

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

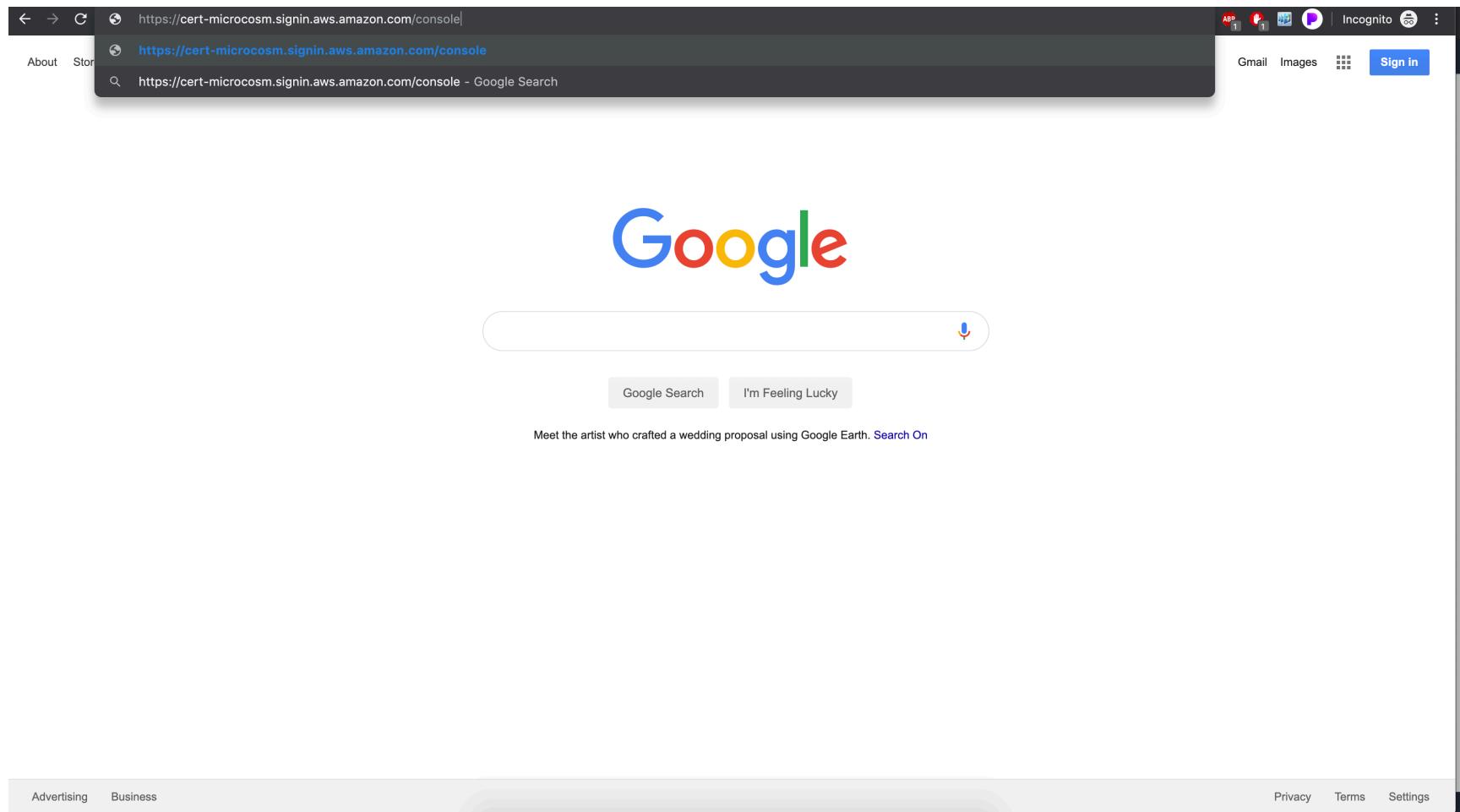
Introduction

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.

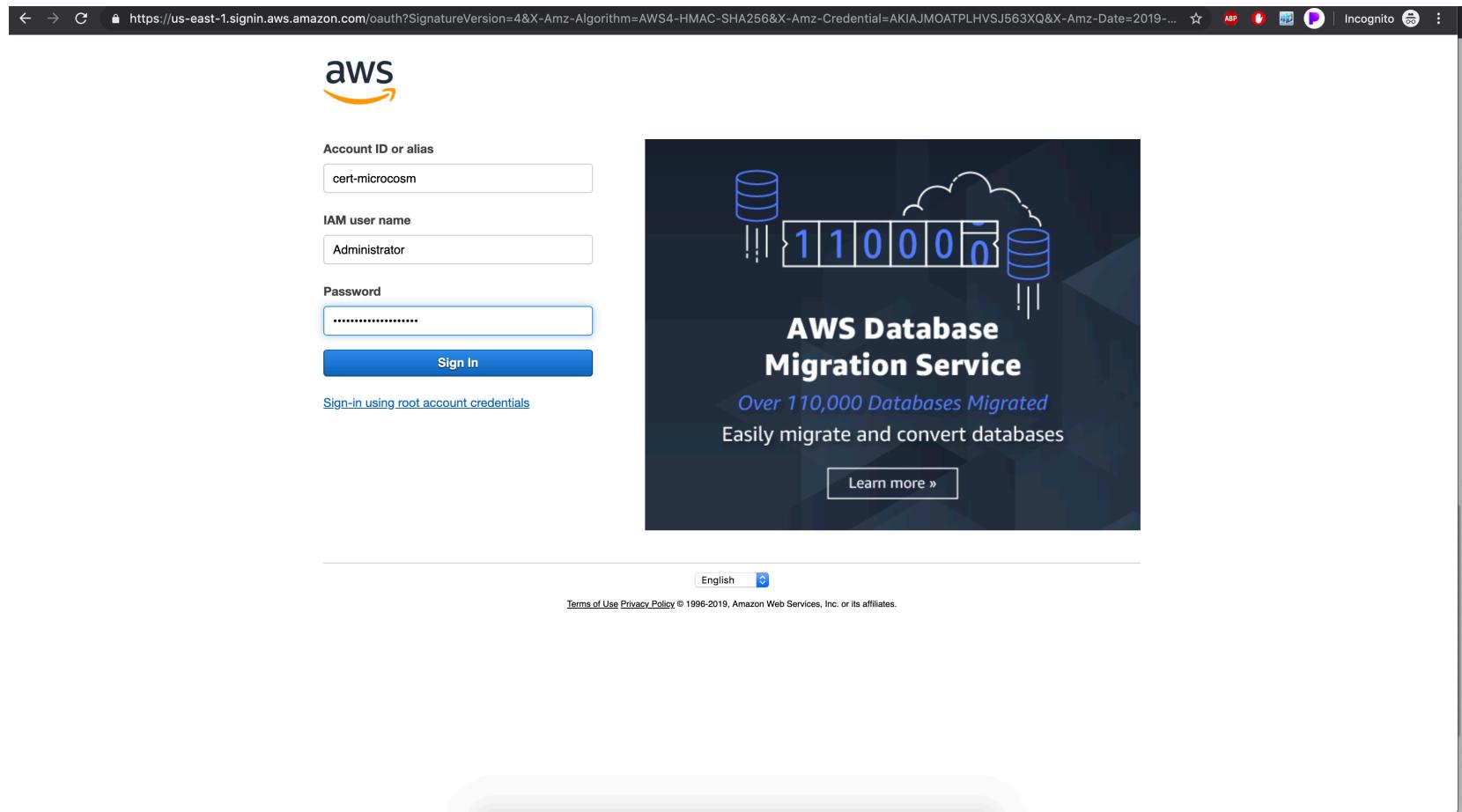
[Return to Table of Contents](#)

Login

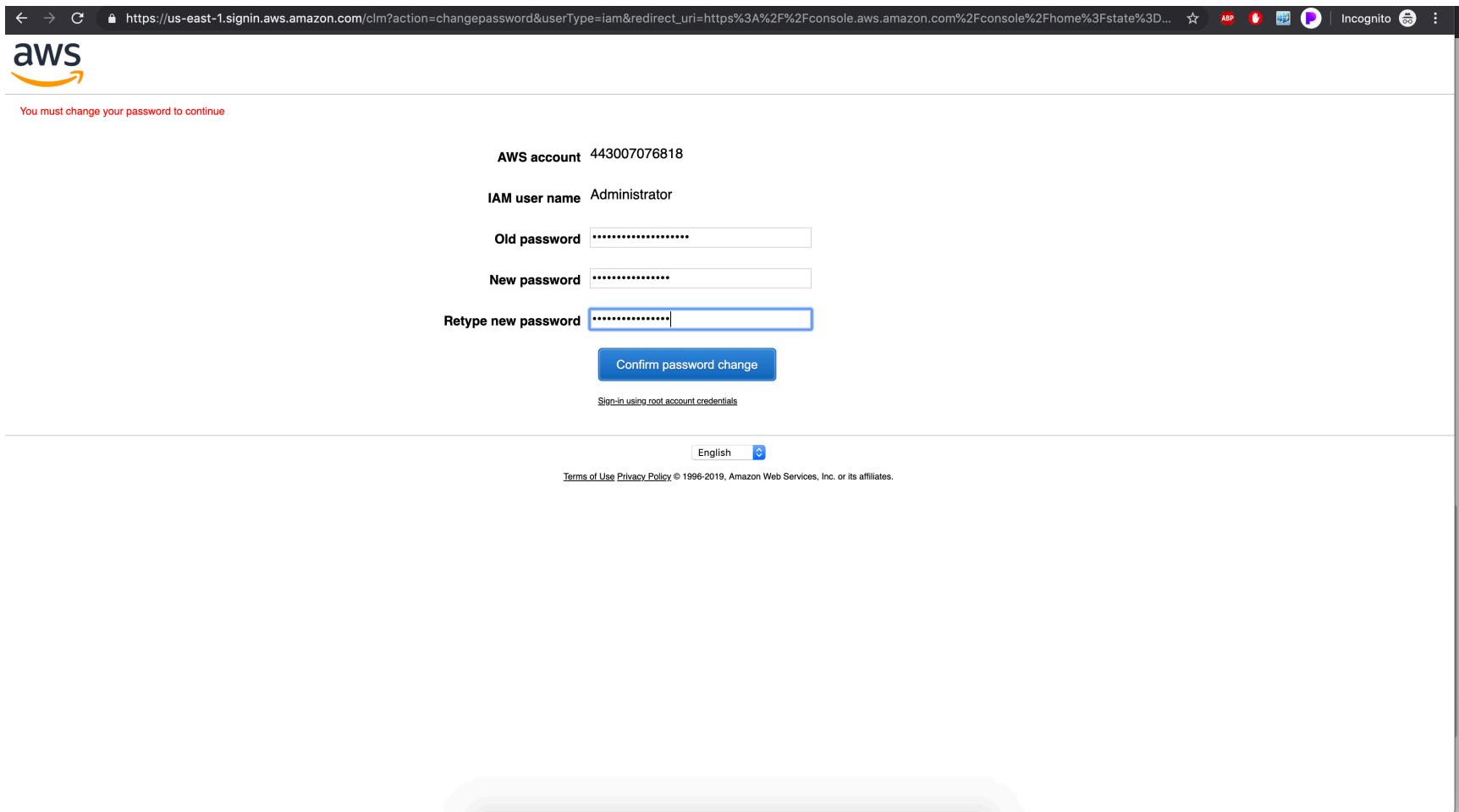
- Navigate to the URL received (eg <https://cert-microcosm.signin.aws.amazon.com/console>)



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



- Change your password to something you will remember



The screenshot shows a password change form on the AWS IAM login page. The URL is https://us-east-1.siginn.aws.amazon.com/clm?action=changepassword&userType=iam&redirect_uri=https%3A%2F%2Fconsole.aws.amazon.com%2Fconsole%2Fhome%3Fstate%3D... . The page displays the AWS logo and account information: AWS account 443007076818 and IAM user name Administrator. It includes fields for Old password, New password, and Retype new password, all of which are masked with dots. A blue rectangular box highlights the Retype new password field. Below these fields is a blue "Confirm password change" button. At the bottom of the form, there is a link "Sign-in using root account credentials" and a language selection dropdown set to English. The footer contains links for Terms of Use and Privacy Policy, along with a copyright notice: © 1996-2019, Amazon Web Services, Inc. or its affiliates.

- 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.

The screenshot shows the AWS Management Console homepage. At the top, there's a navigation bar with the AWS logo, 'Services' dropdown, 'Resource Groups' dropdown, and user information ('Administrator @ cert-microcosm', 'Ohio', 'Support'). Below the header, the main title 'AWS Management Console' is displayed. On the left, there's a sidebar titled 'AWS services' with a 'Find Services' search bar and a 'All services' link. The central area features a 'Build a solution' section with six quick-start options: 'Launch a virtual machine' (With EC2, 2-3 minutes, icon of a CPU), 'Build a web app' (With Elastic Beanstalk, 6 minutes, icon of a cloud with code), 'Build using virtual servers' (With Lightsail, 1-2 minutes, icon of a server with a plus sign), 'Connect an IoT device' (With AWS IoT, 5 minutes, icon of a circular network), 'Start a development project' (With CodeStar, 5 minutes, icon of a developer keyboard), and 'Register a domain' (With Route 53, 3 minutes, icon of a shield with the number 53). To the right, there are two sections: 'Access resources on the go' (with a mobile phone icon) and 'Explore AWS' (with links to 'Amazon Redshift', 'Run Serverless Containers with AWS Fargate', 'Scalable, Durable, Secure Backup & Restore with Amazon S3', and 'AWS Marketplace').

[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.

####Create a Cluster

- 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. On the left, a sidebar menu includes options 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 brief description of what an ECS cluster is. A callout box provides information about the new ARN and resource ID format, with a link to 'Configure ECS ARN setting'. Below this are two buttons: 'Create Cluster' (blue) and 'Get Started' (grey). The main table displays one cluster named 'microcosm' with the following metrics: Services (5), Running tasks (6), Pending tasks (0), and Container instances (0). The table has columns for Cluster name, Services, Running tasks, Pending tasks, and Container instances. At the bottom, there are links for Feedback, English (US), and a copyright notice from 2008-2019.

