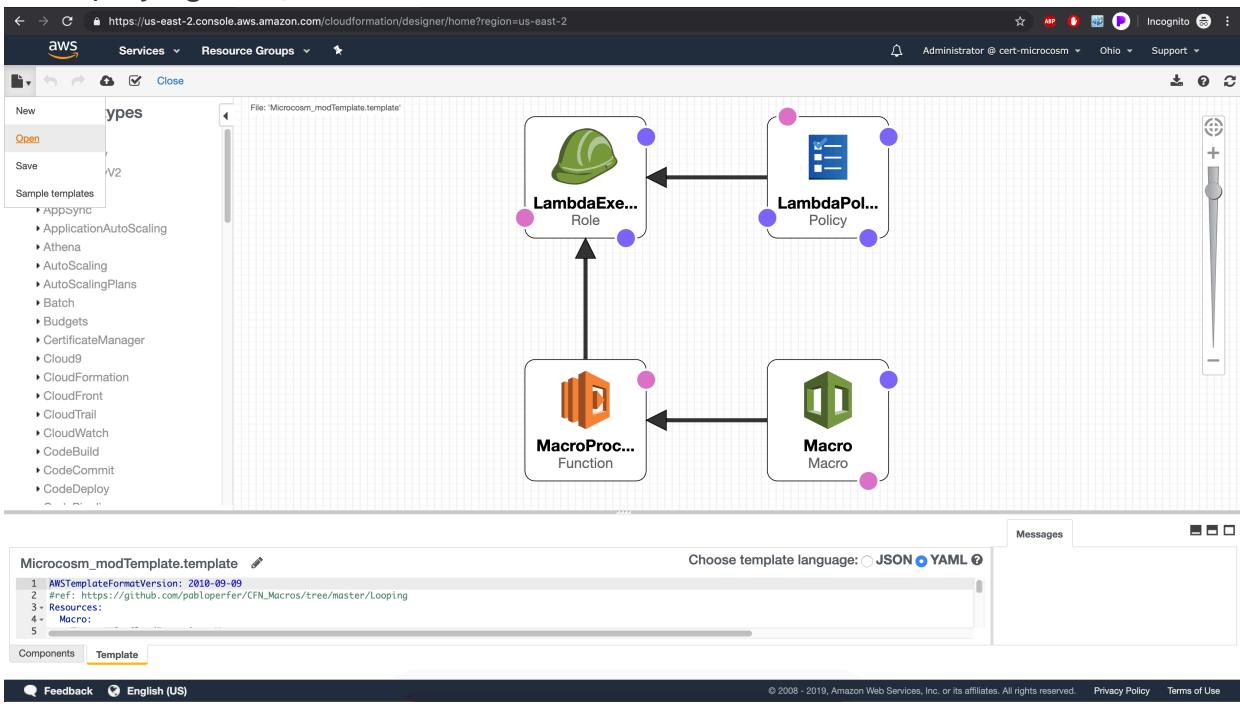
- Using the Services dropdown at the top on most AWS pages, navigate to CloudFormation. and select Create Stack



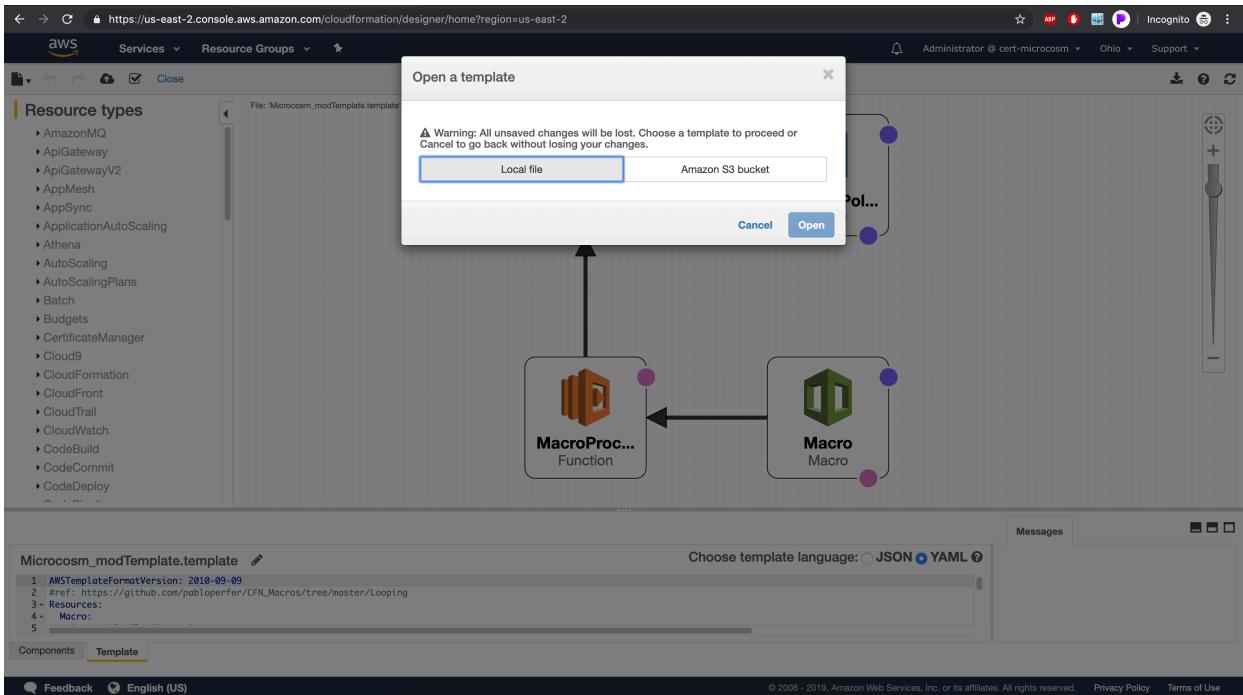
- To create or view a template, on the Select Template page, click Design Template



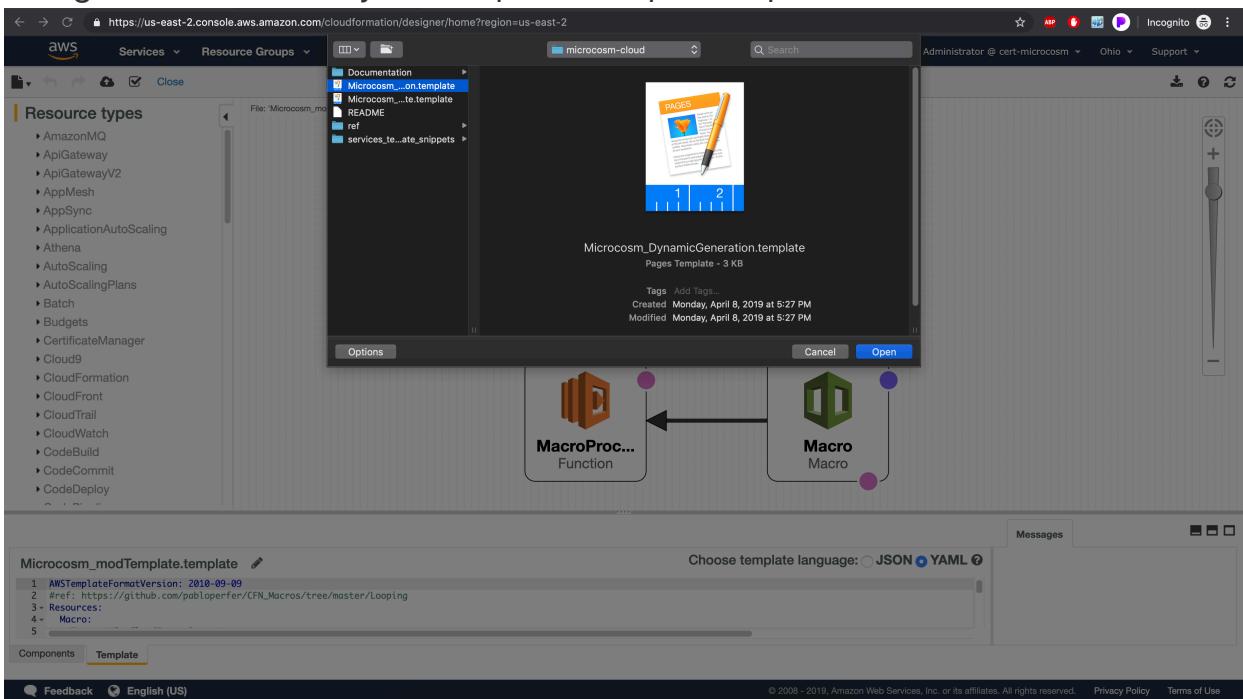
- To load an existing template, select open from the file icon in the left menu. (Note, the template pictured graphically here represents the Macro stack we'll be deploying later)



- Select the source of your template. It is good practice to name your templates with the extension `.template` or some variation thereof



- Navigate to and select your template and press Open



- With the template loaded, you are free to inspect, augment, change, or update its various elements. The template depicted here instantiates a VPC with the accompanying internet gateway, routes, security group, a user account, etc. This is our Dynamic template - as in when the template is loaded, the Macro template uses a lambda function, written in python, to duplicate the elements within the template as many times as specified. Note, that AWS templates have

a hard limit of 200 elements per template and AWS have a default limit of 5 VPCs and internet gateways (see above for requesting additional VPCs in a region).

The screenshot shows the AWS CloudFormation Designer interface. On the left, a sidebar lists 'Resource types' including AmazonMQ, ApiGateway, AppMesh, AppSync, ApplicationAutoScaling, Athena, AutoScaling, AutoScalingPlans, Batch, Budgets, CertificateManager, Cloud9, CloudFormation, CloudFront, CloudTrail, CloudWatch, CodeBuild, CodeCommit, and CodeDeploy. The main area displays a diagram of a network architecture with various resources: two EC2 instances (StudentVP...), an IAM user (StudentUser...), an Internet gateway (StudentVP...InternetGateway), a route table (StudentVP...Route), and a security group (StudentVP...SecurityGroup). Arrows indicate dependencies between these resources. Below the diagram is the template's JSON/YAML code:

```

1 AWSTemplateFormatVersion: 2010-09-09
2 Description: >-
3 This template generates the requested number of IAM users that will have the
4 same custom EC2 policy for EC2 resources tagged with Owner:devteam. The user
5 will be prompted to reset their passwords on next sign-in. Also an EC2

```

At the bottom, tabs for 'Components' and 'Template' are visible, along with a 'Feedback' link, language selection ('English (US)'), and copyright information.

- Returning to the select template screen, select upload a template to Amazon S3
- Navigate to the Microcosm Macro template and select open
- Press Next

The screenshot shows the 'Create stack' wizard in the AWS CloudFormation console. The top navigation bar includes 'CloudFormation' and 'Stacks'. The main area is titled 'Select Template' and contains the following steps:

- Select Template**
- Specify Details**
- Options**
- Review**

Below these steps is a 'Design a template' section with a 'Design template' button. Under 'Choose a template', it says: 'A template is a JSON/YAML-formatted text file that describes your stack's resources and their properties.' There are three options:

- Select a sample template
- Upload a template to Amazon S3
  - Choose File:
- Specify an Amazon S3 template URL

At the bottom right are 'Cancel' and 'Next' buttons.

- **IMPORTANT** name the stack being created from this template Macro . The

dynamic template being loaded next requires this naming for proper deployment and variable reference.

- Press Next

Create stack

**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

[Select Template](#) [Options](#) [Review](#)

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

- Press Next

1

**Permissions**

You can choose an IAM role that CloudFormation uses to create, modify, or delete resources in the stack. If you don't choose a role, CloudFormation uses the permissions defined in your account. [Learn more.](#)

IAM Role

**Rollback Triggers**

Rollback triggers enable you to have AWS CloudFormation monitor the state of your application during stack creation and updating, and to rollback that operation if the application breaches the threshold of any of the alarms you've specified. [Learn more.](#)

Monitoring Time  Minutes  
Minimum value of 0. Maximum value of 180.

Type	ARN (Amazon Resource Name)	Available triggers remaining: 5
1 AWS::CloudWatch::Alarm	<input type="text"/>	<a href="#">+</a>

**Advanced**

You can set additional options for your stack, like notification options and a stack policy. [Learn more.](#)

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

- Select the acknowledgement and press Create

Stack name: Macro

**Options**

Tags  
No tags provided

Rollback Triggers  
No monitoring time provided  
No rollback triggers provided

Advanced

Notification	Disabled
Termination Protection	Disabled
Timeout	none
Rollback on failure	Yes

**Capabilities**

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

This template contains Identity and Access Management (IAM) resources. Check that you want to create each of these resources and that they have the minimum required permissions. In addition, they have custom names. Check that the custom names are unique within your AWS account. [Learn more.](#)

I acknowledge that AWS CloudFormation might create IAM resources with custom names.