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

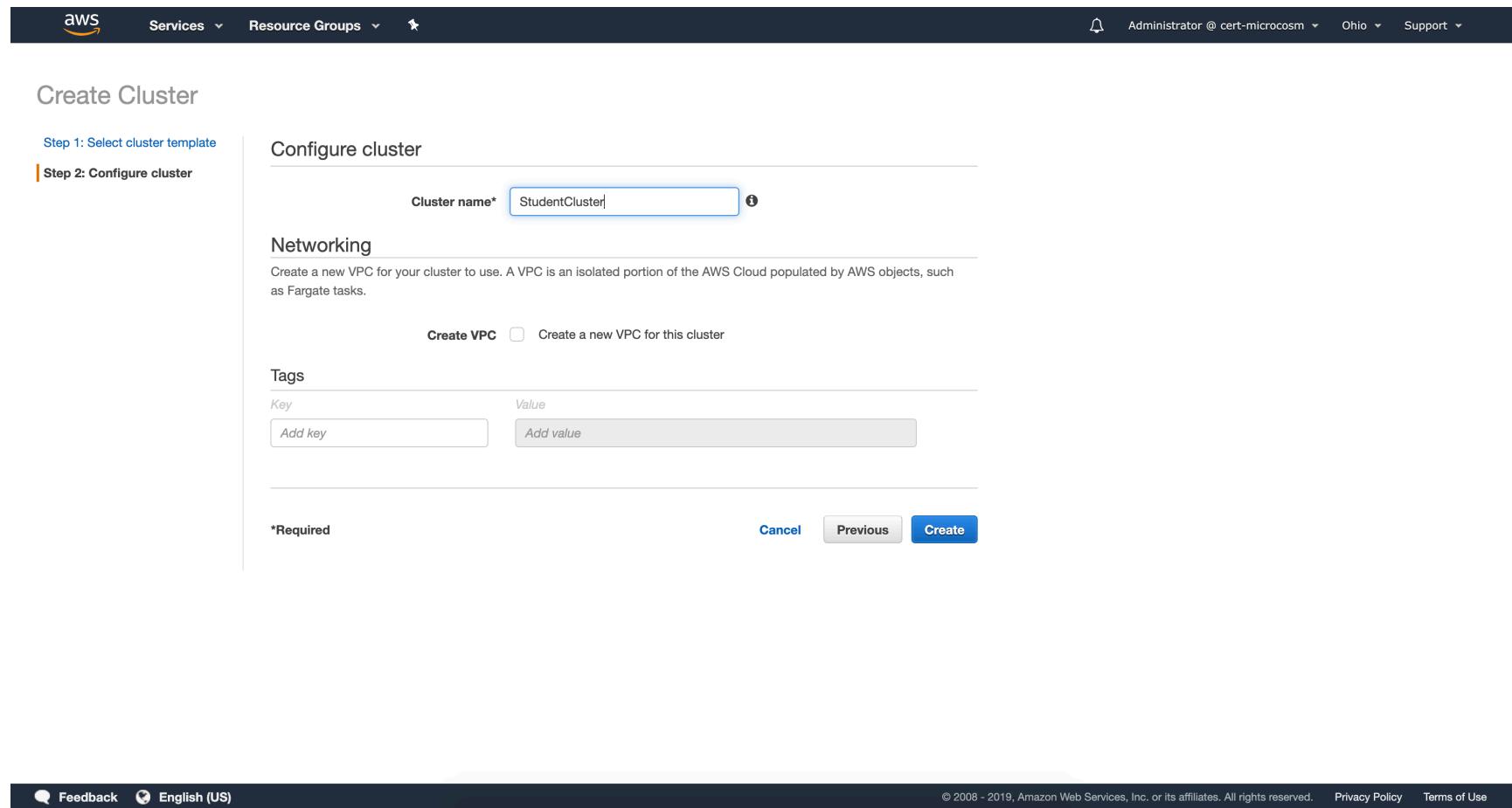
- We are going to be using AWS Fargate, so leave the default selection and click Next Step

The screenshot shows the 'Create Cluster' wizard on the AWS console. The top navigation bar includes the AWS logo, 'Services' dropdown, 'Resource Groups' dropdown, and user information ('Administrator @ cert-microcosm', 'Ohio', 'Support'). The main title is 'Create Cluster'. The first step, 'Step 1: Select cluster template', is highlighted with an orange border. Below it, 'Step 2: Configure cluster' is shown in grey. The 'Select cluster template' section contains three options:

- Networking only**:
Resources to be created:
 - Cluster
 - VPC (optional)
 - Subnets (optional)Powered by AWS Fargate
- EC2 Linux + Networking**:
Resources to be created:
 - Cluster
 - VPC
 - SubnetsAuto Scaling group with Linux AMI
- EC2 Windows + Networking**:
Resources to be created:
 - Cluster
 - VPC
 - SubnetsAuto Scaling group with Windows AMI

At the bottom of the screen, there is a note: '*Required'. On the right side, there are 'Cancel' and 'Next step' buttons.

- Enter a Cluster Name and select Create. We already have a VPC in place, so there is no need to check the Create VPC box. If you do not have a VPC in place yet on your own account, select create VPC.



The screenshot shows the 'Create Cluster' wizard on the AWS Lambda service. The user is on Step 2: Configure cluster. The 'Cluster name*' field is filled with 'StudentCluster'. Under the 'Networking' section, there is a checkbox for 'Create a new VPC for this cluster', which is checked. Below that is a 'Tags' section with two input fields: 'Key' and 'Value', both labeled 'Add key' and 'Add value'. At the bottom, there is a note '*Required' and three buttons: 'Cancel', 'Previous', and a blue 'Create' button.

- Once the cluster is created successfully select View Cluster

The screenshot shows the AWS ECS Launch status page. At the top, there are navigation links for AWS Services, Resource Groups, and a user profile for Administrator @ cert-microcosm. Below the header, the title "Launch status" is displayed, followed by a message: "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." There are two buttons: "Back" and "View Cluster". The main content area shows the message "ECS status - 1 of 1 complete StudentCluster" and a green box containing a checkmark icon and the text "ECS cluster" followed by "ECS Cluster StudentCluster successfully created". At the bottom of the page, there are links for Feedback, English (US), and footer links for Privacy Policy and Terms of Use.

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

The screenshot shows the AWS ECS console interface. On the left, a navigation menu is open under the 'Clusters' section, showing options like Amazon EKS, Clusters, Task Definitions, and others. The main content area is titled 'Cluster : StudentCluster'. It displays cluster status metrics: Status (ACTIVE), Registered container instances (0), Pending tasks count (0 Fargate, 0 EC2), Running tasks count (0 Fargate, 0 EC2), Active service count (0 Fargate, 0 EC2), and Draining service count (0 Fargate, 0 EC2). Below this, there's a table header for managing services, with columns for Service Name, Status, Service type, Task Definition, Desired tasks, Running tasks, Launch type, and Platform version. A note at the bottom of the table says 'No results'. At the bottom of the page, there are links for Feedback, English (US), and various AWS terms and policies.

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

The screenshot shows the AWS ECS Clusters page. On the left, a sidebar lists services: 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 defines an ECS cluster as a regional grouping of one or more container instances. It includes a note about the new ARN and resource ID format, a 'Configure ECS ARN setting' link, and 'Create Cluster' and 'Get Started' buttons. Below this, there are three cluster cards: 'StudentCluster > FARGATE' (Services: 0, Running tasks: 0, Pending tasks: 0) and 'EC2' (Services: 0, Running tasks: 0, Pending tasks: 0, CPUUtilization: No data, MemoryUtilization: No data, Container instances: 0). The third card, 'microcosm > FARGATE', is partially visible.

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	Mount Poi
jenkins	jenkins	h1kkan/jenkins-docker:lts	8080	jenkins_home	/var/jenkins_home
sonarqube	sonarqube	sonarqube:lts	9000	sonarqube_conf	/opt/sonarqube/conf
				sonarqube_data	/opt/sonarqube/data
				sonarqube_extensions	/opt/sonarqube/extensions
				sonarqube_bundled-plugins	/opt/sonarqube/lib/plugins
gitlab	gitlab	gitlab/gitlab-ce	443	gitlab-config	/etc/gitlab
			80	gitlab-logs	/var/log/gitlab
				gitlab-data	/var/opt/gitlab
owaspZAP	owaspzap	owasp/zap2docker-stable	8080		
			8090		
nexus	sonatype_nexus	sonatype/nexus	8081	nexus-data	/sonatype-work

[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. The left sidebar has 'Task Definitions' selected. The main area displays a table of task definitions with columns for 'Task Definition' and 'Latest revision status'. All entries are marked as 'ACTIVE'. The table includes rows for 'first-run-task-definition', 'gitlab', 'hubot-slack', 'jenkins', 'sonarqube', and 'zap2docker'. The page also features standard AWS navigation elements like a search bar, a top navigation bar with links for Services, Resource Groups, and Support, and a footer with links for Feedback, English (US), Privacy Policy, and Terms of Use.

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

- Select the Fargate Type and press Next Step

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

Create new Task Definition

Step 1: Select launch type compatibility

Step 2: Configure task and container definitions

Select launch type compatibility

Select which launch type you want your task definition to be compatible with based on where you want to launch your task.

FARGATE



Price based on task size
Requires network mode awsvpc
AWS-managed infrastructure, no Amazon EC2 instances to manage

EC2



Price based on resource usage
Multiple network modes available
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

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

AWS Services Resource Groups mappu cosm Ohio Support

Create new Task Definition

Step 1: Select launch type compatibility

Step 2: Configure task and container definitions

Configure task and container definitions

A task definition specifies which containers are included in your task and how they interact with each other. You can also specify data volumes for your containers to use. [Learn more](#)

Task Definition Name* nexus

Requires Compatibilities* FARGATE

Task Role ecsTaskExecutionRole [Edit](#)

Optional IAM role that tasks can use to make API requests to authorized AWS services. Create an Amazon Elastic Container Service Task Role in the [IAM Console](#)

Network Mode awsvpc

If you choose <default>, ECS will start your container using Docker's default networking mode, which is Bridge on Linux and NAT on Windows. <default> is the only supported mode on Windows.

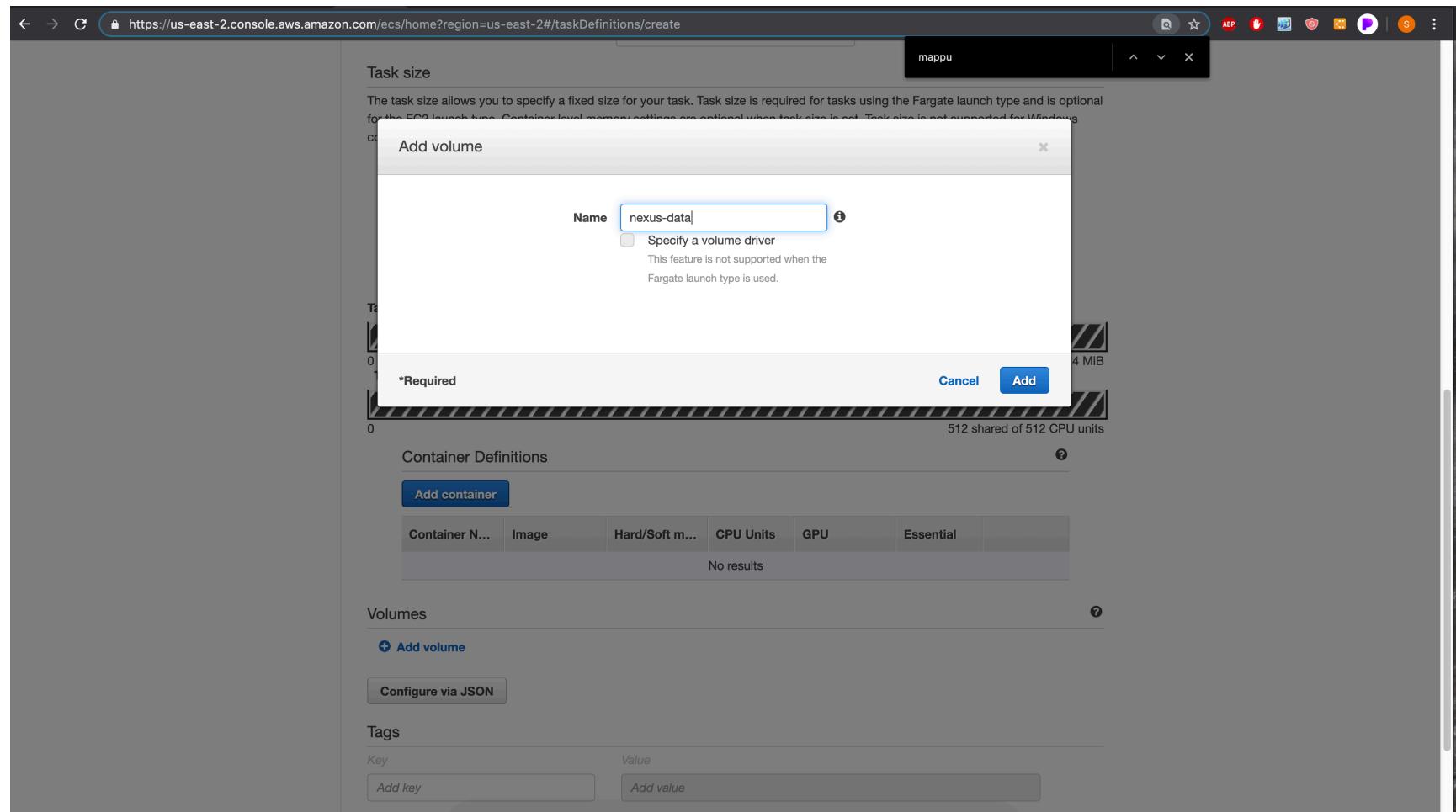
Task execution IAM role

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.