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

Cancel Previous Create

- On the following screen, you will see the Macro stack being created. While waiting for the creation to complete, explore the tabs giving details of the stacks and creation process

Stack Name	Created Time	Status	Drift Status	Description
Macro	2019-04-11 10:22:42 UTC-0600	CREATE_COMPLETE	NOT_CHECKED	

**Events**

Date	Status	Type	Logical ID	Status Reason
2019-04-11	CREATE_COMPLETE	AWS::CloudFormation::Stack	Macro	
10:23:17 UTC-0600	CREATE_COMPLETE	AWS::IAM::Policy	LambdaPolicy	
10:23:16 UTC-0600	CREATE_COMPLETE	AWS::CloudFormation::Macro	Macro	
10:23:11 UTC-0600	CREATE_IN_PROGRESS	AWS::CloudFormation::Macro	Macro	Resource creation initiated
10:23:10 UTC-0600	CREATE_IN_PROGRESS	AWS::CloudFormation::Macro	Macro	
10:23:08 UTC-0600	CREATE_COMPLETE	AWS::Lambda::Function	MacroProcessor	Resource creation initiated
10:23:08 UTC-0600	CREATE_IN_PROGRESS	AWS::Lambda::Function	MacroProcessor	Resource creation initiated
10:23:07 UTC-0600	CREATE_IN_PROGRESS	AWS::Lambda::Function	MacroProcessor	Resource creation initiated
10:23:07 UTC-0600	CREATE_IN_PROGRESS	AWS::IAM::Policy	LambdaPolicy	
10:23:04 UTC-0600	CREATE_COMPLETE	AWS::IAM::Role	LambdaExecutionRole	

Feedback English (US)

- Repeat the Stack creation process for the Microcosm Dynamic Template

Select Template

Select the template that describes the stack that you want to create. A stack is a group of related resources that you manage as a single unit.

Design a template Use AWS CloudFormation Designer to create or modify an existing template. [Learn more](#).

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  
Choose File: Microcosm\_Dy...on.template

Specify an Amazon S3 template URL

- Give the stack a name
- Enter the number of users to be created. If zero is set, 1 user and the associated VPC, etc are created. If 1.n.n are entered, 1...n *additional* users and VPCs are created (creating 2...n total users and VPCs). Remember not to exceed the number of VPCs in your limit, nor the number of elements in the template itself (the macro augments the template before it is deployed) or the stack creation will fail.

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

InstanceType	t2.micro	WebServer EC2 instance type
NumberOfUsers	0	Enter the number of users to create for the Dev Team
SSHLocation	0.0.0.0/0	Lockdown SSH access to the bastion host (default can be accessed from anywhere)

- Press Next

Permissions

You can choose an IAM role that CloudFormation uses to create, modify, or delete resources in the stack. If you don't choose a role, CloudFormation uses the permissions defined in your account. [Learn more.](#)

**IAM Role** Choose a role (optional)  
Enter role arn

**Rollback Triggers**

Rollback triggers enable you to have AWS CloudFormation monitor the state of your application during stack creation and updating, and to rollback that operation if the application breaches the threshold of any of the alarms you've specified. [Learn more](#)

**Monitoring Time** 0-180 Minutes  
Minimum value of 0. Maximum value of 180.

Type	ARN (Amazon Resource Name)	Available triggers remaining: 5
1 AWS::CloudWatch::Alarm	[Input Field]	[+]

**Advanced**

You can set additional options for your stack, like notification options and a stack policy. [Learn more.](#)

Cancel Previous Next

Feedback English (US) © 2008 - 2019, Amazon Web Services, Inc. or its affiliates. All rights reserved. Privacy Policy Terms of Use

- Check the boxes acknowledging the behaviour of the template
- Select Create Change Set. This initiates the lambda function in the Macro stack deployed previously

No rollback triggers provided

**Advanced**

**Notification** Disabled

**Termination Protection** Disabled

**Timeout** none

**Rollback on failure** Yes

**Capabilities**

**Transforms might require access capabilities**

A transform might add Identity and Access Management (IAM) resources that could provide entities access to make changes to your AWS account. If a transform adds IAM resources, you must acknowledge their capabilities to create or update them. Ensure that you want to create or update the IAM resources, and that they have the minimum required permissions. In addition, if they have custom names, check that the names are unique within your AWS account. [Learn more](#).

I acknowledge that AWS CloudFormation might create IAM resources.  
 I acknowledge that AWS CloudFormation might create IAM resources with custom names.

**Transforms**

**Check the following transforms: ["443007076818:Macro"]**

You must use a change set to create this stack because it includes one or more transforms. The change set shows the resources that transforms add to your stack's template. Choose Create Change Set, check the resources that the transforms add, and then choose Execute. [Learn more](#).

Create Change Set Cancel Previous Execute

Feedback English (US) © 2008 - 2019, Amazon Web Services, Inc. or its affiliates. All rights reserved. Privacy Policy Terms of Use

- Once the Change Set has been generated, select the Execute button

The screenshot shows the AWS CloudFormation console interface. A modal dialog box at the top right says: "Check the following transforms: ['443007076818::Macro']". Below it, a note states: "You must use a change set to create this stack because it includes one or more transforms. The change set shows the resources that transforms add to your stack's template. Choose Create Change Set, check the resources that the transforms add, and then choose Execute. [Learn more.](#)"

Below the modal, a table lists the resources to be added:

Action	Logical ID	Physical ID	Resource type	Replacement
Add	StudentEC2VPC		AWS::EC2::VPC	
Add	StudentUser		AWS::IAM::User	
Add	StudentVPCGatewayAttachment		AWS::EC2::VP CGatewayAttachment	
Add	StudentVPCInternetGateway		AWS::EC2::InternetGateway	
Add	StudentVPCRoute		AWS::EC2::Route	
Add	StudentVPCRoutingTable		AWS::EC2::RouteTable	
Add	StudentVPCSUBNET		AWS::EC2::Subnet	
Add	StudentVPCSecurityGroup		AWS::EC2::SecurityGroup	AWS::EC2::Subnet

At the bottom right of the screen are buttons for "Cancel", "Previous", and "Execute".

- When the creation is complete, observe the various elements created by both of the stack creation processes

- AWS Lambda Function Deployed**

- From the top services menu select Lambda
  - Select the MacroProcessor function

The screenshot shows the AWS Lambda service dashboard. On the left, a sidebar menu has "Functions" selected. The main content area shows a table of functions:

Function name	Description	Runtime	Code size	Last modified
MacroProcessor	IAM Users Macro processor function	Python 3.6	889 bytes	4 minutes ago

At the bottom right of the screen are buttons for "Feedback", "English (US)", "Privacy Policy", and "Terms of Use".

- Scroll down and observe the Lambda function code. This code is editable and when save is pressed and a change set is created for the Dynamic stack, the new version of the function will be run against the template (whether its the same template re-uploaded or the dynamic

template which has been changed)

The screenshot shows the AWS Lambda console interface for a function named "MacroProcessor". The "Function code" tab is selected. The code editor displays the Python script "index.py" which contains the lambda handler logic. The code uses deep copy operations to modify a template object based on user input.

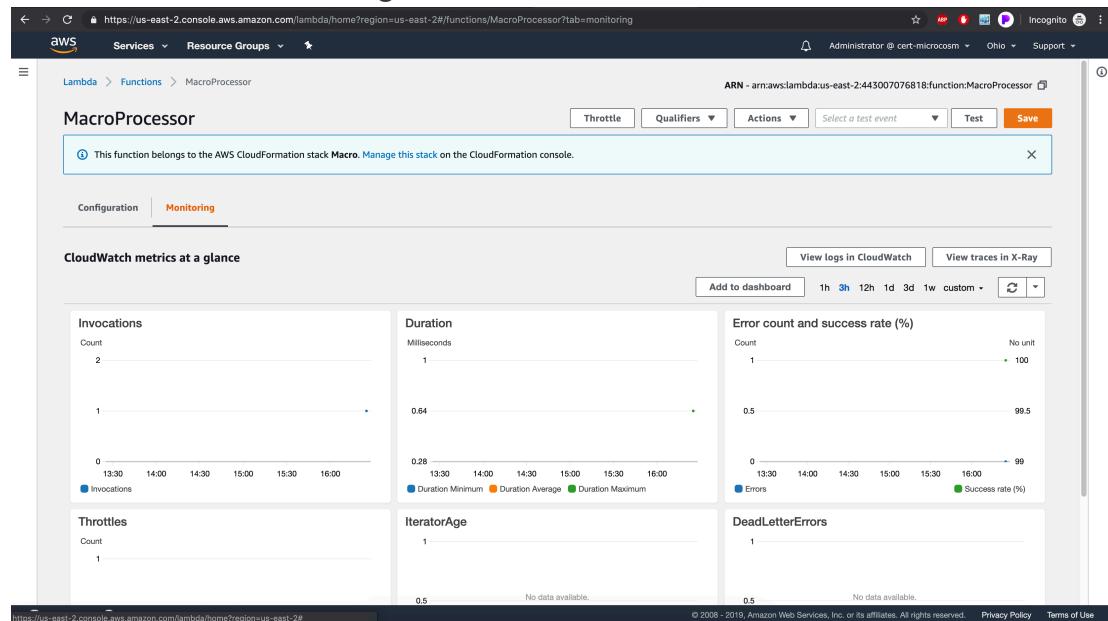
```
import json
import copy

def lambda_handler(event, context):
    FinalFragment = event["fagment"]
    Number = event["TemplateFragmentValues"]["NumberOfUsers"]
    # declare a new dictionary to have the resources object
    TemplateDetailsVar = {}
    TemplateDetailsVar = FinalFragment["Resources"]

    for i in range(int(Number)):
        # modify the IAM User resource logical id in order to add it later to "Resources" object the number of times requested
        numStr = str(i)
        TemplateDetailsVar["StudentUser"+numStr] = copy.deepcopy(TemplateDetailsVar["StudentUser"])
        TemplateDetailsVar["StudentEC2VPC"+numStr] = copy.deepcopy(TemplateDetailsVar["StudentEC2VPC"])
        TemplateDetailsVar["StudentVPCSUBNET"+numStr] = copy.deepcopy(TemplateDetailsVar["StudentVPCSUBNET"])
        TemplateDetailsVar["StudentVPCSecurityGroup"+numStr] = copy.deepcopy(TemplateDetailsVar["StudentVPCSecurityGroup"])
        TemplateDetailsVar["StudentVPCRoutingTable"+numStr] = copy.deepcopy(TemplateDetailsVar["StudentVPCRoutingTable"])

        dependencies.append(["{}","format(TempalteDetailsVar[\"StudentVPCRoutingTable\"]+numStr)[\"DependsOn\"]"])
        TempalteDetailsVar["StudentVPCRoutingTable"+numStr]["Properties"]["VpcId"] = "{}".format(TempalteDetailsVar["StudentVPCRoutingTable"+numStr]["Properties"]["VpcId"])
        TempalteDetailsVar["StudentVPCRoutingTable"+numStr]["Properties"]["DependsOn"] = dependencies
```

- Scrolling back up and selecting the Monitoring tab will allow you to see the lambda functions behaviours in graphical form, or you can select to view the CloudWatch logs



- After Selecting CloudWatch, select a LogStream representing the most recent code run

The screenshot shows the AWS CloudWatch service interface. The top navigation bar includes links for Services, Resource Groups, and a user account (Administrator @ cert-microcosm). Below the navigation is a search bar with options to 'Search Log Group', 'Create Log Stream', and 'Delete Log Stream'. A filter dropdown is set to 'Log Stream Name Prefix'. The main content area displays a table of log streams, each with a checkbox, a timestamp, and a link. The columns are 'Log Streams' and 'Last Event Time'. The table lists numerous log entries, such as '2019/04/11[\$LATEST]c94293df5fbc42eaea323d4e9e2a85f5e' and '2019/04/09[\$LATEST]j319574b1043d4b67b2249689fb7f10', all timestamped from April 2019.

Log Streams	Last Event Time
2019/04/11[\$LATEST]c94293df5fbc42eaea323d4e9e2a85f5e	2019-04-11 10:26 UTC-6
2019/04/09[\$LATEST]j72598d21254450850f589cd4e5b6	2019-04-09 12:44 UTC-6
2019/04/09[\$LATEST]j319574b1043d4b67b2249689fb7f10	2019-04-09 11:53 UTC-6
2019/04/08[\$LATEST]j7386ed3c3a7419ab1e8039ef7abac7e	2019-04-08 17:07 UTC-6
2019/04/08[\$LATEST]j4589na7392764fa238949a16c6bba	2019-04-08 17:03 UTC-6
2019/04/08[\$LATEST]j1408685438ba420c93ced1138d81816	2019-04-08 16:55 UTC-6
2019/04/08[\$LATEST]j484dd551c1439a8ab5168670e233ac	2019-04-08 16:46 UTC-6
2019/04/08[\$LATEST]da7eff41c04832a4bfba775494fe	2019-04-08 16:22 UTC-6
2019/04/08[\$LATEST]ccdd2f0fa040288a8c49ef2924fa	2019-04-08 16:12 UTC-6
2019/04/08[\$LATEST]j914ef63b1c4bf4bd63007e070eac7	2019-04-08 16:10 UTC-6
2019/04/08[\$LATEST]jd9513d7348259459a6cd9236113	2019-04-08 16:09 UTC-6
2019/04/08[\$LATEST]j24781ab0402408929861c427e288c1	2019-04-08 16:07 UTC-6
2019/04/08[\$LATEST]j44089c602d43fcfa161746a1355945	2019-04-08 16:04 UTC-6
2019/04/08[\$LATEST]j64621a210b046d887187653167b624	2019-04-08 15:59 UTC-6
2019/04/08[\$LATEST]j3266799eb64caab57047b8c835f6	2019-04-08 15:50 UTC-6
2019/04/08[\$LATEST]j709987b85684114aa3c3c076e9d75	2019-04-08 15:48 UTC-6
2019/04/08[\$LATEST]j70bbb01f65334d1bd8ead3e82f1bc18	2019-04-08 15:34 UTC-6
2019/04/08[\$LATEST]je08102e8274494b0c73d439cb65a	2019-04-08 15:22 UTC-6
2019/04/08[\$LATEST]j00684282d24295a4f30a902db6b2	2019-04-08 15:02 UTC-6
2019/04/08[\$LATEST]j425b029444442c2bf54d9552a162f	2019-04-08 14:47 UTC-6
2019/04/08[\$LATEST]jce50a33bd9b4ca6917e054f6e7172	2019-04-08 14:28 UTC-6
2019/04/08[\$LATEST]j4b3e3e1caba4290838fb5317116740a	2019-04-08 14:13 UTC-6
2019/04/08[\$LATEST]jdf0ffcc3cf59345dc8103a335055777	2019-04-08 14:11 UTC-6
2019/04/08[\$LATEST]jaf60fbdd7f5f169794edb355e664204	2019-04-08 13:29 UTC-6

- View the logs generated by the lambda function

The screenshot shows the AWS CloudWatch Log Events viewer. The left sidebar includes navigation links for CloudWatch Dashboards, Alarms, INSUFFICIENT OK, Billing, Events, Rules, Event Buses, Logs (which is selected), Insights, Metrics, and Favorites. The main content area displays log entries for the /aws/lambda/MacroProcessor stream. The top navigation bar shows the URL https://us-east-2.console.aws.amazon.com/cloudwatch/home?region=us-east-2#logEventViewer:group=/aws/lambda/MacroProcessor;stream=2019/04/11/\$LATEST and the AWS logo. The top right corner shows the user Administrator @ cert-microcosm, the region Ohio, and support links.

**CloudWatch Log Events**

CloudWatch > Log Groups > /aws/lambda/MacroProcessor > 2019/04/11/\$LATEST [c94293df5f8c42ea323d4e9e2a85f8c...]

Administrator @ cert-microcosm | Ohio | Support

Filter events

Time (UTC +00:00) Message

2019-04-11

No older events found at the moment. [Retry](#).

Time (UTC +00:00)	Message
16:26:34	START RequestId: f40b8d79-75d1-4182-a8eb-aed854299233 Version: \$LATEST
16:26:34	PRINT EVENT :
16:26:34	{ "region": "us-east-2", "accountId": "443007076818", "fragment": { "AWSTemplateFormatVersion": "2010-09-09", "Description": "This template generates the requested number of IAM users that can be used for testing.", "Metadata": { "AWS::CloudFormation::Interface": { "ParameterOrder": [ "Path", "LoginProfile", "Password", "PasswordResetRequired" ] } }, "Properties": { "Path": "/User", "LoginProfile": { "Password": "\$bzc\$UrFLwx9HFB-49eRtt!", "PasswordResetRequired": true } } } }
16:26:34	PRINT RESOURCES :
16:26:34	{ "StudentUser": { "Type": "AWS::IAM::User", "Properties": { "Path": "/", "LoginProfile": { "Password": "\$bzc\$UrFLwx9HFB-49eRtt!", "PasswordResetRequired": true } } } }
16:26:34	END RequestId: f40b8d79-75d1-4182-a8eb-aed854299233
16:26:34	REPORT RequestId: f40b8d79-75d1-4182-a8eb-aed854299233 Duration: 0.64 ms Billed Duration: 100 ms Memory Size: 128 MB Max Memory Used: 40 MB

No newer events found at the moment. [Retry](#).

## ◦ IAM User

- From the top services menu select IAM
  - View the Student user created and its properties
    - This user information can be given to anyone to allow them to login and access the allowed services

The screenshot shows the AWS IAM Users page. The left sidebar includes options like Dashboard, Groups, Users (which is selected), Roles, Policies, Identity providers, Account settings, and Credential report. The main content area displays a table with three users:

User name	Groups	Access key age	Password age	Last activity	MFA
Administrator	Administrators	Yesterday	Yesterday	Today	Not enabled
[REDACTED]		6 days	6 days	6 days	Not enabled
VPCCreation-StudentUser-WMCVKHLQ8WCF	None	None	Today	None	Not enabled

A red box highlights the text "Text" at the bottom right of the table.

## ◦ VPC

- View the VPC properties of the Student VPC created (IPv4 CIDR=10.0.0.0/24). Note, there are 2 VPCs here. One is the one we just created, the other is the default VPC that exists natively with all AWS accounts.
- Explore the Subnets, Route Tables, Internet Gateways, etc in the left sidebar menu that were created in association with the new VPC

The screenshot shows the AWS VPC Dashboard. The left sidebar lists various VPC-related resources: Virtual Private Cloud, Subnets, Route Tables, Internet Gateways, Egress Only Internet Gateways, DHCP Options Sets, Elastic IPs, Endpoints, Endpoint Services, NAT Gateways, Peering Connections, Security, Network ACLs, Security Groups, Virtual Private Network (VPN), Customer Gateways, and Virtual Private Gateways. The main content area displays a table of VPCs:

Name	VPC ID	State	IPv4 CIDR	IPv6 CIDR	DHCP options set	Main Route table	Main Network ACL	Ter
vpc-01ca865d17b887b66	available	10.0.0.0/24	-	dopt-5946034	rtb-0a1f862380212061b	acl-0a03d1f85b31db190	def	
vpc-9dd0c3f5	available	172.31.0....	-	dopt-5946034	rtb-d764bcfc	acl-597f8132	def	

[Return to Table of Contents](#)

## APPENDIX B

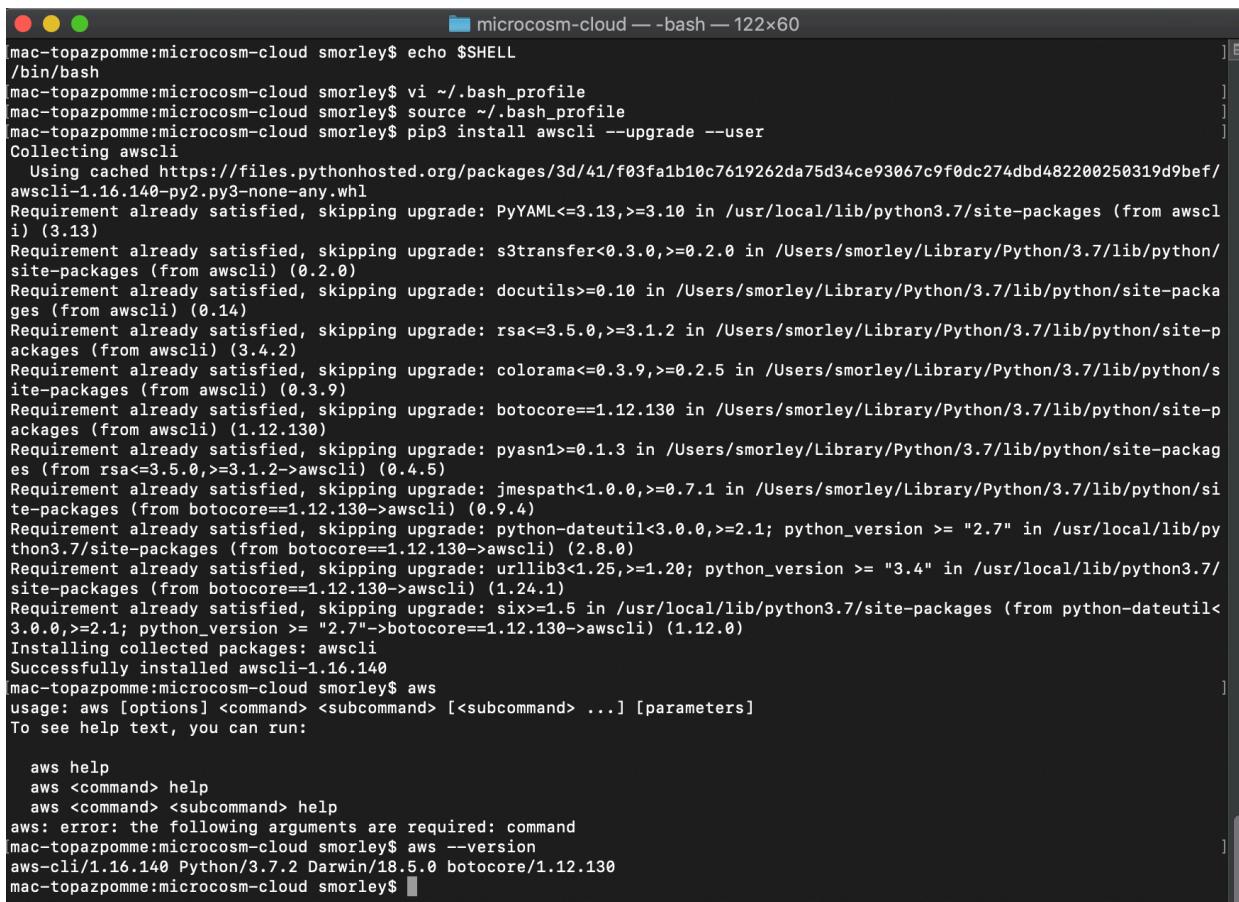
# AWS Command Line Interface (CLI) and Setting up Elastic Container Registry (ECR)

- If you haven't already, install python3 [making use of these instructions](#).
- Add the path to the python binaries to your `~/.bash_profile` in order to access the 'aws' after it is installed



```
microcosm-cloud — vi ~./.bash_profile — 122x60
export PATH=$PATH:/Users/smorley/Library/Python/3.7/bin
```