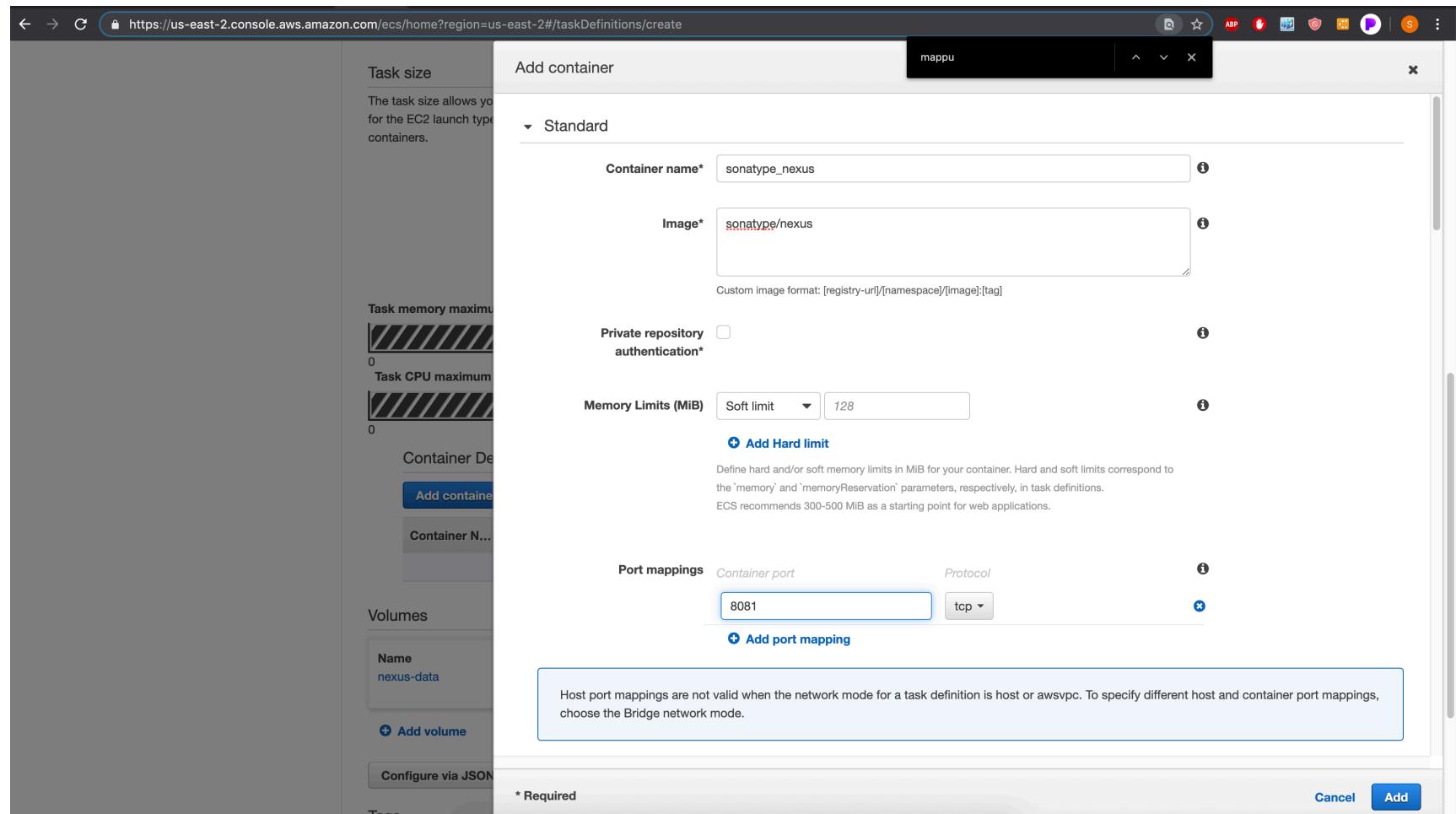
Task execution role ecsTaskExecutionRole [Edit](#)

Task size

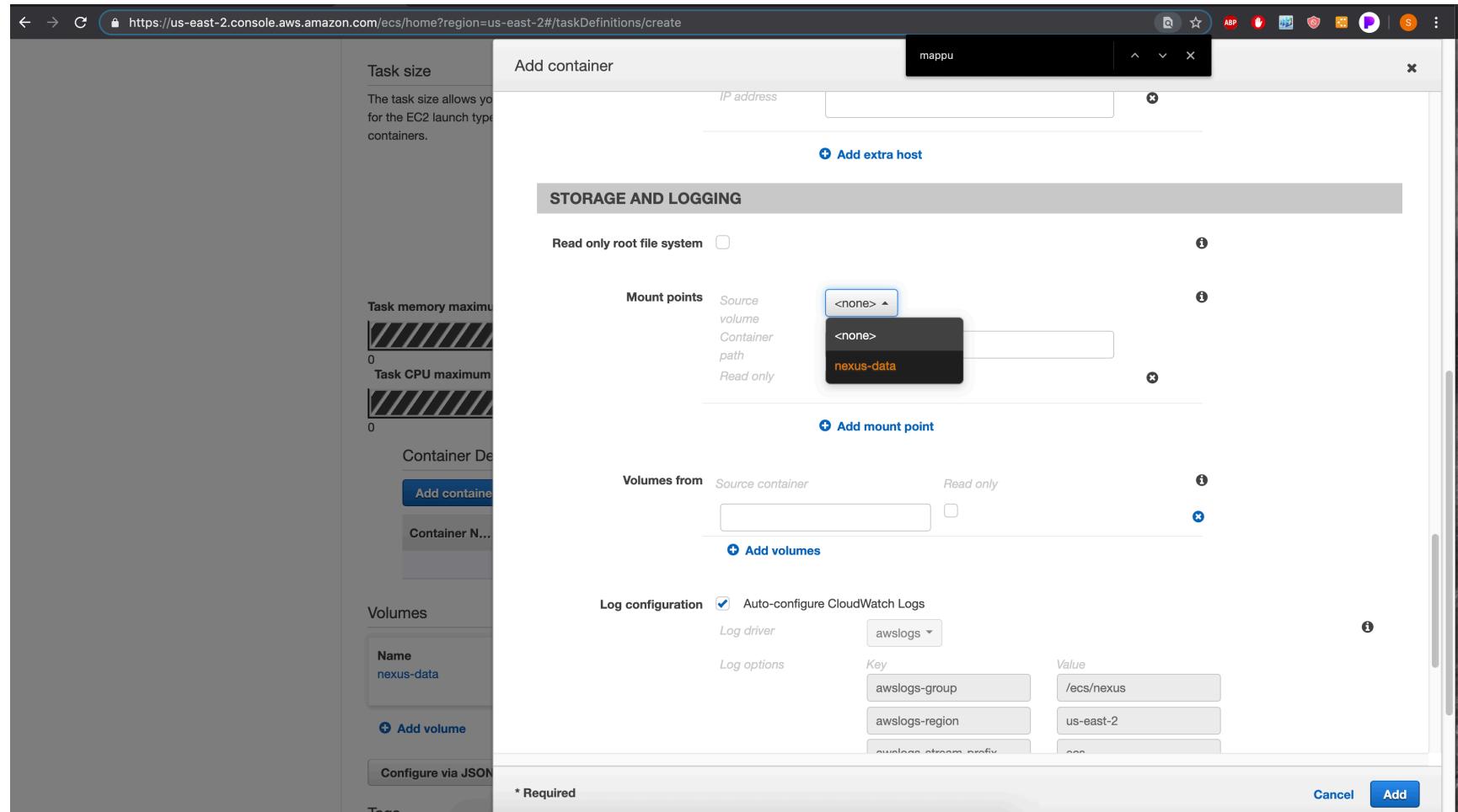
- Select Add Volume, enter the Volume Name and and press add



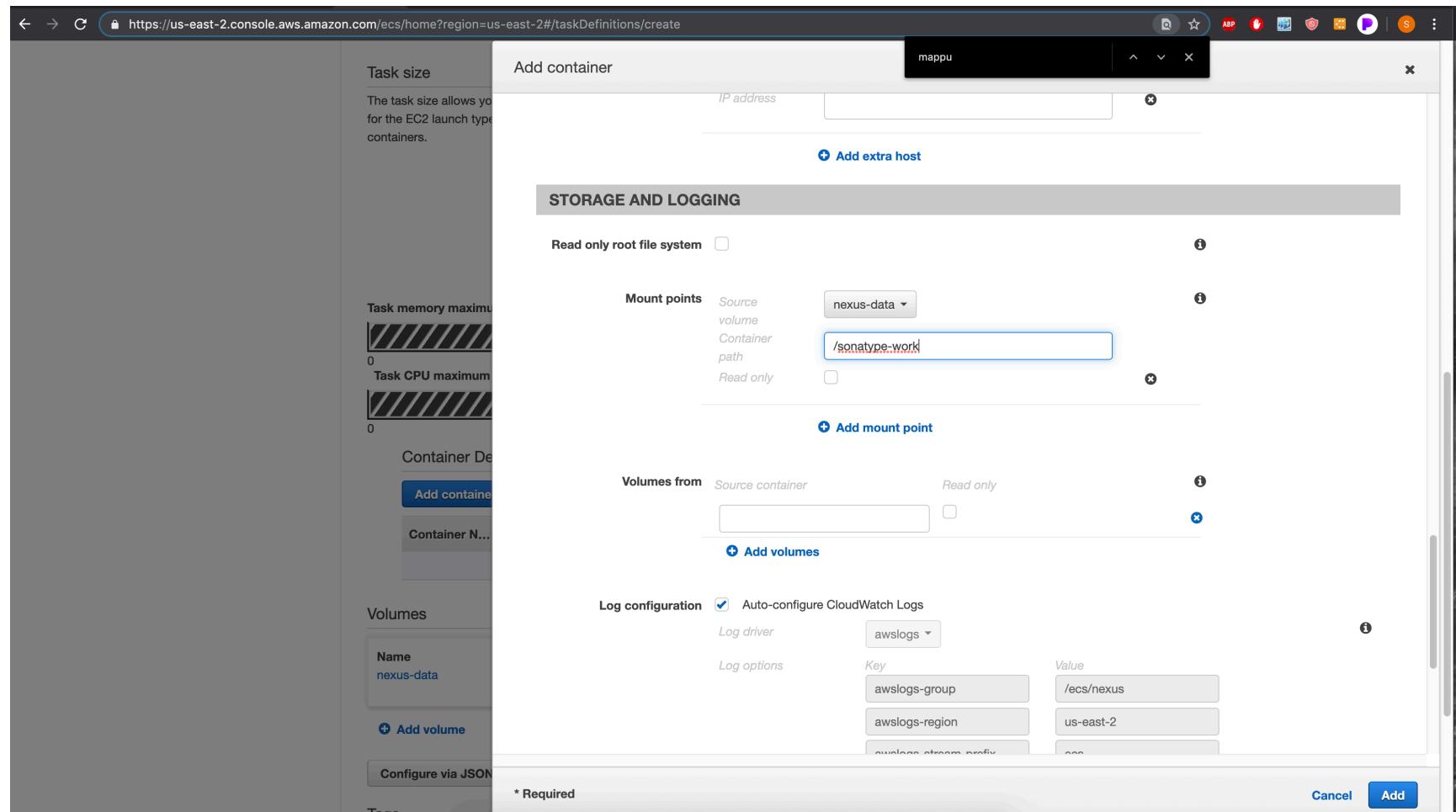
- Select Add Container.
- Enter the Container Name, Image, Port mappings number(s) (select Add port mapping to add additional ports)



- 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

The screenshot shows the AWS CloudWatch Metrics console. At the top, there is a search bar with the query "mappu". Below the search bar, a message states "The valid CPU range for 1GB memory is: 0.25 vCPU - 0.5 vCPU". The main area displays a single data series for the metric "mappu". The chart has two horizontal axes: the top axis represents CPU units from 0 to 1024 shared of 1024 MiB, and the bottom axis represents CPU units from 0 to 512 shared of 512 CPU units. The data series consists of diagonal hatching across both axes, indicating a constant value across the entire range.

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

- Review the task definition, if desired

The screenshot shows the AWS ECS Task Definitions page. The task definition name is 'nexus:1'. Key settings include:

- Task Definition Name:** nexus
- Task Role:** ecsTaskExecutionRole
- Network Mode:** awsvpc (selected)
- Compatibilities:** EC2, FARGATE
- Requires compatibilities:** FARGATE
- Task execution IAM role:** (Task execution role: ecsTaskExecutionRole)

[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-stand 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

The screenshot shows the AWS ECS Task Definitions page for a task named 'nexus:1'. The 'Actions' dropdown menu is open, displaying options: Run Task, Create Service, Update Service, and Deregister. Other visible settings include the 'Definition Name' (nexus), 'Task Role' (ecsTaskExecutionRole), 'Network Mode' (awsvpc), and compatibility (EC2, FARGATE). A note about network mode states: 'If you choose <default>, ECS will start your container using Docker's default networking mode, which is Bridge on Linux and NAT on Windows. <default> is the only supported mode on Windows.' Below the task definition details, there is a section for the 'Task execution IAM role', which is currently set to 'ecsTaskExecutionRole'. The 'Task size' section is also partially visible.

- 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 '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' is 'jenkinsdefault'. The 'Service name' is 'sonatype_nexus'. The 'Service type*' is 'REPLICA'. The 'Number of tasks' is '1'. The 'Minimum healthy percent' is '100'. The 'Maximum percent' is '200'. At the bottom, there's a 'Deployments' section with the placeholder text 'Choose a deployment option for the service.'

- 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

The screenshot shows the 'Create Service' wizard on the AWS ECS console. The current step is 'Step 2: Configure network'. The 'Configure network' section is active, indicated by a blue border. The 'VPC and security groups' tab is selected. Under 'Cluster VPC*', the dropdown shows 'vpc-0a8f63ffd28e39ced [10.0.0.0/...]' with a help icon. The 'Subnets*' dropdown is open, showing two options: 'subnet-0c12a48c3055e9708 (10.0.0.0/24) | ECS Jenkinsdefault - Public Subnet 1 - us-east-2a assign ipv6 on creation: Disabled' and 'subnet-0a93e2b8145176367 (10.0.1.0/24) | ECS Jenkinsdefault - Public Subnet 2 - us-east-2b assign ipv6 on creation: Disabled'. The first option is highlighted. The 'Security groups*' dropdown is also open, showing a single item: 'ECS Jenkinsdefault'. The 'Auto-assign public IP' field is set to 'None'. Below these fields is a 'Health check grace period' section with a note about ignoring ELB health checks for up to 2,147,483,647 seconds. At the bottom of the configuration section is a note that 'Health check grace period requires a load balancer.' The 'Load balancing' section follows, with a note about using an existing load balancer or creating a new one in the Amazon EC2 console. The 'Load balancer' radio button is set to '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	Copy to new
<input type="checkbox"/>	sg-050f7aa077e6edfa2	gitlab-2784	2019-04-16T15:33:53.395Z	Copy to new
<input type="checkbox"/>	sg-0793fec6a9c4a1c01	default	default VPC security group	Copy to new

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

The screenshot shows the AWS CloudWatch Metrics console with the URL <https://us-east-2.console.aws.amazon.com/ecs/home?region=us-east-2#/taskDefinitions/nexus/1/createService>. The top navigation bar includes the AWS logo, Services, Resource Groups, a bell icon for notifications, and account information for Administrator @ cert-microcosm, Ohio, and Support.

The main content area is titled "Create Service" and displays "Step 3: Set Auto Scaling (optional)". A sub-section titled "Set Auto Scaling (optional)" explains that it allows automatically adjusting the service's desired count based on CloudWatch alarms. It includes two radio button options:

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

At the bottom of the form, there is a note: "*Required". On the right side, there are three buttons: "Cancel", "Previous", and a blue "Next step" button.

The footer of the page includes links for Feedback, English (US), and legal notices: © 2008 - 2019, Amazon Web Services, Inc. or its affiliates. All rights reserved., Privacy Policy, and Terms of Use.

- Review and press Create Service

The screenshot shows the AWS ECS Create Service wizard with three configuration steps:

- Configure network:**
 - Minimum healthy percent: 100
 - Maximum percent: 200
- Configure service discovery:**
 - Namespace: ns-iz26fjly45cjajqgg
 - Service discovery name: sonatype_nexus
 - Enable ECS task health propagation: true
 - DNS record type and TTL: A 60
- Set Auto Scaling (optional):**
 - not configured

At the bottom, there are navigation buttons: Cancel, Previous, and Create Service (highlighted in blue).

- Once the service is successfully created, select View Service

The screenshot shows the AWS CloudWatch Metrics interface with a search bar at the top. Below the search bar, there are two main sections: 'Metrics' and 'Logs'. Under 'Metrics', there is a table with columns 'Metric Name', 'Series', 'Dimensions', and 'Last Value'. One row in the table is highlighted in red. At the bottom of the page, there is a section titled 'Additional integrations you can connect to your ECS service'.

ECS Service status - 4 of 4 completed

Configure Task Networking

Create security group

✓ Create security group
sonaty-6743 succeeded sg-08677ef9f87f9e049

Set inbound rules

✓ Set inbound rules
succeeded sg-08677ef9f87f9e049

Configure Service Discovery

Create service discovery service

✓ Create service discovery service
arn:aws:servicediscovery:us-east-2:443007076818:service/srv-bcfmvbuvwuvxelnz created

Create Service

Create service: sonatype_nexus

✓ Service created
Service created. Tasks will start momentarily. View: [sonatype_nexus](#)

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

The screenshot shows the AWS ECS Task Details page. The task ID is 325b875c-9e8f-4686-872a-937c986973d7, running in the Jenkins default cluster. The task definition is nexus:1, and it is using the FARGATE launch type. The platform version is 1.3.0. The task role is ecsTaskExecutionRole. The last status was ACTIVATING, and the desired status is RUNNING. The task was created at 2019-04-12 14:25:35 -0600. The network mode is awsvpc, and the container has an ENI ID of eni-070590cb8cb942258, a private IP of 10.0.0.86, and a public IP of 13.58.236.137. The container's MAC address is 02:38:3b:e6:c3:da. The container is named sonatype_nexus and is currently running.

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.

The screenshot shows the AWS ECS Clusters page. On the left, a sidebar lists services: 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". It defines an ECS cluster as a regional grouping of one or more container instances. A note mentions the introduction of a new ARN and resource ID format. Below this, there are "Create Cluster" and "Get Started" buttons. The page displays two clusters: "jenkinsdefault > FARGATE" and "microcosm > FARGATE". The "jenkinsdefault" cluster details are as follows:

Category	Value
Services	6
Running tasks	5
Pending tasks	1
EC2	
Services	0
Running tasks	0
Pending tasks	0
CPUUtilization	No data
MemoryUtilization	No data
Container instances	0

The "microcosm" cluster details are as follows:

Category	Value
Services	0
Running tasks	0
Pending tasks	0
CPUUtilization	No data
MemoryUtilization	No data
Container instances	0

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

The screenshot shows the AWS ECS Cluster details page for 'jenkinsdefault'. The cluster status is ACTIVE. It lists 0 registered container instances, 0 pending tasks, 6 running tasks, 6 active services, and 0 draining services. The tasks are listed in a table with columns: Task, Task definition, Container instan..., Last status, Desired status, Started By, Group, Launch type, and Platform version. All tasks are currently RUNNING.

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-9e8f-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](#)

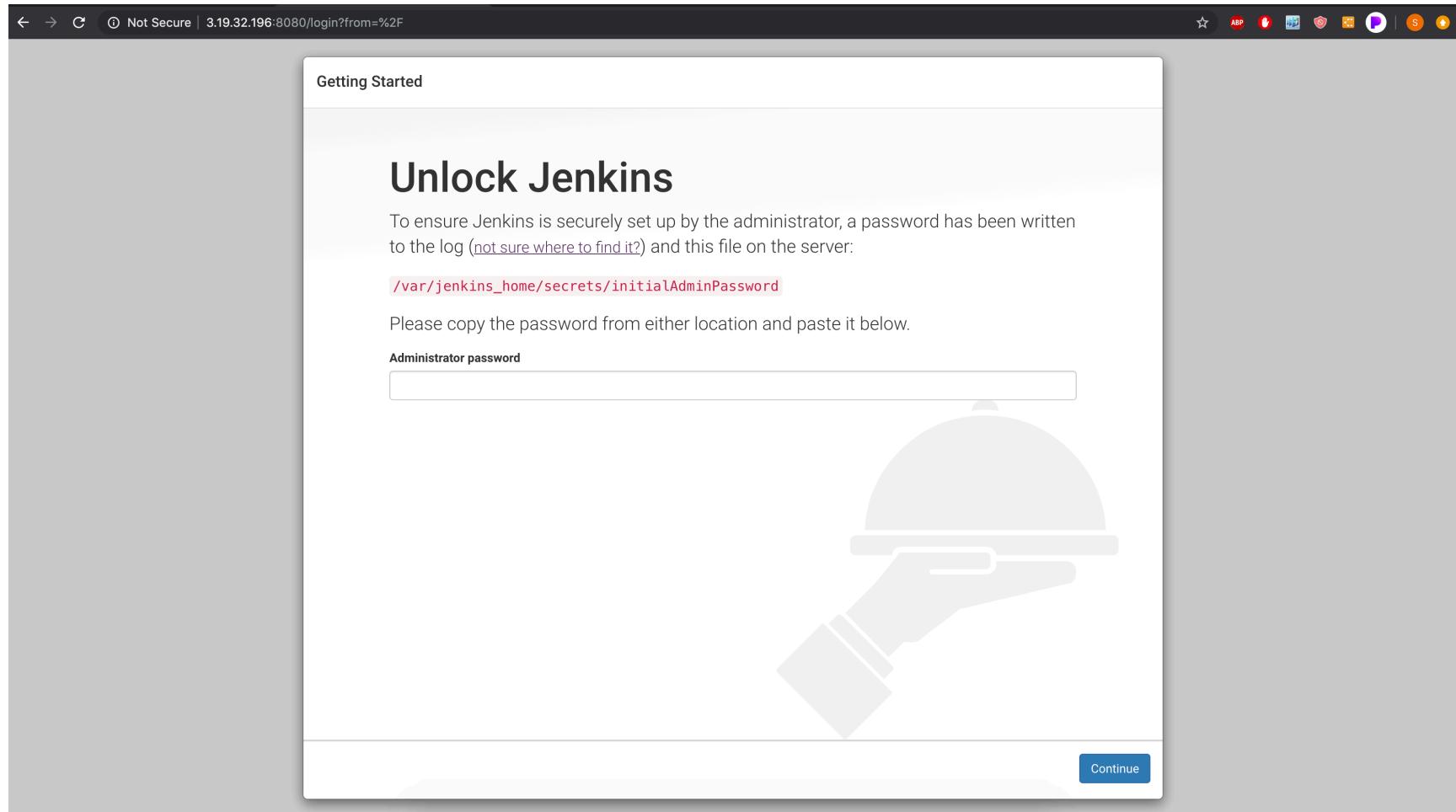
Setting up Jenkins for DevOps Pipeline (Microcosm)

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

Log Stream	Last Event Time
ecs/jenkins/92c59a81-6a6a-45e0-bc7b-3932928fdfa9	2019-04-16 13:27 UTC-6
ecs/jenkins/dd06962c-acb9-49e5-b514-70c1d84adf06	2019-04-16 13:24 UTC-6
ecs/jenkins/234135b5-3ad6-40a5-a8bc-2c96a5977dd3	2019-04-16 13:18 UTC-6
ecs/jenkins/cb06a3e4-8354-4392-9abc-b577bd0ecaef	2019-04-16 13:12 UTC-6
ecs/jenkins/4c39ca9d-512e-43e9-af0e-49f6d1b46d87	2019-04-16 13:06 UTC-6
ecs/jenkins/23da544d-46a1-4c1f-bcfe-59439fd398b3	2019-04-16 13:01 UTC-6
ecs/jenkins/1baa1470-602c-4503-8ec6-d2b7c934d8ce	2019-04-16 12:55 UTC-6
ecs/jenkins/f87338b8-e9fe-462b-99ec-00c1ca93b043	2019-04-16 12:49 UTC-6
ecs/jenkins/ef11b1d9-3bcf-4c4e-9efe-1f78adb17ed0	2019-04-16 12:43 UTC-6
ecs/jenkins/a4b79c1e-377a-4bd8-b3a4-79bece8e8ae4	2019-04-16 12:38 UTC-6
ecs/jenkins/40e70ae0-edc1-4778-8a5d-e77ef096b7f5	2019-04-16 12:36 UTC-6
ecs/jenkins/30d0d420-f7ca-4a41-b81f-5ab001dde223	2019-04-16 12:35 UTC-6
ecs/jenkins/7eea7fe8-2a5d-4170-a077-66a3ba95635d	2019-04-16 12:34 UTC-6
ecs/jenkins/1927ea4d-fbb8-4dbc-b942-870140cc4ced	2019-04-16 12:32 UTC-6
ecs/jenkins/12c54ba2-90d8-416d-ab59-c718a7624384	2019-04-16 12:31 UTC-6
ecs/jenkins/a6329022-7683-488a-af9f-4ac0a1dc0df3	2019-04-16 12:30 UTC-6
ecs/jenkins/02182562-c89e-4287-a4df-ae69b0f53a2c	2019-04-16 12:29 UTC-6
ecs/jenkins/eff68630-8228-403f-b4dd-c44916ba2d35	2019-04-16 12:27 UTC-6
ecs/jenkins/81ed7740-ea0e-40a7-9954-55bf447afec3	2019-04-16 12:26 UTC-6
ecs/jenkins/30da76c7-0dcb-4448-9e93-9ee15e385485	2019-04-16 12:25 UTC-6
ecs/jenkins/1b42226d-fba0-4bf4-824b-a13b16e4d68d	2019-04-16 12:10 UTC-6
ecs/jenkins/7071fa1c-47f5-43ba-a40c-8a5a5a61a720	2019-04-16 11:53 UTC-6
ecs/jenkins/c272e68f-c302-4bb1-8610-cee6c63c6bc3	2019-04-16 10:36 UTC-6
ecs/jenkins/0be7fe7b-eb7d-43c3-9076-9bad78e68e9f	2019-04-16 15:59 UTC-6

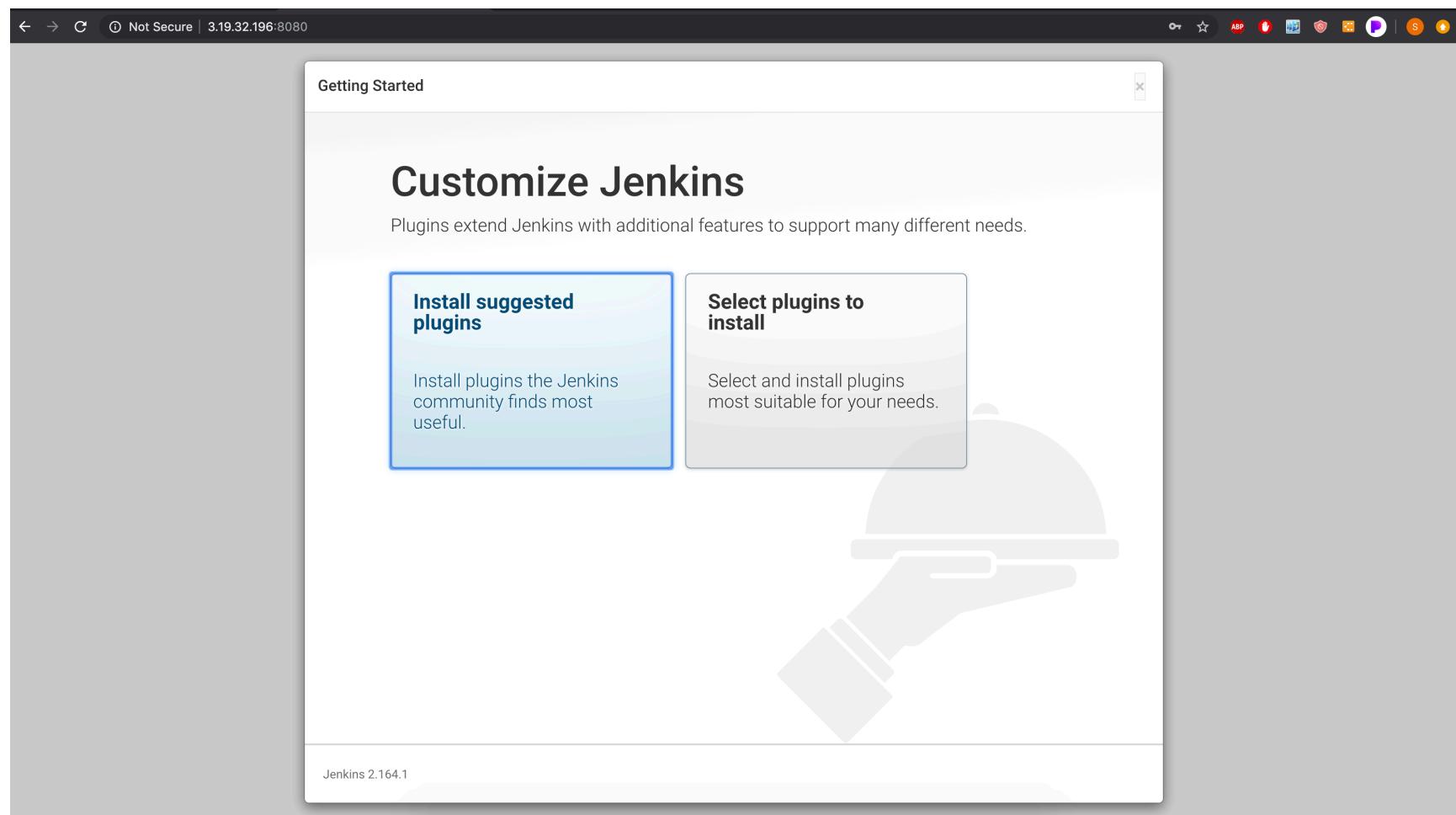
<https://us-east-2.console.aws.amazon.com/cloudwatch/home?region=us-east-2#logEventViewer;group=/ecs/jenkins;stream=ecs/jenkins/92c59a81-6a6a-45e0-bc7b-3932928fdfa9;start=2019-04-15T19:27:55Z>

affiliates. All rights reserved. [Privacy Policy](#) [Terms of Use](#)