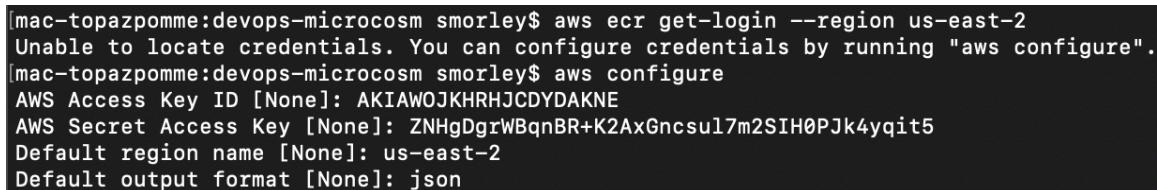
- Install the AWS CLI



```
microcosm-cloud — -bash — 122x60
mac-topazpomme:microcosm-cloud smorley$ echo $SHELL
/bin/bash
mac-topazpomme:microcosm-cloud smorley$ vi ~./.bash_profile
mac-topazpomme:microcosm-cloud smorley$ source ~./.bash_profile
mac-topazpomme:microcosm-cloud smorley$ pip3 install awscli --upgrade --user
Collecting awscli
  Using cached https://files.pythonhosted.org/packages/3d/41/f03fa1b10c7619262da75d34ce93067c9f0dc274dbd482200250319d9bef/
awscli-1.16.140-py2.py3-none-any.whl
Requirement already satisfied, skipping upgrade: PyYAML<=3.13,>=3.10 in /usr/local/lib/python3.7/site-packages (from awscli (3.13))
Requirement already satisfied, skipping upgrade: s3transfer<0.3.0,>=0.2.0 in /Users/smorley/Library/Python/3.7/lib/python/site-packages (from awscli) (0.2.0)
Requirement already satisfied, skipping upgrade: docutils>=0.10 in /Users/smorley/Library/Python/3.7/lib/python/site-packages (from awscli) (0.14)
Requirement already satisfied, skipping upgrade: rsa<=3.5.0,>=3.1.2 in /Users/smorley/Library/Python/3.7/lib/python/site-packages (from awscli) (3.4.2)
Requirement already satisfied, skipping upgrade: colorama<=0.3.9,>=0.2.5 in /Users/smorley/Library/Python/3.7/lib/python/site-packages (from awscli) (0.3.9)
Requirement already satisfied, skipping upgrade: botocore==1.12.130 in /Users/smorley/Library/Python/3.7/lib/python/site-packages (from awscli) (1.12.130)
Requirement already satisfied, skipping upgrade: pyasn1>=0.1.3 in /Users/smorley/Library/Python/3.7/lib/python/site-packages (from rsa<=3.5.0,>=3.1.2->awscli) (0.4.5)
Requirement already satisfied, skipping upgrade: jmespath<1.0.0,>=0.7.1 in /Users/smorley/Library/Python/3.7/lib/python/site-packages (from botocore==1.12.130->awscli) (0.9.4)
Requirement already satisfied, skipping upgrade: python-dateutil<3.0.0,>=2.1; python_version >= "2.7" in /usr/local/lib/python3.7/site-packages (from botocore==1.12.130->awscli) (2.8.0)
Requirement already satisfied, skipping upgrade: urlib3<1.25,>=1.20; python_version >= "3.4" in /usr/local/lib/python3.7/site-packages (from botocore==1.12.130->awscli) (1.24.1)
Requirement already satisfied, skipping upgrade: six>=1.5 in /usr/local/lib/python3.7/site-packages (from python-dateutil<3.0.0,>=2.1; python_version >= "2.7"->botocore==1.12.130->awscli) (1.12.0)
Installing collected packages: awscli
Successfully installed awscli-1.16.140
mac-topazpomme:microcosm-cloud smorley$ aws
usage: aws [options] <command> <subcommand> [<subcommand> ...] [parameters]
To see help text, you can run:

  aws help
  aws <command> help
  aws <command> <subcommand> help
aws: error: the following arguments are required: command
mac-topazpomme:microcosm-cloud smorley$ aws --version
aws-cli/1.16.140 Python/3.7.2 Darwin/18.5.0 botocore/1.12.130
mac-topazpomme:microcosm-cloud smorley$
```

- Login to AWS using your secret key - created when your IAM user was created (See IAM creation above)



```
mac-topazpomme:devops-microcosm smorley$ aws ecr get-login --region us-east-2
Unable to locate credentials. You can configure credentials by running "aws configure".
[mac-topazpomme:devops-microcosm smorley$ aws configure
AWS Access Key ID [None]: AKIAWOJKHRHJCDYDAKNE
AWS Secret Access Key [None]: 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
Default region name [None]: us-east-2
Default output format [None]: json
```

- Get Temporary AWS ECR Login credentials to use with Docker

```
[mac-topazpomme:devops-microcosm smorley$ aws ecr get-login --region us-east-2 --no-include-email
docker login -u AWS -p eyJYX1lsb2FkIjoiKzhsTwtC2krRjP0l3pVfcjBlajNvbWhPUgpMw50K01XnvJluUprHTGfdQtmZq2vBy3hYz29Xt0
SS1YzYkpDldGdvZ0kvahWxZzY4lEnaTQxFcaVJxemdWvnNsEc4Ymg3NDbvWmJtQfHqs8WmdrnZn1DbdBf1Qaqpw51nza0hPzkr5n09vuswtvVvBjWxzrZ
1EzwdnbClcydnF5bjY0Wdhau014b3F2bkdoV09hKzM4b2p1NwdbVwTMwMsXhwewpBpQncyNFBXVdd1cldKeG5Sbgt2envrSxhmqjFhz20ze5ksMkzhRTFydnh
3gvbVnR0w5Em2dzv1nt1LyUfhcm1pdKkNFJERZyAneAxZemChVsAz2U0F6MgZucl3dnJUSS9IzHJczFxvVc3hnSm0RXY5TuatoWZzb0prvF1S1h
N62ndWpb1Rxm0Y3am2dusehodujRn0F9Vkf1lNgv3NImefMaHbhYt2UxMzbUjVa1Rp1dm1M0k5L01WmTxh0bHrrk4vUi1S9qwaXnrGnxhMfcQkozv
GxaYtnyofvrcKzmnE5Bu01mdFnsL0NqNc2tUsd3aV1KemZlZwcyDzVzWzickowT3VtndrpWty3Rdrzb3jaMtrwRz3bkl1dUo55unpvefyK0pJtms0aEvDufn
JQ2x3sfuS05pRz0ExYhRfeZgq5cU5GeUJxR314bUrxeFyWd0o0Mkr0uy8vse1DttC3QWReL0UzNbNaEt1N2E5a3k5YzPVYXQzUGNxueFEUkjaaFyzQyd
TrkQ1VQwUdZUyyW4EfWlnJN0u24rfjw73l5by9IuXvhng9iWeQzxUhVhnxRc30u1Whn5aEvxdnp3bnhVca9tVuuu1svJrl1yZw9lgMzQwv9hwm9
FwmIxRhrjUjbhsu9sutzozGzmwGxrzs9zu9uRf11sJszKk3bE1yVzdcm09jMxdoNg1oUhpxnxnPyUhjK01oNfpob252z1Vrmt40GxmrvrweHfln01Iq1g5N
zddeErVjDnbksVzs9d1JhvHhsN9iuoxb1Lb1Vs1R5V1z2ek1Mvhu30T5MxbGaU5ck2pmeVvq0o42zxVmnhQwaCh2c5mwBkhhb1ByUyHnHuFwU5b5bUq
bmcxmuVwvao1uiuh6aUrzzM5tfNzGwNmavzsY21Dw0JxM0FmWpaZxbjUfJWm0FkCwvUy1Ekd3ntD2NycHnsZhCbu05yKj1UvDvrmZxanPrT2yZwpObzv
G6zU3RzbmxoSkpVmU9yUgxKtvh2bfUlcisZmkd4an1CmW3Y29KzDm2Ril3OrVkuUwUdfNmJtr2ZfrP0tvxMckJ3P0t6Lc3KjYXrh2aV61joiQvFFQkFia13L21
nd01nNe50D2f1chntsv14Nehmhb1R2MvdhriRhd2dRwtr11xwmdBqufInHdmQv1Lks29aswh2Y05Bwuhnb0c4d2JrsuJBREJQmdrcwraUc5dzBcQndFd0hnw
UpzSpvjqvDvREJbxrnVnQkvfreXrykvQzNwy0htR3c2b5ksBsuJfSue3UueWt1Mkpo0X15MmzeteTrd3j1m9nrgW0d1arNcsowIvcwm1kM0DPl1vai96T09
BnepiVpUhrh4M4QwfQwvRnmR99l1Viy1LfnLms3d2o1Lc2JZjwzaw9uijoiMiisInR5CuGuiojeQvrbx0fFwsiSIn4cGlyBpr24j0je1ntQ5NjQ40DF9
p
s://443007076818.dkr.ecr.us-east-2.amazonaws.com
```

- Docker login using received credentials

- Alternately - get login credentials and login in one step without spamming screen:

```
[mac-topazpomme:devops-microcosm smorley$ eval $(aws ecr get-login --region us-east-2 --no-include-email | sed 's|https://||' )  
WARNING! Using --password via the CLI is insecure. Use --password-stdin.  
Login Succeeded
```

- List Docker images to find which ones you want to upload

REPOSITORY	TAG	IMAGE ID	CREATED	SIZE
blacktip.ecru.cert.org/riplr/data_mart_loadims	latest	c400e2deb0f0	12 days ago	293MB
blacktip.ecru.cert.org/riplr/data_mart_loadims	v032019	c400e2deb0f0	12 days ago	293MB
data_mart_loadims	latest	c400e2deb0f0	12 days ago	293MB
blacktip.ecru.cert.org/riplr/data_mart_load	latest	754a9160dac1	12 days ago	950MB
blacktip.ecru.cert.org/riplr/data_mart_load	v032019	754a9160dac1	12 days ago	950MB
data_mart_load	latest	754a9160dac1	12 days ago	950MB
blacktip.ecru.cert.org/riplr/data_mart_web	latest	b49f18a39028	12 days ago	4.77GB
blacktip.ecru.cert.org/riplr/data_mart_web	v032019	b49f18a39028	12 days ago	4.77GB
data_mart_web	latest	b49f18a39028	12 days ago	4.77GB
gitlab/gitlab-ce	latest	9a2bee28183e	13 days ago	1.78GB
sonarqube	lts	6927219e0bd7	13 days ago	822MB
sonarqube	lts0419	6927219e0bd7	13 days ago	822MB

- Tag the image with the ECR repository url and image tracking name (eg latest)

```
mac-topazpomme:devops-microcosm smorley$ docker tag sonarqube:lts 443007076818.dkr.ecr.us-east-1.amazonaws.com/sonarqube:lts
```

- Verify tag is correct by relisting images

REPOSITORY	TAG	IMAGE ID	CREATED	SIZE
blacktip.ecru.cert.org/riplr/data_mart_loadims	latest	c400e2deb0f0	12 days ago	293MB
blacktip.ecru.cert.org/riplr/data_mart_loadims	v032019	c400e2deb0f0	12 days ago	293MB
data_mart_loadims	latest	c400e2deb0f0	12 days ago	293MB
blacktip.ecru.cert.org/riplr/data_mart_load	latest	754a9160dac1	12 days ago	950MB
blacktip.ecru.cert.org/riplr/data_mart_load	v032019	754a9160dac1	12 days ago	950MB
data_mart_load	latest	754a9160dac1	12 days ago	950MB
blacktip.ecru.cert.org/riplr/data_mart_web	latest	b49f18a39028	12 days ago	4.77G
B				
blacktip.ecru.cert.org/riplr/data_mart_web	v032019	b49f18a39028	12 days ago	4.77G
B				
data_mart_web	latest	b49f18a39028	12 days ago	4.77G
B				
gitlab/gitlab-ce	latest	9a2bee28183e	13 days ago	1.78G
B				
443007076818.dkr.ecr.us-east-1.amazonaws.com/sonarqube	lts	6927219e0bd7	13 days ago	822MB

- Create the desired ECR repository (1 repository per image type, in this example only sonarqube images) and push the image

```
mac-topazpomme:devops-microcosm smorley$ aws ecr create-repository --repository-name sonar
qube
{
    "repository": {
        "repositoryArn": "arn:aws:ecr:us-east-2:443007076818:repository/sonarqube",
        "registryId": "443007076818",
        "repositoryName": "sonarqube",
        "repositoryUri": "443007076818.dkr.ecr.us-east-2.amazonaws.com/sonarqube",
        "createdAt": 1554929452.0
    }
}
[mac-topazpomme:devops-microcosm smorley$ docker tag sonarqube:lts 443007076818.dkr.ecr.us-]
east-2.amazonaws.com/sonarqube:lts
[mac-topazpomme:devops-microcosm smorley$ docker push 443007076818.dkr.ecr.us-east-2.amazon]
aws.com/sonarqube:lts
The push refers to repository [443007076818.dkr.ecr.us-east-2.amazonaws.com/sonarqube]
68e5087b6ffc: Pushed
237afe62393f: Pushed
079758d1bbb7: Pushed
e11f2ab9e2f4: Pushed
f7d12d471667: Pushed
f350d0146bb3: Pushed
e38df31d449c: Pushed
af5ae4841776: Pushed
b17cc31e431b: Pushed
12cb127eee44: Pushed
604829a174eb: Pushed
fbba641a8b943: Pushed
lts: digest: sha256:8cbd208b264ab1404bce2ab16bcd6ba9a31a35e5b3e800c0d72295d719c087e8 size:
2839
```