- 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 Log Groups interface. The left sidebar is collapsed, showing options like CloudWatch, Dashboards, Alarms, ALARM (0), INSUFFICIENT (0), OK (0), Billing, Events, Rules, Event Buses, Logs (selected), Insights, Metrics, and Favorites. The main area displays log events for the /ecs/jenkins stream. The log entries are timestamped at 2019-04-16 19:28:31 UTC. The first few lines show the initial setup of the Jenkins application context, including the refresh of the static web application context and the pre-instantiation of singletons. Subsequent lines show the Jenkins install process, including the generation of an initial admin password and the update of the update center data file. The interface includes a filter bar at the top, a toolbar with various icons, and a footer with links for Feedback, English (US), Copyright notice (© 2008 - 2019, Amazon Web Services, Inc. or its affiliates. All rights reserved.), Privacy Policy, and Terms of Use.

- Click Install suggested plugins



- Wait while the plugins are installed

The screenshot shows the Jenkins 'Getting Started' page. At the top, there's a navigation bar with icons for back, forward, and search, followed by 'Not Secure | 3.19.32.196:8080'. Below the header is a large 'Getting Started' title with a progress bar underneath. The main content area contains a table of Jenkins plugins. The table has two sections: a grid of installed plugins and a list of available plugins. The installed section includes Folders Plugin, OWASP Markup Formatter Plugin, Build Timeout, Credentials Binding Plugin, Timestampper, Workspace Cleanup Plugin, Ant Plugin, Gradle, Pipeline, GitHub Branch Source Plugin, Pipeline: GitHub Groovy Libraries, Pipeline: Stage View Plugin, Git plugin, Subversion, SSH Slaves plugin, Matrix Authorization Strategy Plugin, and PAM Authentication plugin. The available section lists LDAP Plugin, Email Extension Plugin, and Mailer Plugin. A sidebar on the right lists all the installed plugins. At the bottom of the page, it says 'Jenkins 2.164.1'.

✓ Folders Plugin	✓ OWASP Markup Formatter Plugin	✓ Build Timeout	✓ Credentials Binding Plugin
✓ Timestampper	✓ Workspace Cleanup Plugin	✓ Ant Plugin	✓ Gradle
✓ Pipeline	✓ GitHub Branch Source Plugin	✓ Pipeline: GitHub Groovy Libraries	✓ Pipeline: Stage View Plugin
✓ Git plugin	✗ Subversion	✗ SSH Slaves plugin	✗ Matrix Authorization Strategy Plugin
✗ PAM Authentication plugin	✗ LDAP Plugin	✗ Email Extension Plugin	✗ Mailer Plugin

Installed Plugins:

- Folders
- JDK Tool
- OWASP Markup Formatter
- Build Timeout
- Credentials Binding
- Timestampper
- Workspace Cleanup
- Ant
- Gradle
- Pipeline
- Github Branch Source
- Pipeline: GitHub Groovy Libraries
- Pipeline: Stage View
- Git
- MapDB API

Available Plugins:

- LDAP
- Email Extension
- Mailer

Jenkins 2.164.1

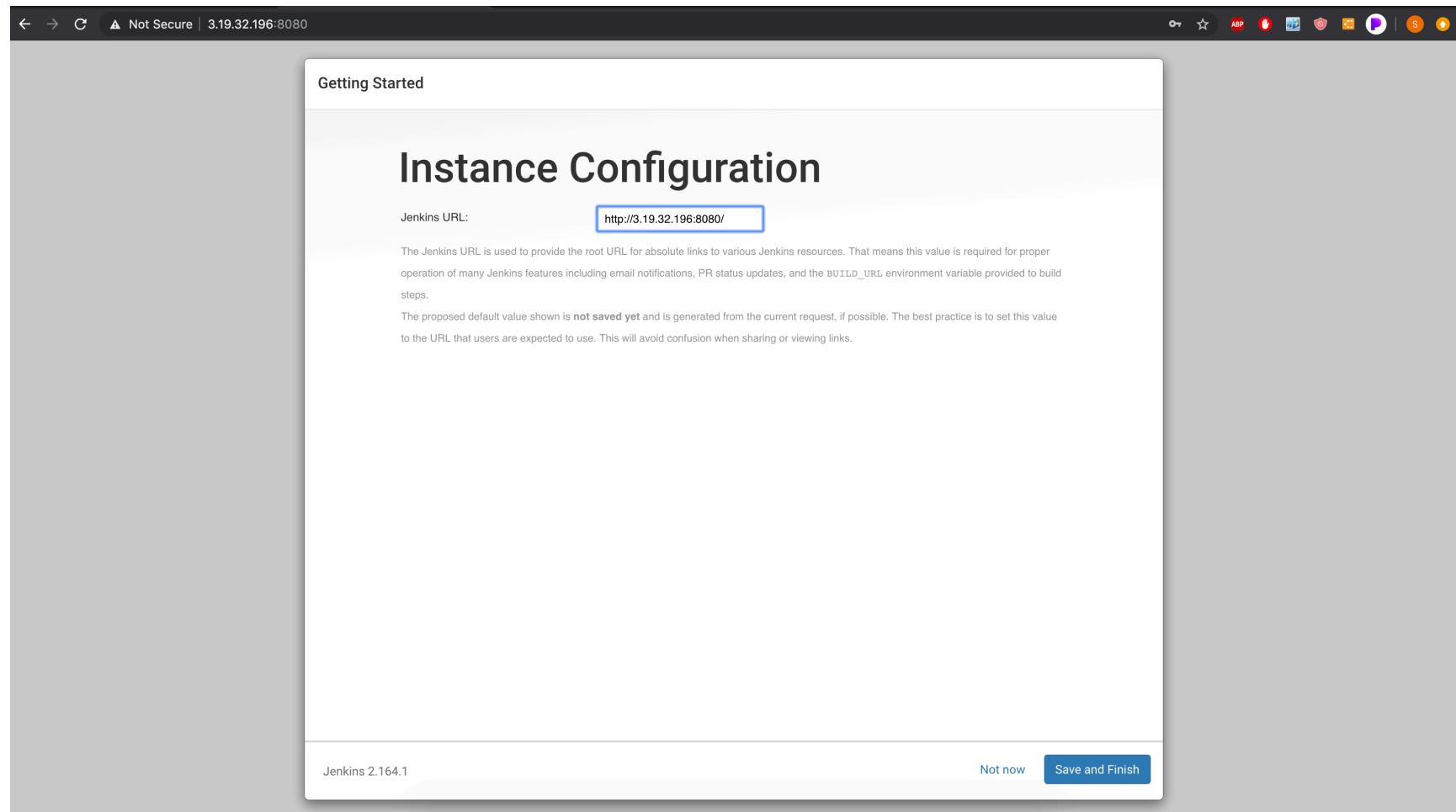
- Fill in the information for the admin user. Make sure to remember your password.
- Select Save and Continue

The screenshot shows a web browser window with the address bar displaying 'Not Secure | 3.19.32.196:8080'. The main content is titled 'Getting Started' and features a large heading 'Create First Admin User'. Below the heading are five input fields for user information:

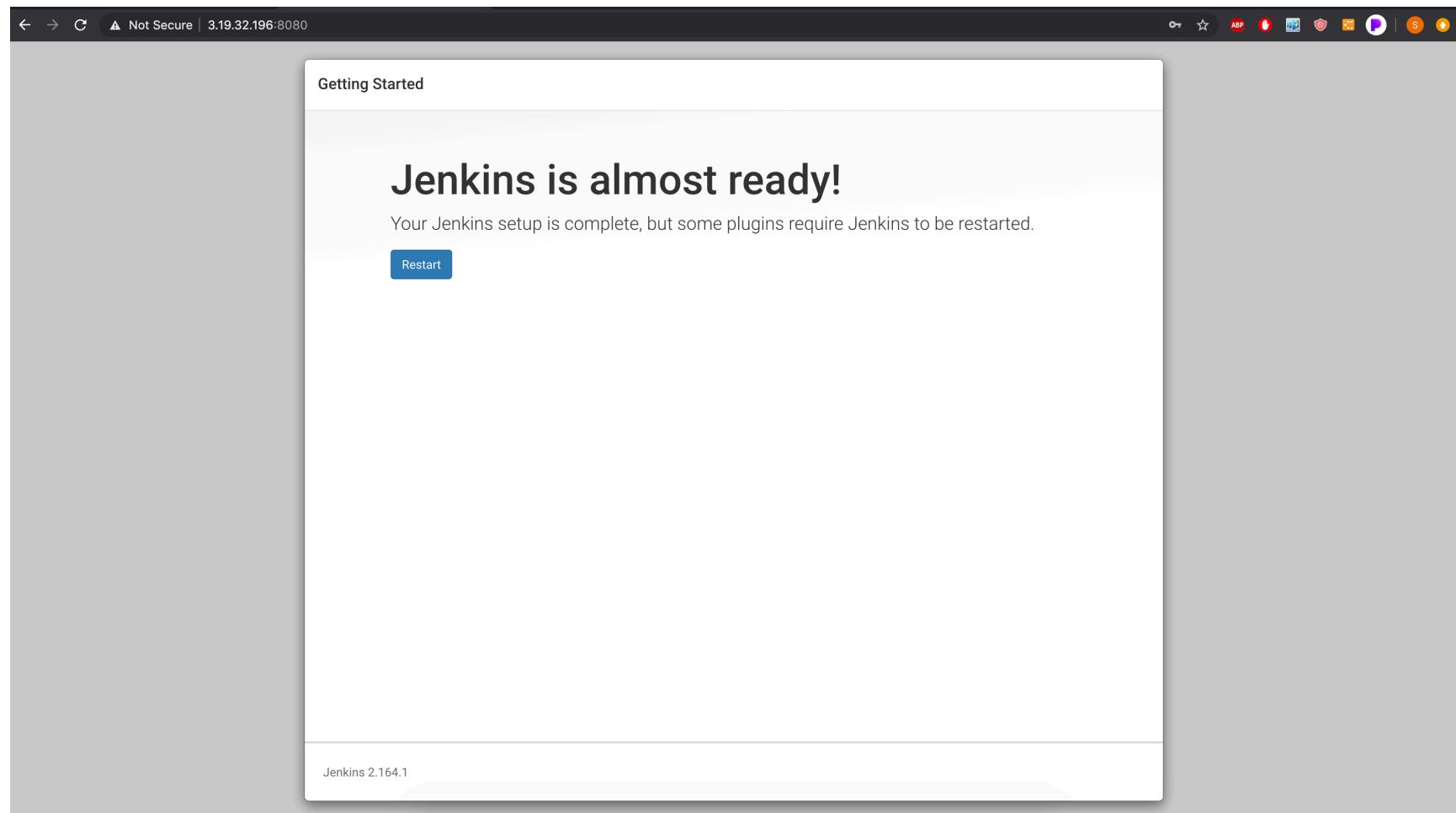
Username:	admin
Password:
Confirm password:
Full name:	admin
E-mail address:	admin@example.com

At the bottom of the page, there are two buttons: 'Continue as admin' and a blue button labeled 'Save and Continue'.