- Repeat for as many images as desired to add to the container registry

```
mac-topazpomme:devops-microcosm smorley$ aws ecr create-repository --repository-name h1kkan/jenkins-docker
{
    "repository": {
        "repositoryArn": "arn:aws:ecr:us-east-2:443007076818:repository/h1kkan/jenkins-docker",
        "registryId": "443007076818",
        "repositoryName": "h1kkan/jenkins-docker",
        "repositoryUri": "443007076818.dkr.ecr.us-east-2.amazonaws.com/h1kkan/jenkins-docker",
        "createdAt": 1554931781.0
    }
}
mac-topazpomme:devops-microcosm smorley$ docker tag h1kkan/jenkins-docker:lts 443007076818.dkr.ecr.us-east-2.amazonaws.com/h1kkan/jenkins-docker:lts
mac-topazpomme:devops-microcosm smorley$ docker push 443007076818.dkr.ecr.us-east-2.amazonaws.com/h1kkan/jenkins-docker:lts
The push refers to repository [443007076818.dkr.ecr.us-east-2.amazonaws.com/h1kkan/jenkins-docker]
4717473633af: Pushed
01e56e46ef42: Pushed
0e21db749a1b: Pushed
1abb109c6333: Pushed
ea9254dfb470: Pushed
781f3534b8f0: Pushed
69ed8c046be0: Pushed
1cf9d4808e4e: Pushed
2b138287e104: Pushed
e674c0445366: Pushed
86cea145f8b3: Pushed
653304fe4cf3: Pushed
04b4fa9a597a: Pushed
0d790265fcc4: Pushed
d97f26ef1623: Pushed
e87475e5811f: Pushed
51e6d0a07219: Pushed
ba57bc494b22: Pushed
3443d6cf0f1f: Pushed
f3a38968d075: Pushed
a327787b3c73: Pushed
5bb0785f2eee: Pushed
lts: digest: sha256:a2a91ba9249a31a33ab9877bab42c7dea49955b30740c7c7beb6ffa124bc77ca size: 4925
```

- View repositories and their details in teh AWS ECR Console

<https://us-east-2.console.aws.amazon.com/ecr/get-started?region=us-east-2>

Amazon Container Services

Compute

# Amazon Elastic Container Registry

## Easily store, manage, and deploy container images

Amazon Elastic Container Registry (ECR) is a fully-managed container registry that makes it easy for developers to store, manage, and deploy container images.

**Create a repository**

**Get Started**

**Pricing (US)**

You pay only for the amount of data you store in your repositories and data transferred to the Internet.

[Learn more](#)

**How it works**

**Benefits**

Fully managed	Secure
Amazon ECR eliminates the need to operate and scale the infrastructure	Amazon ECR transfers your container images over HTTPS and

[Feedback](#) [English \(US\)](#)

© 2008 - 2019, Amazon Web Services, Inc. or its affiliates. All rights reserved. [Privacy Policy](#) [Terms of Use](#)

<https://us-east-2.console.aws.amazon.com/ecr/repositories?region=us-east-2>

Amazon Container Services

ECR > Repositories

### Repositories (5)

Find Repositories

Repository name	URI	Created at
gillax/hubot-slack-jenkins	443007076818.dkr.ecr.us-east-2.amazonaws.com/gillax/hubot-slack-jenkins	04/10/19, 3:08:49 PM
gitlab/gitlab-ce	443007076818.dkr.ecr.us-east-2.amazonaws.com/gitlab/gitlab-ce	04/10/19, 3:20:15 PM
h1kkan/jenkins-docker	443007076818.dkr.ecr.us-east-2.amazonaws.com/h1kkan/jenkins-docker	04/10/19, 3:29:41 PM
owasp/zap2docker-stable	443007076818.dkr.ecr.us-east-2.amazonaws.com/owasp/zap2docker-stable	04/10/19, 3:32:20 PM
sonarqube	443007076818.dkr.ecr.us-east-2.amazonaws.com/sonarqube	04/10/19, 2:50:52 PM
sonatype/nexus	443007076818.dkr.ecr.us-east-2.amazonaws.com/sonatype/nexus	04/10/19, 3:27:06 PM

[View push commands](#) [Delete](#) [Create repository](#)

https://us-east-2.console.aws.amazon.com/ecr/repositories?region=us-east-2

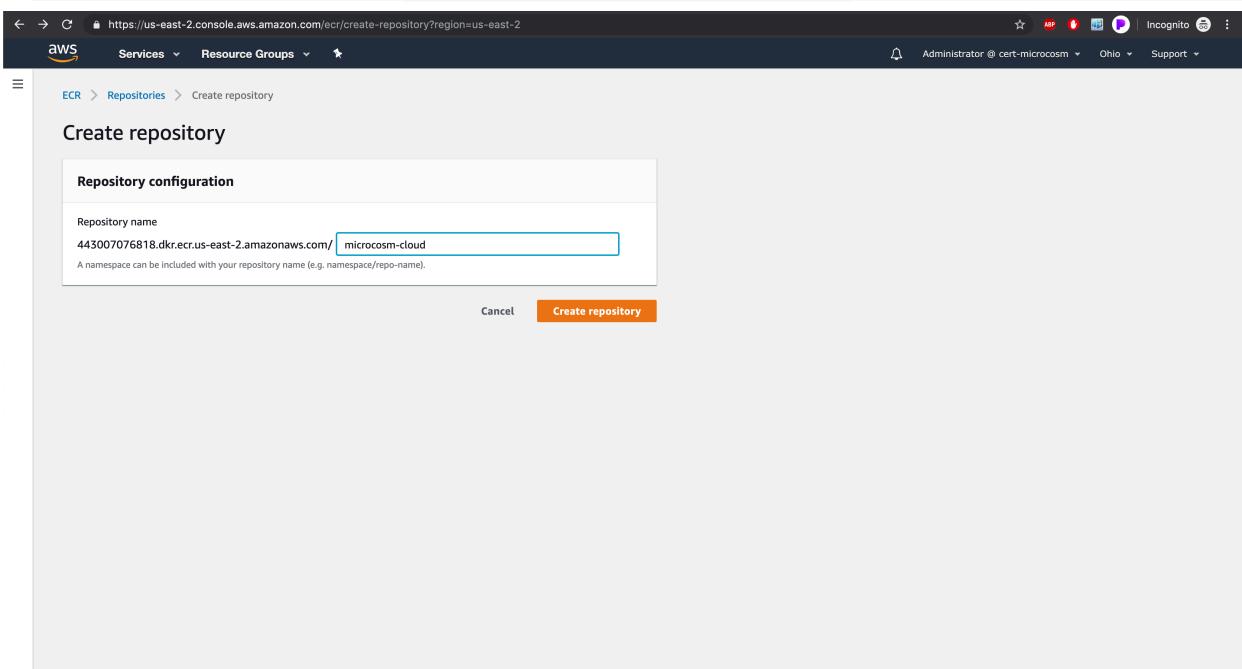
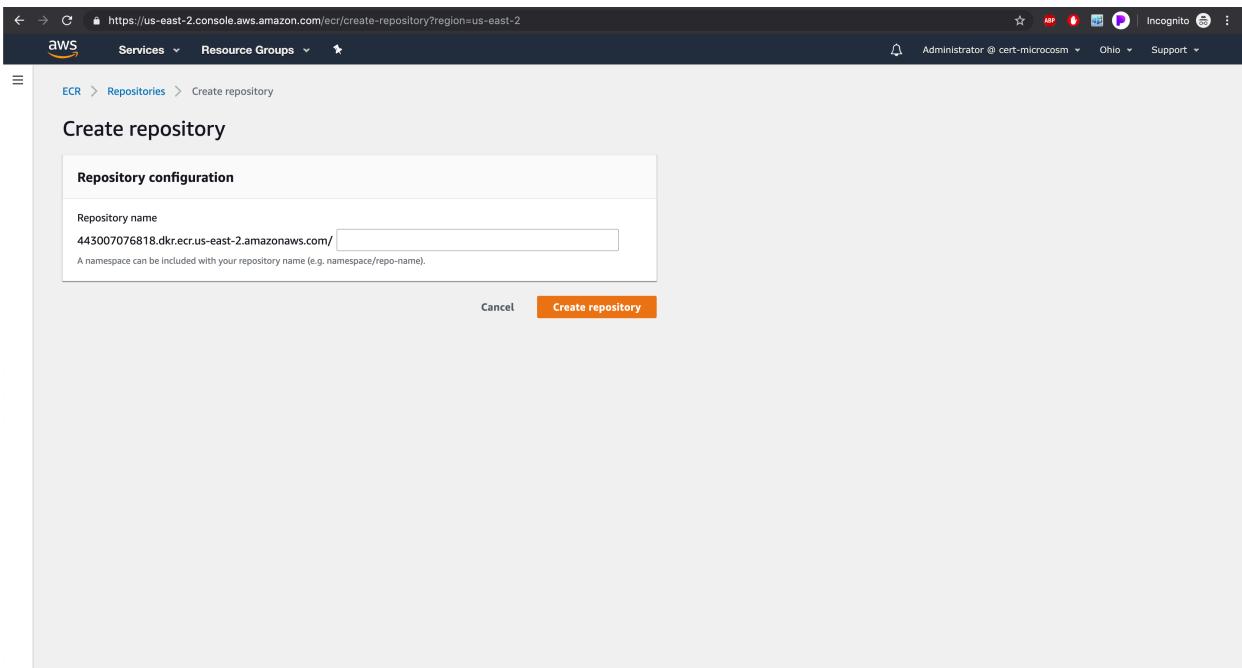
© 2008 - 2019, Amazon Web Services, Inc. or its affiliates. All rights reserved. [Privacy Policy](#) [Terms of Use](#)

The screenshot shows the AWS ECR console interface. On the left, there's a sidebar with navigation links for Amazon Container Services, Amazon ECS, Amazon EKS, and Amazon ECR. The main area is titled 'gillax/hubot-slack-jenkins' and shows a table of images. The table has columns for Image tag, Image URI, Pushed at, Digest, and Size (MB). One row is visible for the 'latest' tag.

Image tag	Image URI	Pushed at	Digest	Size (MB)
latest	443007076818.dkr.ecr.us-east-2.amazonaws.com/gillax/hubot-slack-jenkins:latest	04/10/19, 3:18:08 PM	sha256:4c04456cf...	285.24

- NOTE: it is possible to create repositories using the AWS ECR Console, but it is faster to do using the CLI and the terminal.

The screenshot shows the AWS ECR home page. It includes a 'Create a repository' button, a 'How it works' diagram illustrating the workflow from code to container, a 'Benefits' section listing 'Fully managed' and 'Secure', and a 'Getting started' sidebar with links to 'What is Amazon ECR?', 'Getting started with ECR', and 'Set up a CI/CD pipeline with ECR'.



The screenshot shows the AWS ECR (Amazon Elastic Container Registry) service in the AWS Management Console. A success message at the top says "Successfully created repository". The main area displays a list of repositories under the heading "Repositories (1)". The single repository listed is "microcosm-cloud", with its URI shown as "443007076818.dkr.ecr.us-east-2.amazonaws.com/microcosm-cloud". The interface includes standard AWS navigation elements like "View push commands", "Delete", and "Create repository".

[Return to Table of Contents](#)

## ECS Get Started Wizard

### Using Get Started Wizard

Follow the screen shots for reference

The screenshot shows the "Get Started" page for Amazon Elastic Container Service (ECS). The top features a video player with the title "Amazon Elastic Container Service (ECS)" and a play button showing "0:00 / 2:49". Below the video, a paragraph of text describes ECS's capabilities: "Amazon ECS makes it easy to deploy, manage, and scale Docker containers running applications, services, and batch processes. Amazon ECS places containers across your cluster based on your resource needs and is integrated with familiar features like Elastic Load Balancing, EC2 security groups, EBS volumes and IAM roles." A prominent blue "Get started" button is located below the text. At the bottom, three icons represent key features: "Run containers at scale" (multiple overlapping containers), "Flexible container placement" (a stack of containers with a gear icon), and "Integrated and extensible" (a building with a circular arrow icon).

Screenshot of the AWS ECS Clusters page. The left sidebar shows navigation options: Amazon ECS, Clusters (selected), Task Definitions, Amazon EKS, Clusters, Amazon ECR, Repositories, AWS Marketplace, Discover software, and Subscriptions. The main content area is titled "Clusters" and contains a message about the new ARN and resource ID format. It includes a "Configure ECS ARN setting" link and two buttons: "Create Cluster" and "Get Started". Below this, there's a "View" dropdown set to "list", a "Get Started" button, and a message stating "No clusters found". At the bottom right, there are navigation arrows and a "view all" link.

Screenshot of the "Getting Started with Amazon Elastic Container Service (Amazon ECS) using Fargate" wizard. The left sidebar lists steps: Step 1: Container and Task (selected), Step 2: Service, Step 3: Cluster, and Step 4: Review. The main content area starts with a diagram titled "Diagram of ECS objects and how they relate", showing a nested structure from Container definition down to Cluster. Below this, the "Container definition" step shows a list of pre-defined containers: "sample-app" (image: httpd:2.4, memory: 0.5GB (512), cpu: 0.25 vCPU (256)), "nginx" (image: nginx:latest, memory: 0.5GB (512), cpu: 0.25 vCPU (256)), "tomcat-webserver" (image: tomcat, memory: 2GB (2048), cpu: 1 vCPU (1024)), and "custom" (with a "Configure" button). At the bottom, there's a "Task definition" section with an "Edit" button.

The screenshot shows the AWS ECR (Amazon Container Registry) interface. On the left, there's a sidebar for 'Amazon Container Services' with sections for 'Amazon ECS', 'Amazon EKS', and 'Amazon ECR'. Under 'Amazon ECR', 'Images' is selected. The main area displays a table titled 'Images (1)' with one item: 'lts'.

Image tag	Image URI	Pushed at	Digest	Size (MB)
lts	443007076818.dkr.ecr.us-east-2.amazonaws.com/h1kkan/jenkins-docker:lts	04/10/19, 3:53:58 PM	sha256:a2a91ba92...	644.06

The screenshot shows the AWS ECS (Amazon Elastic Container Service) console. A modal window titled 'Edit container' is open. It has a 'Standard' tab selected. Inside, there are fields for 'Container name\*' (set to 'jenkins'), 'Image\*' (set to '443007076818.dkr.ecr.us-east-2.amazonaws.com/h1kkan/jenkins-docker:lts'), and 'Memory Limits (MiB)' (set to '128'). Below these, there's a note about hard and soft memory limits. The 'Port mappings' section shows a mapping from 'Container port 8080' to 'Protocol tcp'. At the bottom, there's a note about host port mappings being invalid for network mode. Buttons for 'Cancel' and 'Update' are at the bottom right.

The screenshot shows the AWS Elastic Container Service (ECS) console interface. The top part displays the 'Edit container' dialog, which includes sections for 'STORAGE AND LOGGING', 'Mount points', 'Volumes from', and 'Log configuration'. The 'Log configuration' section is set to 'Auto-configure CloudWatch Logs' with 'awslogs' as the log driver. The bottom part shows the 'Task definition' configuration page, where multiple containers are defined: 'sample-app', 'nginx', 'tomcat-webserver', and 'jenkins'. The 'jenkins' container is currently selected and has its configuration details visible. The task definition is named 'first-run-task-definition' and uses 'FARGATE' compatibility.

**STORAGE AND LOGGING**

Read only root file system

**Mount points**

Source volume	<none>
Container path	/var/jenkins_home
Read only	<input type="checkbox"/>

**Volumes from** Source container Read only

**Log configuration**  Auto-configure CloudWatch Logs

Log driver: awslogs

Log options:

Key: awslogs-group	Value: /ecs/first-run-task-definition
Key: awslogs-region	Value: us-east-2
Key: awslogs-stream-prefix	Value: ecs
Add key	Add value

**Container definition**

Choose an image for your container below.

**sample-app**  
image : httpd:2.4  
memory : 0.5GB (512)  
cpu : 0.25 vCPU (256)

**tomcat-webserver**  
image : tomcat  
memory : 2GB (2048)  
cpu : 1 vCPU (1024)

**Task definition**

\* Required

**Task definition**

A task definition is a blueprint for your application, and describes one or more containers through attributes. Some attributes are configured at the task level but the majority of attributes are configured per container.

Task definition name: first-run-task-definition

Network mode: awsvpc

Task execution role: Create new

Compatibilities: FARGATE

Task memory: 0.5GB (512)

Task CPU: 0.25 vCPU (256)

**jenkins** Configure

image : 443007076818.dkr.ecr.us-east-2.amazonaws.com/h1kkan/jenkins-docker:its  
memory :  
cpu :

**Task definition**

Task definition name: first-run-task-definition

Network mode: awsvpc

Task execution role: Create new

Compatibilities: FARGATE

Task memory: 0.5GB (512)

Task CPU: 0.25 vCPU (256)

**Required**

**Cancel** **Next**

Getting Started with Amazon Elastic Container Service (Amazon ECS) using Fargate

**Step 1: Container and Task**

**Step 2: Service**  

**Step 3: Cluster**

**Step 4: Review**

**Diagram of ECS objects and how they relate**

**Define your service**

A service allows you to run and maintain a specified number (the "desired count") of simultaneous instances of a task definition in an ECS cluster.

Service name  Edit

Number of desired tasks

Security group  Automatically create new  
A security group is created to allow all public traffic to your service only on the container port specified. You can further configure security groups and network access outside of this wizard.

Load balancer type  None  
 Application Load Balancer

\*Required Cancel Previous Next

Feedback English (US) © 2008 - 2019, Amazon Web Services, Inc. or its affiliates. All rights reserved. Privacy Policy Terms of Use

Getting Started with Amazon Elastic Container Service (Amazon ECS) using Fargate

**Step 1: Container and Task**

**Step 2: Service**

**Step 3: Cluster**  

**Step 4: Review**

**Diagram of ECS objects and how they relate**

**Configure your cluster**

The infrastructure in a Fargate cluster is fully managed by AWS. Your containers run without you managing and configuring individual Amazon EC2 instances.

To see key differences between Fargate and standard ECS clusters, see the [Amazon ECS documentation](#).

Cluster name   

Cluster names are unique per account per region. Up to 255 letters (uppercase and lowercase), numbers, hyphens, and underscores are allowed.

VPC ID  Automatically create new  

Subnets  Automatically create new  

\*Required Cancel Previous Next

Feedback English (US) © 2008 - 2019, Amazon Web Services, Inc. or its affiliates. All rights reserved. Privacy Policy Terms of Use

<https://us-east-2.console.aws.amazon.com/ecs/home?region=us-east-2#/firstRun>

Review

Task definition

Task definition name: first-run-task-definition

Network mode: awsvpc

Task execution role: Create new

Container name: jenkins

Image: 443007076818.dkr.ecr.us-east-2.amazonaws.com/h1kkan/jenkins-docker:its

Memory: 512

Port: 8080

Protocol: HTTP

Service

Service name: jenkins-service

Number of desired tasks: 1

Cluster

Cluster name: microcosm

VPC ID: Automatically create new

Subnets: Automatically create new

\*Required

Create

<https://us-east-2.console.aws.amazon.com/ecs/home?region=us-east-2#/firstRun>

Getting Started with Amazon Elastic Container Service (Amazon ECS) using Fargate

Launch Status

We are creating resources for your service. This may take up to 10 minutes. When we're complete, you can view your service.

Back View service Enabled after service creation completes successfully

Additional features that you can add to your service after creation

Scale based on metrics  
You can configure scaling rules based on CloudWatch metrics

Preparing service : 2 of 9 complete

ECS resource creation

- Cluster microcosm: complete
- Task definition first-run-task-definition:3: complete
- Service: pending

Additional AWS service integrations

- Log group: pending (The log group [ /ecs/first-run-task-definition ] already exists)
- CloudFormation stack: complete
- VPC: pending
- Subnet 1: pending
- Subnet 2: pending
- Security group: pending

Feedback English (US)

© 2008–2019, Amazon Web Services, Inc. or its affiliates. All rights reserved. Privacy Policy Terms of Use

Getting Started with Amazon Elastic Container Service (Amazon ECS) using Fargate

**Launch Status**

We are creating resources for your service. This may take up to 10 minutes. When we're complete, you can view your service.

[Back](#) [View service](#)

Additional features that you can add to your service after creation

**Scale based on metrics**  
You can configure scaling rules based on CloudWatch metrics

Preparing service : 9 of 9 complete

ECS resource creation ..... complete ✓  
Cluster [microcosm](#) ..... complete ✓  
Task definition [first-run-task-definition:3](#) ..... complete ✓  
Service [jenkins-service](#) ..... complete ✓

Additional AWS service integrations ..... complete ✓  
Log group [The log group \[/ecs/first-run-task-definition\]](#) already exists ..... complete ✓  
CloudFormation stack [EC2ContainerService-microcosm](#) ..... complete ✓  
VPC [vpc-0530acf15c751393](#) ..... complete ✓  
Subnet 1 [subnet-0521ea54a64435dd4](#) ..... complete ✓  
Subnet 2 [subnet-0e50c4d05af5052a7](#) ..... complete ✓  
Security group [sg-06f07d8d65ef6b2fd](#) ..... complete ✓

Feedback English (US)

https://us-east-2.console.aws.amazon.com/ecs/home?region=us-east-2#/clusters/microcosm/services/jenkins-service/details

Clusters > [microcosm](#) > Service: [jenkins-service](#)

**Service : jenkins-service**

[Update](#) [Delete](#)

Cluster	<a href="#">microcosm</a>	Desired count	1
Status	<a href="#">ACTIVE</a>	Pending count	1
Task definition	<a href="#">first-run-task-definition:3</a>	Running count	0
Service type	REPLICA		
Launch type	FARGATE		
Platform version	LATEST(1.3.0)		
Service role	AWSServiceRoleForECS		

[Details](#) [Tasks](#) [Events](#) [Auto Scaling](#) [Deployments](#) [Metrics](#) [Tags](#) [Logs](#)

### Load Balancing

Load Balancer Name	Container Name	Container Port
No load balancers		

### Network Access

Allowed VPC	<a href="#">vpc-0530acf15c751393</a>
Allowed subnets	<a href="#">subnet-0521ea54a64435dd4, subnet-0e50c4d05af5052a7</a>
Security groups*	<a href="#">sg-06f07d8d65ef6b2fd</a>
Auto-assign public IP	ENABLED

<https://us-east-2.console.aws.amazon.com/ecs/home?region=us-east-2#/clusters/microcosm/services/jenkins-service/deployments>

Deployment Id	Status	Desired count	Pending count	Running count	Created time	Updated time
ecs-svc/9223370481860...	PRIMARY	1	0	1	2019-04-11 08:46:40 -0600	2019-04-11 08:47:53 -0600

<https://us-east-2.console.aws.amazon.com/ecs/home?region=us-east-2#/clusters/microcosm/services/jenkins-service/tasks>

Task	Task Definition	Last status	Desired status	Group	Launch type	Platform version
a92d17a2-260c-4fdb-b88...	first-run-task-definition:3	RUNNING	RUNNING	service:jenkins-service	FARGATE	1.3.0

<https://us-east-2.console.aws.amazon.com/ecs/home?region=us-east-2#/clusters/microcosm/tasks/a92d17a2-260c-4fdb-b880-cc48376ea9ec/details>

Task : a92d17a2-260c-4fdb-b880-cc48376ea9ec

Details Tags Logs

Cluster: microcosm  
Launch type: FARGATE  
Platform version: 1.3.0  
Task definition: first-run-task-definition:3  
Group: service:jenkins-service  
Task role: None  
Last status: RUNNING  
Desired status: RUNNING  
Created at: 2019-04-11 08:46:48 -0600  
Started at: 2019-04-11 08:47:46 -0600

### Network

Network mode: awsvpc  
ENI Id: eni-03efb0f0edd0d969f2  
Subnet Id: subnet-0e50c4d05af5052a7  
Private IP: 10.0.1.191  
Public IP: 18.191.252.84  
Mac address: 06:5e:3e:bd:77:c4

### Containers

Last updated on April 11, 2019 8:48:52 AM (0m ago)

Name	Container Id	Status	Image	CPU Units	Hard/Soft memo...	Essential
jenkins	4ca0f74c-2b10-4f11-ad6f-fcab3ac20078	RUNNING	443007076818.dkr...	0	--/-	true

<https://us-east-2.console.aws.amazon.com/ecs/home?region=us-east-2#/taskDefinitions>

Create new Task Definition Create new revision Actions

Status: (ACTIVE) INACTIVE

Filter in this page < 1-1 > Page size 50

Task Definition	Latest revision status
first-run-task-definition	ACTIVE

The screenshot shows the AWS ECS Clusters page. On the left, there's a sidebar with links like Amazon ECS, Clusters (which is selected), Task Definitions, Amazon EKS, Clusters, Amazon ECR, Repositories, AWS Marketplace, Discover software, and Subscriptions. The main content area is titled 'Clusters' and contains a message about the new ARN and resource ID format. Below that are 'Create Cluster' and 'Get Started' buttons. The main view shows a cluster named 'microcosm' under 'FARGATE'. It displays metrics: 1 Services, 1 Running tasks, 0 Pending tasks. Under 'EC2', it shows 0 Services, 0 Running tasks, 0 Pending tasks, and 0 Container instances. There are also 'No data' sections for CPUUtilization and MemoryUtilization. At the bottom, there are 'Feedback', 'English (US)', and navigation links.