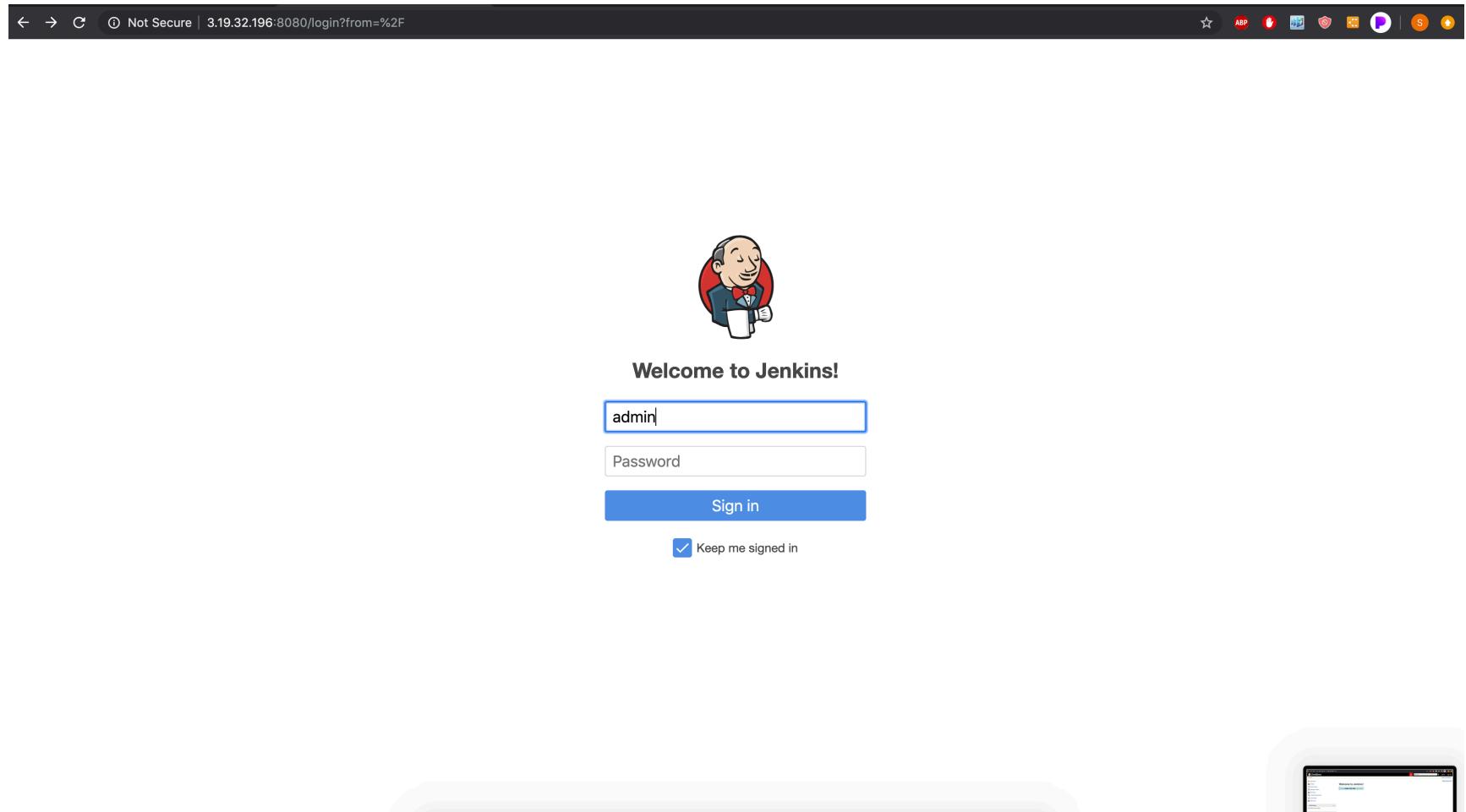
- Select Save and Finish



- 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



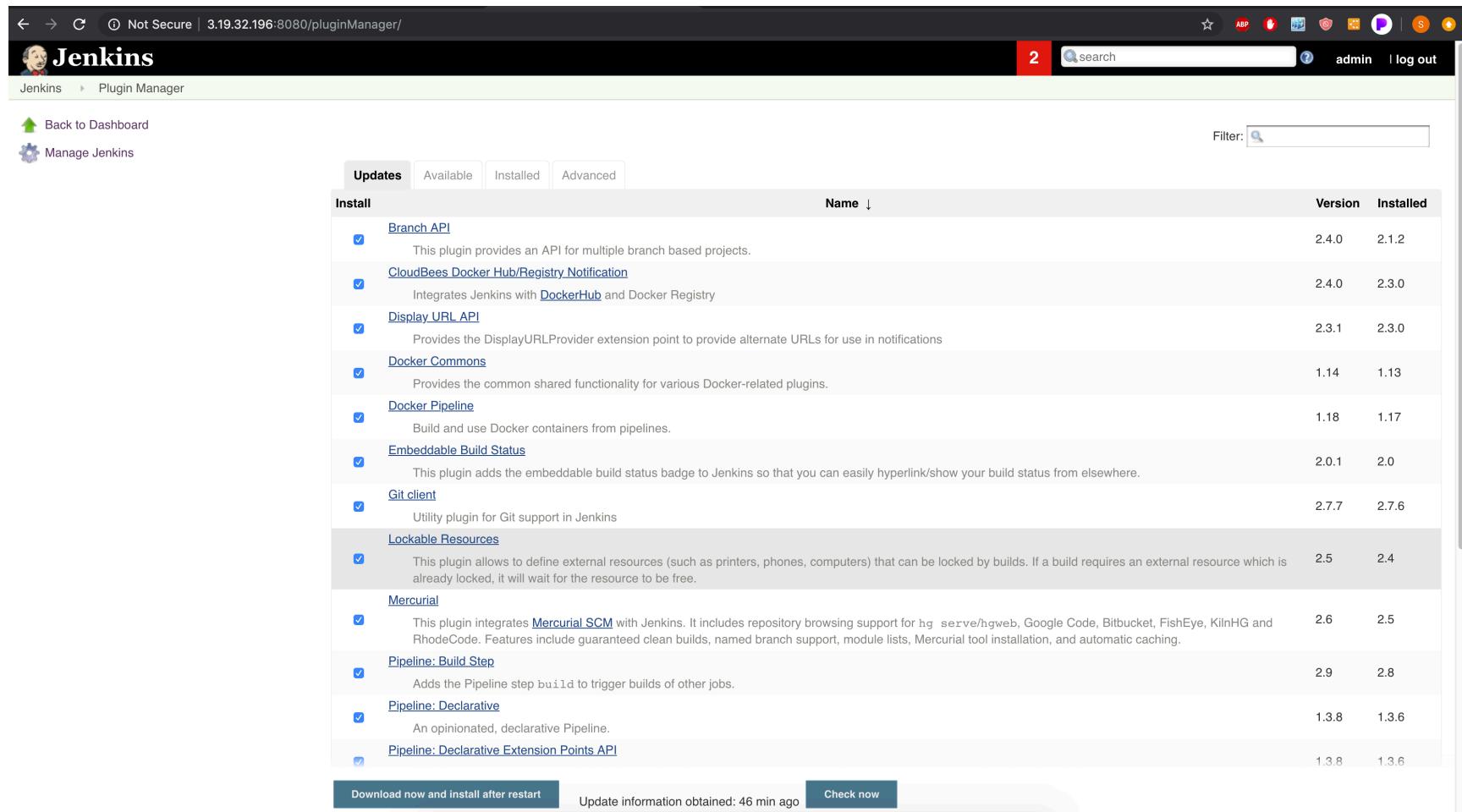
- From the main page, Select the warning in the corner and select Go to plugin manager
 - If no warnings are present, Select Manage Jenkins on the Left, then scroll down and select Manage Plugins

The screenshot shows the Jenkins dashboard at <http://3.19.32.196:8080>. The top right corner indicates there are 2 notifications. The dashboard features a sidebar with links like 'New Item', 'People', 'Build History', etc., and sections for 'Build Queue' (empty) and 'Build Executor Status' (2 Idle). A central message encourages creating new jobs. A prominent alert box at the top right informs about a new Jenkins version (2.164.2) available for download ([changelog](#)). It also lists several security vulnerabilities:

- 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](#)

A 'Manage Jenkins' link is located at the bottom right of the alert box. The footer of the page includes the generation time 'Page generated: Apr 16, 2019 7:42:06 PM UTC' and links to 'REST API' and 'Jenkins ver. 2.164.1'.

- 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's a navigation bar with links for 'Back to Dashboard' and 'Manage Jenkins'. Below that is a search bar and user information ('admin | log out'). The main area has tabs for 'Updates', 'Available' (which is selected), 'Installed', and 'Advanced'. A 'Filter:' input field is also present. The table lists available plugins under the 'Install' section, with columns for Name, Version, and Installed status. 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 'Download now and install after restart' button at the bottom.

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

Plugin Name	Description	Version
SLOCCount	For a given Parameterized Project, this plugin shows the builds sorted by the parameters used to execute the builds.	1.5
SonarQube Scanner	This plugin parses SLOCCount output files to produce project and build reports.	1.24
Sonargraph Integration	This plugin allows an easy integration of SonarQube , the open source platform for Continuous Inspection of code quality.	2.8.1
Sonargraph	This plugin integrates Sonargraph functionality into Jenkins, for Sonargraph versions 8 and 9	2.2.1
Splunk	This plugin integrates Sonargraph functionality into Jenkins	1.6.4
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
StepCounter	Captures Statistics related to Jenkins Builds, Build Step, SCM checkouts, Jobs and Queue and sends them where you want.	2.0.3
Summary Display	This plugin is Step Counter plugin.	2.0.0
TAP	This plugin shows ACI reports.	1.15
Task Scanner	The Task Scanner Plug-in reached end-of-life. All functionality has been integrated into the Warnings Next Generation Plugin .	2.2.2
Tattletale	Integration plugin for Tattletale analysis tool.	4.53
Test Results Analyzer	This plugin shows history of test execution results in a tabular or graphical format.	0.3
Test stability history	This plugin displays test stabilities - i.e. the history of failed tests.	0.3.5
Testability Explorer	Displays test stabilities - i.e. the history of failed tests.	2.3
TestComplete support	Plugin for the Testability Explorer http://code.google.com/p/testability-explorer/	0.4

Install without restart Download now and install after restart Update information obtained: 14 min ago Check now

- 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 interface. At the top, there's a navigation bar with icons for back, forward, and search, along with user information for 'admin' and a log out link. Below the header, the main title is 'Installing Plugins/Upgrades'. On the left, there's a sidebar with links to 'Back to Dashboard', 'Manage Jenkins', and 'Manage Plugins'. The main content area has a section titled 'Preparation' with a bulleted list: 'Checking internet connectivity', 'Checking update center connectivity', and 'Success'. Below this is a table listing various Jenkins plugins with their current 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 two informational paragraphs with arrows: one pointing to the top page with the text '(you can start using the installed plugins right away)' and another pointing to a checkbox labeled 'Restart Jenkins when installation is complete and no jobs are running'. A footer at the very bottom indicates the page was generated on April 16, 2019, at 7:51:39 PM UTC, with links to the REST API and Jenkins version 2.164.1.

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

The screenshot shows the Jenkins Manage Jenkins interface. On the left, there's a sidebar with links like New Item, People, Build History, Manage Jenkins (which is selected), My Views, Lockable Resources, Credentials, and New View. Below these are two collapsed sections: Build Queue (No builds in the queue) and Build Executor Status (1 Idle, 2 Idle). The main content area has a header "Manage Jenkins" and a message about a new Jenkins version (2.164.2) available for download. It lists several security warnings for 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
- 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

At the bottom, there are four configuration links with icons:

- Configure System (gear icon)
- Configure Global Security (padlock icon)
- Configure Credentials (key icon)
- Global Tool Configuration (wrench icon)

- 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. The left sidebar lists several tools: Gradle, Mercurial, SonarScanner for MSBuild, SonarQube Scanner, Ant, and Maven. Under the Maven section, there is a form for adding a new Maven installation. The 'Name' field contains 'petclinic'. A checkbox labeled 'Install automatically' is checked. Below the form, there is a link 'Install from Apache' and a dropdown menu showing 'Version 3.6.1'. At the bottom of the page are 'Save' and 'Apply' buttons, and a red 'Delete Installer' button.

- 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 'Extract *.zip/*.tar.gz' installer is selected, with its download URL set to https://github.com/zaproxy/zaproxy/releases/download/2.6.0/ZAP_2.6.0_Linux.tar.gz. The subdirectory for extraction is specified as 'ZAP_2.6.0'. The 'Install automatically' checkbox is checked. Buttons for 'Delete Installer' and 'Delete Custom tool' are visible.

Slack & Hubot integration with Jenkins

[Return to Table of Contents](#)

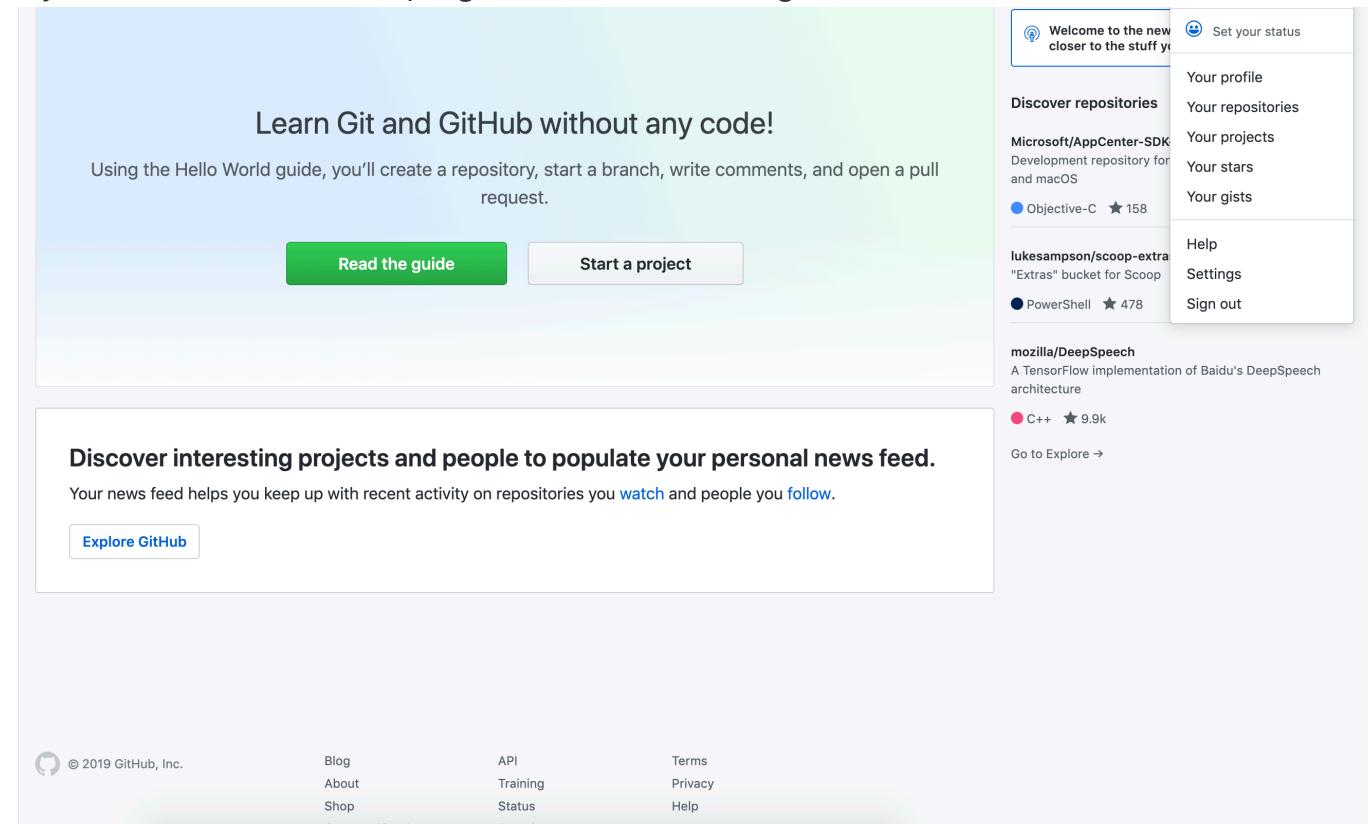
Deploying test code in DevOps Pipeline

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

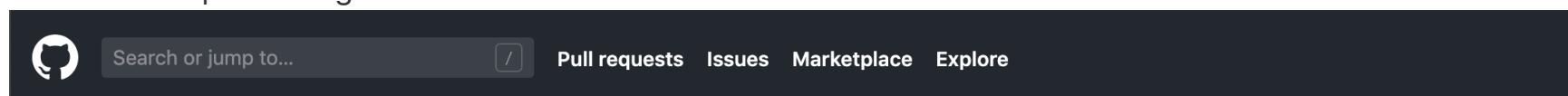


The screenshot shows the GitHub homepage. On the right side, there is a vertical user menu with the following items:

- Welcome to the new closer to the stuff you
- Set your status
- Your profile
- Your repositories
- Your projects
- Your stars
- Your gists
- Help
- Settings
- Sign out

The main content area features a large banner with the text "Learn Git and GitHub without any code!" and a "Read the guide" button. Below this, there is a section titled "Discover interesting projects and people to populate your personal news feed." with a "Explore GitHub" button. At the bottom of the page, there is a footer with links to GitHub's blog, API, terms, and privacy policies.

- Select Developer settings from the left menu list



The screenshot shows the GitHub navigation bar. It includes a search bar, a pull requests button, an issues button, a marketplace button, and an explore button. The developer settings link is located in the top right corner of the navigation bar.

Personal settings

Profile

Account

Emails

Notifications

Billing

SSH and GPG keys

Security

Sessions

Blocked users

Repositories

Organizations

Saved replies

Applications

Developer settings

Public profile

Name

Public email

Select a verified email to display ▾

You have set your email address to private. To toggle email privacy, go to [email settings](#) and uncheck "Keep my email address private."

Bio

Tell us a little bit about yourself

You can @mention other users and organizations to link to them.

URL

Company

You can @mention your company's GitHub organization to link it.

Location

All of the fields on this page are optional and can be deleted at any time, and by filling them out, you're giving us consent to share this data wherever your user profile appears. Please see our [privacy statement](#) to learn more about how we use this information.

Update profile

- Select Person Access Tokens
- Click Generate new token

The screenshot shows the GitHub developer settings interface. At the top, there is a navigation bar with links for Pull requests, Issues, Marketplace, and Explore. Below the navigation bar, the title "Settings / Developer settings" is displayed. On the left, a sidebar contains three options: OAuth Apps, GitHub Apps, and Personal access tokens, with "Personal access tokens" being the active tab. The main content area is titled "Personal access tokens" and features a "Generate new token" button. A note below the button states: "Need an API token for scripts or testing? [Generate a personal access token](#) for quick access to the [GitHub API](#). Personal access tokens function like ordinary OAuth access tokens. They can be used instead of a password for Git over HTTPS, or can be used to [authenticate to the API over Basic Authentication](#)." At the bottom of the page, there is footer information including copyright details (© 2019 GitHub, Inc.), links to Terms, Privacy, Security, Status, Help, Contact GitHub, Pricing, API, Training, Blog, and About.

- 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

The screenshot shows the GitHub developer settings page. The left sidebar has tabs for OAuth Apps, GitHub Apps, and Personal access tokens, with Personal access tokens selected. The main area is titled "New personal access token". It explains that personal access tokens function like ordinary OAuth access tokens and can be used instead of a password for Git over HTTPS or for API authentication. A "Token description" field contains "microcosm_class". Below it is a "What's this token for?" section. Under "Select scopes", it says "Scopes define the access for personal tokens. [Read more about OAuth scopes.](#)". A large table lists various scope options:

Scope	Description
<input type="checkbox"/> repo	Full control of private repositories
<input type="checkbox"/> repo:status	Access commit status
<input type="checkbox"/> repo_deployment	Access deployment status
<input checked="" type="checkbox"/> public_repo	Access public repositories
<input type="checkbox"/> repo:invite	Access repository invitations
<input type="checkbox"/> admin:org	Full control of orgs and teams, read and write org projects
<input type="checkbox"/> write:org	Read and write org and team membership, read and write org projects
<input type="checkbox"/> read:org	Read org and team membership, read org projects
<input type="checkbox"/> admin:public_key	Full control of user public keys
<input type="checkbox"/> write:public_key	Write user public keys
<input type="checkbox"/> read:public_key	Read user public keys
<input type="checkbox"/> admin:repo_hook	Full control of repository hooks
<input type="checkbox"/> write:repo_hook	Write repository hooks
<input type="checkbox"/> read:repo_hook	Read repository hooks
<input type="checkbox"/> admin:org_hook	Full control of organization hooks
<input type="checkbox"/> gist	Create gists

- 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

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

Manage topics

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.

Pull request | Compare

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 landing page. At the top, there's a logo of a stylized fox. Below it, the text "GitLab Community Edition" and "Open source software to collaborate on code". A brief description follows: "Manage Git repositories with fine-grained access controls that keep your code secure. Perform code reviews and enhance collaboration with merge requests. Each project can also have an issue tracker and a wiki." To the right, there's a modal window titled "Change your password". It contains an error message: "1 error prohibited this user from being saved: • Password is too short (minimum is 8 characters)". There are two input fields: "New password" containing "*****" and "Confirm new password" also containing "*****". A blue button at the bottom right of the modal says "Change your password". Below the modal, there are links: "Didn't receive a confirmation email? Request a new one" and "Already have login and password? Sign in". At the bottom of the main page, there's a navigation bar with links to "Explore", "Help", and "About GitLab".

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

Invalid Login or password.

GitLab Community Edition

Open source software to collaborate on code

Manage Git repositories with fine-grained access controls that keep your code secure. Perform code reviews and enhance collaboration with merge requests. Each project can also have an issue tracker and a wiki.

Sign in Register

Username or email
root

Password

Remember me [Forgot your password?](#)

Sign in

Explore Help About GitLab

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

New project

A project is where you house your files (repository), plan your work (issues), and publish your documentation (wiki), [among other things](#).

All features are enabled for blank projects, from templates, or when importing, but you can disable them afterward in the project settings.

Information about additional Pages templates and how to install them can be found in our [Pages getting started guide](#).

Tip: You can also create a project from the command line. [Show command](#)

18.218.153.150/import/github/new

- 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 GitLab interface for importing repositories from GitHub. At the top, there's a navigation bar with links for Projects, Groups, Activity, Milestones, Snippets, and a search bar. Below the navigation is a breadcrumb trail: Projects > GitHub import. The main content area is titled "Import repositories from GitHub" and has a sub-header "Select the projects you want to import". A button "Import all repositories" is visible. A table lists one repository being imported:

From GitHub	To GitLab	Status
morleys/spring-petclinic	root/spring-petclinic	Done Go to project

- 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](#)

Project

Details

Administrator > spring-petclinic > Details

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 38a3a416

Clone with SSH
git@4bb00da20027:root/spring

Clone with HTTP
http://4bb00da20027/root/spr

Copy URL to clipboard

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
Vagrantfile	Commented out chef provisioning. Also, added lib...	3 years ago

<< Collapse sidebar

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. Then 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

The screenshot shows the Jenkins main interface. At the top is a dark header bar with the word "Jenkins" in large white letters. Below it is a light gray navigation bar with the word "Jenkins" and a right-pointing arrow. The main content area has a white background and contains a sidebar on the left with several items:

- [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

`petclinic`
» 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`
 - 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, then SAVE

The screenshot shows the Jenkins configuration interface for the 'petclinic' project. The 'Source Code Management' tab is selected. Under 'Repositories', a Git repository is configured with the URL `http://root@18.218.153.150/root/spring-petclinic.git`, credentials `root/*****`, and a Refspec of `*/master`. An 'Add Repository' button is visible. Under 'Branches to build', a branch specifier `*/master` is listed with an 'Add Branch' button. The 'Repository browser' is set to '(Auto)'. In the 'Additional Behaviours' section, there are options for Mercurial and Subversion, both of which are unselected. The 'Build Triggers' section contains a checkbox for 'Trigger builds remotely (e.g., from scripts)' and a note 'Build after other projects are built'. Buttons for 'Save' and 'Apply' are present at the bottom of this section.

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

The screenshot shows the Jenkins dashboard for the Maven project 'petclinic2'. The left sidebar contains links for Back to Dashboard, Status, Changes, Workspace, Build Now, Delete Maven project, Configure, Modules, SonarQube, Rename, and Embeddable Build Status. The main content area displays the title 'Maven project petclinic2' and a 'SonarQube Quality Gate' section with three items: SonarQube (green icon), Workspace (blue folder icon), and Recent Changes (notebook icon). Below this is a 'Permalinks' section listing four recent builds. At the bottom right, there is a footer bar with the text 'Page generated: Apr 17, 2019 5:23:17 PM UTC REST API 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 links for Projects, Issues, Rules, Quality Profiles, and Quality Gates. A search bar and a 'Log in' button are also present. Below the header, the main content area features a section titled 'Continuous Code Quality' with a 'Log in' button and a 'Read documentation' link. To the right, there are metrics: '0 Projects Analyzed', '0 Bugs', '0 Vulnerabilities', and '0 Code Smells'. A horizontal line separates this from the 'Multi-Language' section, which lists over 20 supported programming languages. Another horizontal line separates this from the 'Quality Model' section, which defines three types of issues: Bugs, Vulnerabilities, and Code Smells. Further down, sections for 'Write Clean Code' and 'Fix The Leak' are shown, each with a brief description and a 'Read More' link.

Continuous Code Quality

Log in Read documentation

0 Projects Analyzed

0 Bugs
0 Vulnerabilities
0 Code Smells

Multi-Language

20+ programming languages are supported by SonarQube thanks to our in-house code analyzers, including:

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	PL/I	

Quality Model

⌘ **Bugs** track code that is demonstrably wrong or highly likely to yield unexpected behavior.

⌚ **Vulnerabilities** are raised on code that is potentially vulnerable to exploitation by hackers.

⌚ **Code Smells** will confuse maintainers or give them pause. They are measured primarily in terms of the time they will take to fix.

Write Clean Code

By fixing new issues as they appear in code, you create and maintain a clean code base. Even on legacy projects, focusing on keeping new code clean will eventually yield a code base you can be proud of.

[Read More](#)

Fix The Leak

The water leak paradigm and the default Quality Gate are based on the leak period - the recent period against which you're tracking issues. For some previous_version makes the most sense, for others the last 30 days is a good option.

[Read More](#)

- Enter a name for your token generation

The screenshot shows the SonarQube web interface. At the top, there's a navigation bar with links for 'Projects', 'Issues', 'Rules', 'Quality Profiles', 'Quality Gates', and 'Administration'. A search bar at the top right contains the placeholder 'Search for projects, sub-projects and files...'. Below the navigation bar, there are buttons for 'My Favorites' and 'All'. A 'Filters' section is present. The main content area displays a welcome message: 'Once you analyze some projects, they will show up here.' Below this, a 'Welcome to SonarQube!' message says: 'Want to quickly analyze a first project? Follow these 2 easy steps.' Step 1, 'Provide a token', shows a text input field containing 'microcosm_token' and a 'Generate' button. A note below states: 'The token is used to identify you when an analysis is performed. If it has been compromised, you can revoke it at any point of time in your user account.' Step 2, 'Run analysis on your project', is partially visible. In the bottom left, there's a 'Coverage' section with three items: '≥ 80%', '< 80%', and '< 70%', each with a count of 0. On the right, a red warning box says: 'Embedded database should be used for evaluation purpose only. The embedded database will not scale, it will not support upgrading to newer versions of SonarQube, and there is no support for migrating your data out of it into a different database engine.' At the bottom right, there's footer text: 'SonarQube™ technology is powered by SonarSource SA Version 6.7.6 (build 38781) - LGPL v3 - Community - Documentation - Get Support - Plugins - Web API - About'.

- 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

The screenshot shows the SonarQube interface with a dark theme. At the top, there's a navigation bar with tabs for 'sonarqube', 'Projects', 'Issues', 'Rules', 'Quality Profiles', 'Quality Gates', and 'Administration'. A search bar at the top right contains the placeholder 'Search for projects, sub-projects and files...'. Below the navigation bar, there are buttons for 'My Favorites' and 'All'. On the left, there's a 'Filters' section. The main content area has a heading 'Once you analyze some projects, they will show up here.' Below this, a 'Welcome to SonarQube!' message says: 'Want to quickly analyze a first project? Follow these 2 easy steps.' Step 1, 'Provide a token', is shown with a token value 'microcosm_token: 2f21c69460caccc10ebd0506b9593beb95919ed2' and a red 'X' button to delete it. A note below says: 'The token is used to identify you when an analysis is performed. If it has been compromised, you can revoke it at any point of time in your user account.' A 'Continue' button is present. Step 2, 'Run analysis on your project', is also listed. In the bottom left, there's a 'Coverage' section with three categories: '≥ 80%', '< 80%', and '< 70%', each with a count of 0. In the bottom right, a note says: 'Embedded database should be used for evaluation purpose only. The embedded database will not scale, it will not support upgrading to newer versions of SonarQube, and there is no support for migrating your data out of it into a different database engine.' Below this, the footer includes the text 'SonarQube™ technology is powered by SonarSource SA', 'Version 6.7.6 (build 38781) - LGPL v3 - Community - Documentation - Get Support - Plugins - Web API - About', and a link to 'http://localhost:8090/'.