## Everything Created By Get Started Wizard

Follow the screen shots for reference

The screenshot shows the AWS CloudFormation Stacks page. The top navigation bar includes the AWS logo, Services, Resource Groups, and a 'CloudFormation' tab which is selected. A message box says 'The redesigned AWS CloudFormation console is available now' and encourages feedback. The main table lists one stack: 'EC2ContainerService-microcosm' created on 2019-04-11 08:45:29 UTC-0600 with a status of 'CREATE\_COMPLETE'. The table has columns for Stack Name, Created Time, Status, Drift Status, and Description. Below the table, tabs for Overview, Outputs, Resources, Events, Template, Parameters, Tags, Stack Policy, Change Sets, and Rollback Triggers are visible. A message at the bottom says 'Select a stack'.

<https://us-east-2.console.aws.amazon.com/cloudformation/home?region=us-east-2#/stack/detail?stackId=arn:aws:cloudformation:us-east-2:443007076818:stack%2FEC2Contain...>

CloudFormation Services Resource Groups

CloudFormation Stacks Stack Detail

### EC2ContainerService-microcosm

Stack name: EC2ContainerService-microcosm

Stack ID: arn:aws:cloudformation:us-east-2:443007076818:stack/EC2ContainerService-microcosm/73623180-5c68-11e9-9551-06faad620df0

Status: CREATE\_COMPLETE

Status reason:

Termination protection: Disabled

Drift status: NOT\_CHECKED [View details](#)

Last drift check time:

IAM role:

Description: AWS CloudFormation template to create a new ECS Fargate First Run stack

Outputs

Key	Value	Description	Export Name
Version	3.0.0	ECS Cloudformation template version	
EcsElbName		Load Balancer for ECS Service	

Resources

To view detailed drift information for specific resources, visit the [Drift Details page](#).

Logical ID	Physical ID	Type	Drift Status	Status	Status Reason
AttachGateway	EC2Co-AttaC-1AVZP5KOSFL1P	AWS::EC2::VPCGatewayAttachment	NOT_CHECKED	CREATE_COMPLETE	
EcsSecurityGroup	sg-06f07c8d65ef6b2fd	AWS::EC2::SecurityGroup	NOT_CHECKED	CREATE_COMPLETE	
InternetGateway	igw-0db5c3b0358004693	AWS::EC2::InternetGateway	NOT_CHECKED	CREATE_COMPLETE	
InternetGateway	igw-0db5c3b0358004693	AWS::EC2::InternetGateway	NOT_CHECKED	CREATE_COMPLETE	
PublicRouteVialgw	EC2Co-Publi-1QU2JZOBMR836	AWS::EC2::Route	NOT_CHECKED	CREATE_COMPLETE	
PublicSubnet1RouteTable...	rtbassoc-07dc01864197254f	AWS::EC2::SubnetRouteTableAssociation	NOT_CHECKED	CREATE_COMPLETE	
PublicSubnet2RouteTable...	rtbassoc-05c00117481d05ba1	AWS::EC2::SubnetRouteTableAssociation	NOT_CHECKED	CREATE_COMPLETE	
PublicSubnetAz1	subnet-0521ea54a64435dd4	AWS::EC2::Subnet	NOT_CHECKED	CREATE_COMPLETE	
PublicSubnetAz2	subnet-0e50c4d05af5052a7	AWS::EC2::Subnet	NOT_CHECKED	CREATE_COMPLETE	
RouteVialgw	rtb-0f28d4e034944d0fc	AWS::EC2::RouteTable	NOT_CHECKED	CREATE_COMPLETE	
Vpc	vpc-0530acf15c751393	AWS::EC2::VPC	NOT_CHECKED	CREATE_COMPLETE	

Events

Filter by: Status

Date	Status	Type	Logical ID	Status Reason
2019-04-11	CREATE_COMPLETE	AWS::CloudFormation::Stack	EC2ContainerService-microcosm	
08:46:35 UTC-0600	CREATE_COMPLETE	AWS::EC2::SubnetRouteTableAssociation	PublicSubnet1RouteTableAssociation	
08:46:30 UTC-0600	CREATE_COMPLETE	AWS::EC2::SubnetRouteTableAssociation	PublicSubnet2RouteTableAssociation	
08:46:29 UTC-0600	CREATE_COMPLETE	AWS::EC2::Route	PublicRouteVialgw	
08:46:15 UTC-0600	CREATE_IN_PROGRESS	AWS::EC2::SubnetRouteTableAssociation	PublicSubnet1RouteTableAssociation	Resource creation initiated
08:46:15 UTC-0600	CREATE_IN_PROGRESS	AWS::EC2::SubnetRouteTableAssociation	PublicSubnet2RouteTableAssociation	Resource creation initiated
08:46:14 UTC-0600	CREATE_IN_PROGRESS	AWS::EC2::SubnetRouteTableAssociation	PublicSubnet1RouteTableAssociation	
08:46:14 UTC-0600	CREATE_IN_PROGRESS	AWS::EC2::Route	PublicRouteVialgw	Resource creation initiated
08:46:13 UTC-0600	CREATE_IN_PROGRESS	AWS::EC2::Route	PublicRouteVialgw	
08:46:10 UTC-0600	CREATE_COMPLETE	AWS::EC2::Subnet	PublicSubnetAz1	
08:46:10 UTC-0600	CREATE_COMPLETE	AWS::EC2::Subnet	PublicSubnetAz2	
08:46:09 UTC-0600	CREATE_COMPLETE	AWS::EC2::VPCGatewayAttachment	AttachGateway	
08:45:59 UTC-0600	CREATE_COMPLETE	AWS::EC2::SecurityGroup	EcsSecurityGroup	

<https://us-east-2.console.aws.amazon.com/cloudformation/home?region=us-east-2#/stack/detail?stackId=arn:aws:cloudformation:us-east-2:443007076818:stack%2FEC2Contain...>

08:45:29 UTC-0600 CREATE\_IN\_PROGRESS AWS::CloudFormation::Stack EC2ContainerService-microcos User Initiated

Template

Parameters

Key	Value	Resolved Value
AsgMaxSize	1	
CreateElasticLoadBalancer	false	
EcsAmIId	ami-044120f0dd7ed0fb4	
EcsClusterName	microcosm	
EcsEndpoint		
EcsInstanceType	t2.micro	
EcsPort	8080	
ElbHealthCheckTarget	HTTP:80/	
ElbPort	80	
IamRoleInstanceProfile	ecsInstanceRole	
IsFargate	true	
KeyName		
SourceCidr	0.0.0.0/0	
SubnetCidrBlock1	10.0.0.0/24	
SubnetCidrBlock2	10.0.1.0/24	
TargetGroupName	ECSFirstRunTargetGroup	
TargetType	ip	
VpcAvailabilityZones	us-east-2a,us-east-2b,us-east-2c	
VpcCidrBlock	10.0.0.0/16	

<https://us-east-2.console.aws.amazon.com/cloudformation/home?region=us-east-2#/stack/detail?stackId=arn:aws:cloudformation:us-east-2:443007076818:stack%2FEC2Contain...>

AMSTemplateFormatVersion: '2010-09-09'  
Description: AWS CloudFormation template to create a new ECS Fargate First Run stack  
Parameters:

```

EcsAmIId:
  Type: String
  Description: ECS AMI Id
EcsInstanceType:
  Type: String
  Description: ECS EC2 instance type
  Default: t2.micro
  ConstraintDescription: must be a valid EC2 instance type.
KeyName:
  Type: String
  Description: Optional - Name of an existing EC2 KeyPair to enable SSH access to the ECS instances
  Default: ''
AsgMaxSize:
  Type: Number
  Description: Maximum size and initial Desired Capacity of ECS Auto Scaling Group
  Default: '1'
IamRoleInstanceProfile:
  Type: String
  Description: Name or the Amazon Resource Name (ARN) of the instance profile associated with the IAM role for the instance
EcsClusterName:
  Type: String
  Description: ECS Cluster Name
  Default: default
EcsPort:
  Type: String
  Description: Optional - Security Group port to open on ECS instances - defaults to port 80
  Default: '80'
ElbPort:
  Type: String
  Description: Optional - Security Group port to open on ELB - port 80 will be open by default
  Default: '80'
ElbHealthCheckTarget:
  Type: String
  Description: Optional - Health Check Target for ELB - defaults to HTTP:80/
  Default: HTTP:80/

```

**Cluster : microcosm**

Status: ACTIVE

Service Name	Status	Service type	Task Definition	Desired tasks	Running tasks	Launch type	Platform version
jenkins-service	ACTIVE	REPLICAS	first-run-task-de...	1	1	FARGATE	LATEST(1.3.0)

**Service Health**

Current Status	Details
Amazon EC2 - US East (Ohio)	Service is operating normally View complete service health details

**Account Attributes**

- Resource ID length management

**Additional Information**

- VPC Documentation
- All VPC Resources
- Forums
- Report an Issue

**Site-to-Site VPN Connections**

Amazon VPC enables you to use your own isolated resources within the AWS cloud, and then connect those resources directly to your own datacenter using industry-standard encrypted IPsec VPN connections.

**Feedback** **English (US)**

Name	VPC ID	State	IPv4 CIDR	IPv6 CIDR	DHCP options set	Main Route table	Main Network ACL
ECS micro...	vpc-0530acf15c751393	available	10.0.0.0/16	-	dopt-5f946034	rtb-089b8ba33139a46fa	acl-07f738971cf93937c
	vpc-9dd0c3f5	available	172.31.0....	-	dopt-5f946034	rtb-d764bc	acl-597fb132

Name	Subnet ID	State	VPC	IPv4 CIDR	Available IPv4	IPv6 CIDR	Availability Zone	Availability Zone ID	Route table
ECS micro...	subnet-0521ea54a64435dd4	available	vpc-0530acf15c751393	10.0.0.0/24	250	-	us-east-2a	use2-az1	rtb-0
ECS micro...	subnet-0e50c4d05af5052a7	available	vpc-0530acf15c751393	10.0.1.0/24	251	-	us-east-2b	use2-az2	rtb-0
	subnet-0ecabb74	available	vpc-9dd0c3f5	172.31.16.0/20	4091	-	us-east-2b	use2-az2	rtb-d
	subnet-73d1ea1b	available	vpc-9dd0c3f5	172.31.0.0/20	4091	-	us-east-2a	use2-az1	rtb-d
	subnet-f1cd15bd	available	vpc-9dd0c3f5	172.31.32.0/20	4091	-	us-east-2c	use2-az3	rtb-d

The screenshot shows the AWS VPC Route Tables page. The left sidebar navigation includes: VPC Dashboard, Filter by VPC, Virtual Private Cloud, Your VPCs, Subnets, Route Tables (selected), Internet Gateways, Egress Only Internet Gateways, DHCP Options Sets, Elastic IPs, Endpoints, Endpoint Services, NAT Gateways, Peering Connections, Security, Network ACLs, Security Groups, Virtual Private Network (VPN), Customer Gateways, and Virtual Private Gateways.

The main content area displays a table of route tables:

Name	Route Table ID	Explicitly Associated with	Main	VPC ID	Owner
rtb-089b8ba33139a46fa	rtb-089b8ba33139a46fa	-	Yes	vpc-0530acf15c751393   ...	443007076818
ECS microc...	rtb-0f28d4e934944d0fc	2 subnets	No	vpc-0530acf15c751393   ...	443007076818
	rtb-d764bcbe	-	Yes	vpc-9dd0c3f5	443007076818

At the bottom of the page, there is a note: "Select an internet gateway above".

Below the table, the URL is https://us-east-2.console.aws.amazon.com/vpc/home?region=us-east-2#RouteTables:sort=routeTableId

The screenshot shows the AWS VPC Internet Gateways page. The left sidebar navigation is identical to the previous page.

The main content area displays a table of internet gateways:

Name	ID	State	VPC	Owner
ECS microc...	igw-0db5c3b0358...	attached	vpc-0530acf15c7...	443007076818
	igw-d403bebc	attached	vpc-9dd0c3f5	443007076818

At the bottom of the page, there is a note: "Select an internet gateway above".

Below the table, the URL is https://us-east-2.console.aws.amazon.com/vpc/home?region=us-east-2#EgressOnlyInternetGateways

Name	Group ID	Group Name	VPC ID	Type	Description	Owner
ECS microc...	sg-0f84b4941885...	EC2ContainerSer...	vpc-0530acf15c7...	EC2-VPC	ECS Allowed Ports	443007076818
	sg-0f84b4941885...	default	vpc-0530acf15c7...	EC2-VPC	default VPC secur...	443007076818
	sg-1e087170	default	vpc-9dd0c3f5	EC2-VPC	default VPC secur...	443007076818

The screenshot shows the AWS ECS Task Definitions page. On the left, there's a sidebar with links for Amazon ECS, Clusters, Task Definitions (which is selected and highlighted in orange), Amazon EKS, Clusters, Amazon ECR, Repositories, AWS Marketplace, Discover software, and Subscriptions. The main content area has a title 'Task Definitions' and a sub-instruction: 'Task definitions specify the container information for your application, such as how many containers are part of your task, what resources they will use, how they are linked together, and which host ports they will use.' Below this is a 'Learn more' link. At the top right, there are buttons for 'Create new Task Definition', 'Create new revision', and 'Actions'. To the right of these is a timestamp: 'Last updated on April 11, 2019 8:59:55 AM (0m ago)'. On the far right are icons for refresh, help, and search. A filter bar says 'Filter in this page' and a dropdown shows 'Latest revision status'. A table lists one task definition:

Task Definition	Latest revision status
first-run-task-definition	ACTIVE

At the bottom, there are links for 'Feedback', 'English (US)', and legal notices: '© 2006 - 2019, Amazon Web Services, Inc. or its affiliates. All rights reserved.', 'Privacy Policy', and 'Terms of Use'.

## Deleting an ECS Service and Associated Tasks

- Navigate to the ECS Console and select your cluster

# Clusters

An Amazon ECS cluster is a regional grouping of one or more container instances on which you can run tasks.

For more information, see the [ECS documentation](#).

## **i** Opt in to the new ARN and resource ID format

Amazon ECS has introduced a new format for ARNs and resource IDs. The ARN format is now consistent across all AWS services.

[Configure ECS ARN setting](#) 

**Create Cluster**

**Get Started**

**View**

 list

 card

**microcosm >**

FARGATE

**10**

Services

**4**

Running tasks

EC2

**0**

Services

**0**

Running tasks

- Scroll down and select your service

Clusters > microcosm

# Cluster : microcosm

Get a detailed view of the resources on your cluster.

		Status	ACTIVE
Registered container instances	0		
Pending tasks count	0	Fargate, 0 EC2	
Running tasks count	4	Fargate, 0 EC2	
Active service count	10	Fargate, 0 EC2	
Draining service count	0	Fargate, 0 EC2	

Services	Tasks	ECS Instances	Metrics	Scheduled Tasks	Tags
<a href="#">Create</a>	<a href="#">Update</a>	<a href="#">Delete</a>	<a href="#">Actions ▾</a>		
<input type="text"/> Filter in this page		Launch type	ALL	Service type	All
<input type="checkbox"/> Service Name				Status	
<input type="checkbox"/> sonarqube				ACTIVE	
<input type="checkbox"/> gitlab				ACTIVE	
<input type="checkbox"/> hubot				ACTIVE	
<input type="checkbox"/> owaspzap				ACTIVE	
<input type="checkbox"/> nagios				ACTIVE	
<input type="checkbox"/> sonatype_nexus				ACTIVE	
<input type="checkbox"/> cloud-discovery				ACTIVE	

- Select update (either from the services menu or the service's description page)

<a href="#">Clusters</a> > <a href="#">microcosm</a> > Service: cloud-discovery	
Service : cloud-discovery	
Cluster	microcosm
Status	ACTIVE
Task definition	cloud-discovery:1
Desired count	1
Pending count	0
Running count	1
	<a href="#">Update</a> <a href="#">Delete</a>

- Set the Number of Tasks to 0 and press Skip to review

Number of tasks	<input type="text" value="0"/>	<a href="#">i</a>
Minimum healthy percent	<input type="text" value="100"/>	<a href="#">i</a>
Maximum percent	<input type="text" value="200"/>	<a href="#">i</a>

---

[Cancel](#) [Skip to review](#)

- Press Update service and the View Service

[Cancel](#) [Previous](#) [Update Service](#)

- Once the Running count of the tasks has dropped to 0 ,

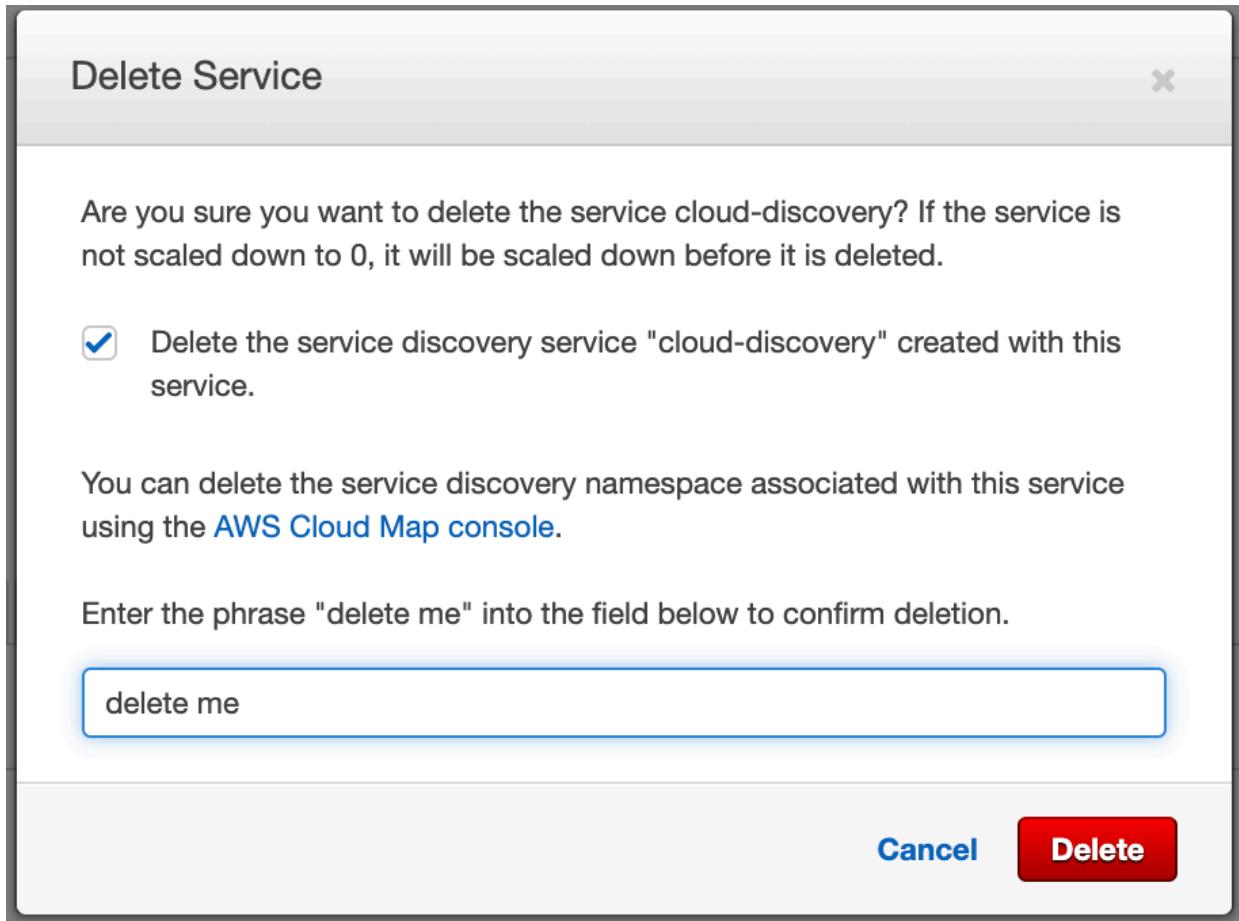
<a href="#">Filter in this page</a>						
Deployment Id	Status	Desired count	Pending count	Running count	Created time	Updated time
ecs-svc/9223370480216347022	PRIMARY	0	0	0	2019-04-30 09:33:48 -0600	2019-05-01 09:15:06 -0600

- Press Delete in the top right corner

[Update](#) [Delete](#)

- Check the Delete Discovery box and enter delete me in the prompt and press

## Delete



### Notes:

- Deleting the services does not remove the Task Definition.
- Stopping a task not associated with a service will remove that task but not affect any other service

[Return to Table of Contents](#)

## APPENDIX C

### Deploy Stand Alone Microcosm Template

- In Cloud Formation, Create a new stack using the `MicrocosmComponents_Standalone_AWS.template.yaml` File
- Follow the prompts on the screen and enter the slack hubot information [ref](#)

- Follow the instructions in the main section of this document to configure each system in the pipeline

[Return to Table of Contents](#)

## Deploy Stand Alone Petclinic Deployment Template

- [Create Ec2 Key Pair](#)
- In Cloud Formation, Create a new stack using the `MicrocosmDeployment_Standalone_AWS.template.yaml` File
- Follow the prompts on the screen and enter the slack hubot information [ref](#)
- Follow the instructions in the main section of this document to configure each system in the pipeline

Due to timeout issues with creating and then immediately using an IAM Instance profile via CloudFormation with an EC2 instance, instructions are included here to attach an IAM Role to the EC2 instances here.

The screenshot shows the AWS Lambda console interface. At the top, there are buttons for "Launch Instance", "Connect", and "Actions". A dropdown menu is open under "Actions", listing various instance management options. The "Instance Settings" option is highlighted. Below the menu, a table provides detailed information about a specific instance:

Attribute	Value
AMI ID	Amazon Linux 2.0.2019.06.04.1 (HVM, SSD) gp2 (2019-06-04)
Platform	-
IAM role	-
Key pair name	Administrator1
Owner	443007076818
Launch time	May 6, 2019 at 4:04:33 PM UTC-6 (less than one hour)
Termination protection	False
Lifecycle	normal
Monitoring	basic
Alarm status	None
Kernel ID	-
RAM disk ID	-
Placement group	-
Partition number	-

Roles > EC2PlusS3RoleTST

## Summary

<b>Role ARN</b>	arn:aws:iam::443007076818:role/EC2PlusS3RoleTST
<b>Role description</b>	<a href="#">Edit</a>
<b>Instance Profile ARNs</b>	arn:aws:iam::443007076818:instance-profile/EC2PlusS3RoleTST
<b>Path</b>	/
<b>Creation time</b>	2019-05-06 14:30 MDT
<b>Maximum CLI/API session duration</b>	1 hour <a href="#">Edit</a>

**Permissions** [Trust relationships](#) [Tags](#) [Access Advisor](#) [Revoke sessions](#)

▼ Permissions policies (3 policies applied)

[Attach policies](#)

Policy name
▶  AmazonEC2FullAccess
▶  AmazonS3FullAccess
▶  CloudWatchLogsFullAccess

▶ Permissions boundary (not set)

Instances > Attach/Replace IAM Role

## Attach/Replace IAM Role

Select an IAM role to attach to your instance. If you don't have any IAM roles, choose Create new IAM role to create a role in the IAM console. If an IAM role is already attached to your instance, the IAM role you choose will replace the existing role.

Instance ID i-05f85871d0aca39a8 (tomcat\_codedeployTST) [i](#)

IAM role\*  [▼](#) [C](#) [Create new IAM role](#) [i](#)

[Return to Table of Contents](#)

# Applying Layer Templates

####In AWS CloudFormation

1. See the instructions above for creating stacks using the Layer 1 templates ([Macro and Dynamic](#)). Important, remember the stack name you assigned to the Dynamic Stack.
2. Create a new stack using the Layer2 template (for guidance see [Deploy Stand Alone Microcosm Template](#). When prompted, enter the stack name of the Layer 1 Dynamic Template. Important, remember the stack name you assigned to the Layer 2 Stack.
3. Create a new stack using the Layer3 template (for guidance see [Deploy Stand Alone Petclinic Deployment Template](#). When prompted, enter the stack names of the Layer 1 Dynamic Stack and the Layer 2 Stack.

**Note Creation can take approx 30 minutes for all layers**

**Note, when deleting all the stacks, delete the most recently added first and WAIT until it's deleted to delete the next layer (otherwise it won't delete as other stacks require its resources). Delete takes about 15 minutes.**

[Return to Table of Contents](#)