- 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 Configuration page under the 'SonarQube servers' section. It includes fields for Name (SonarQube), Server URL (http://18.222.11.148:9000), and Server authentication token. A note indicates that if checked, job administrators will be able to inject SonarQube server configuration as environment variables. Buttons at the bottom include Save and Apply.

Jenkins > configuration

Build Queue
No builds in the queue.

Build Executor Status
1 Idle
2 Idle

of executors: 2

Labels:

Usage: Use this node as much as possible

Quiet period: 5

SCM checkout retry count: 0

Restrict project naming:

Global properties:

- Disable deferred wipeout on this node
- Environment variables
- Prepare jobs environment
- Tool Locations

SonarQube servers

Environment variables: Enable injection of SonarQube server configuration as build environment variables

SonarQube installations

Name: SonarQube

Server URL: http://18.222.11.148:9000

Server authentication token: Default is http://localhost:9000

Version of sonar-maven-plugin:

Additional arguments:

Additional analysis properties: Additional command line arguments to be passed to the SonarQube scanner. For example, -X.

Additional analysis properties in the form of key-value pairs. For example, sonar.analysis.mode=issues.

Save Apply

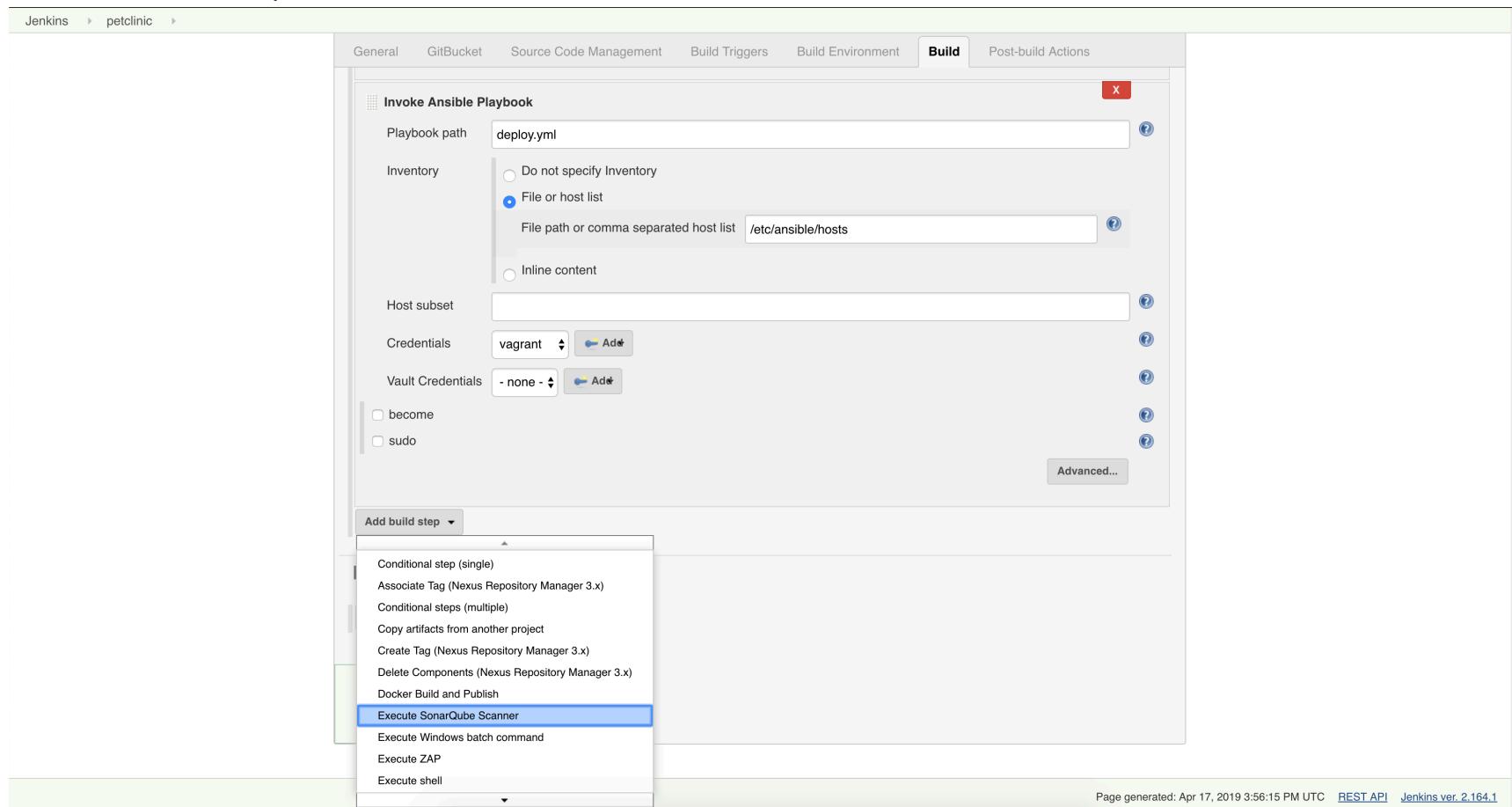
- Go to Manage Jenkins -> Global Tool Configuration
- Under SonarQube Scanner, Select Add SonarQube Scanner

- 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 interface. In the top navigation bar, 'Jenkins' and 'Global Tool Configuration' are visible. A search bar contains the text 'sonar'. Below the search bar, there are sections for Gradle, Mercurial, and SonarScanner for MSBuild. The SonarScanner for MSBuild section is currently active, showing a list of SonarQube Scanner installations. A dropdown menu is open over the list, showing a scrollable list of SonarQube Scanner versions. The version 'SonarQube Scanner 3.3.0.1492' is highlighted with a yellow background. Other versions listed 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. At the bottom of the dropdown menu are 'Save' and 'Apply' buttons.

- Return to the PetClinic configuration screen

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

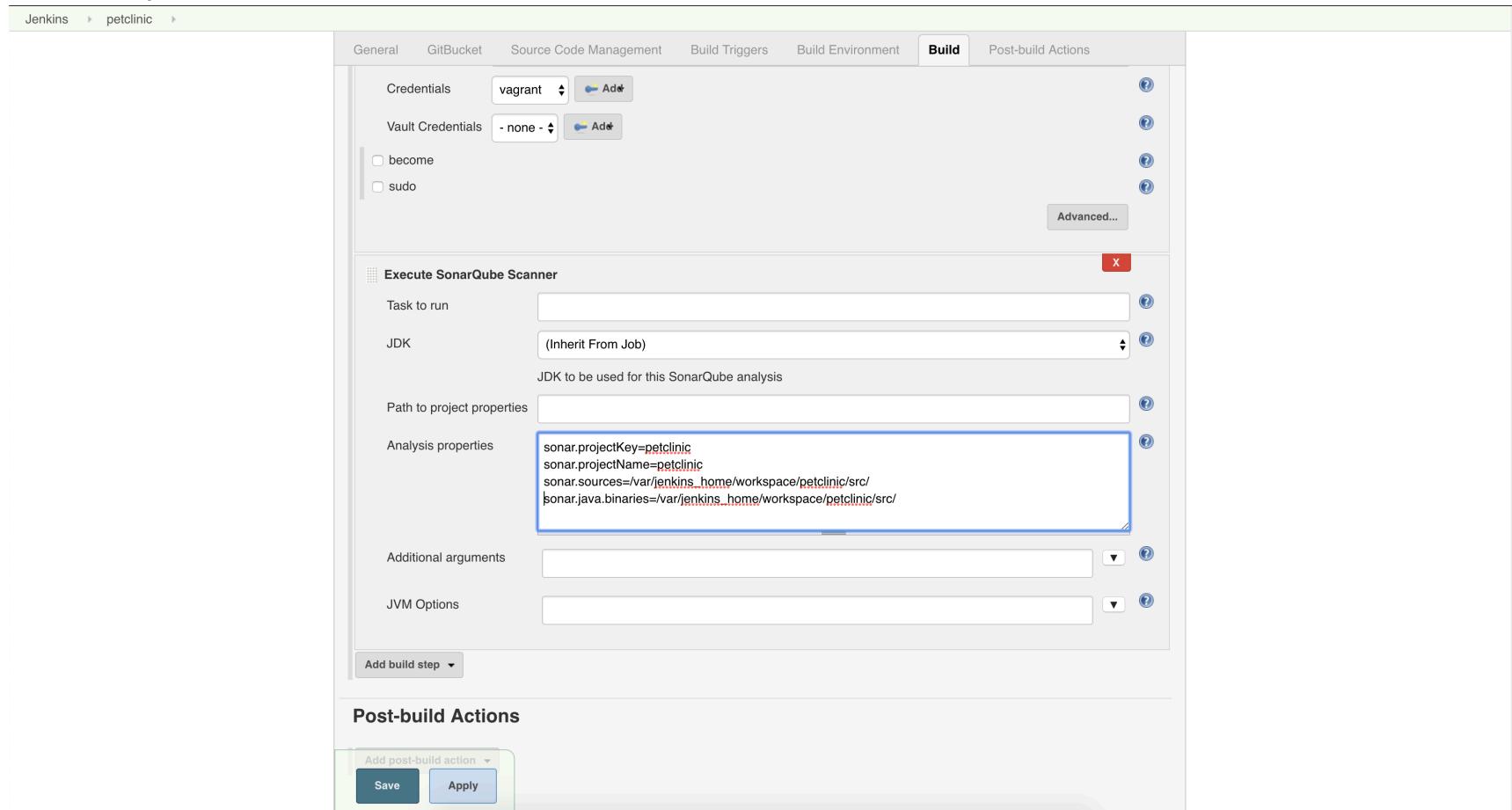


- Under "Analysis properties" enter:

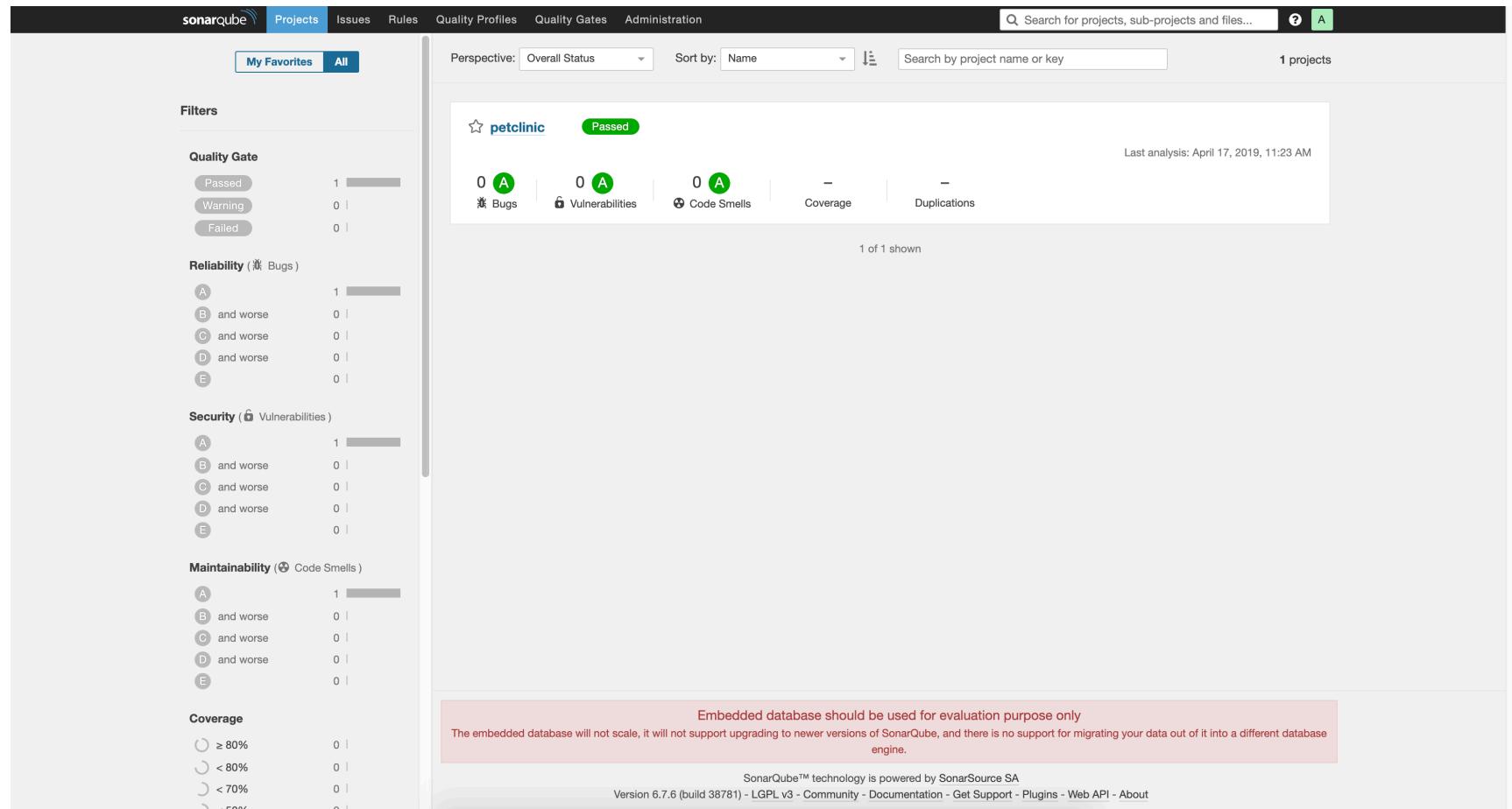
```
sonar.projectKey=petclinic
sonar.projectName=petclinic
sonar.sources=/var/jenkins_home/workspace/petclinic/src/
```

```
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.

Search IAM

Dashboard

Groups

Users

Roles

Policies

Identity providers

Account settings

Credential report

Encryption keys

Roles

What are IAM roles?

IAM roles are a secure way to grant permissions to entities that you trust. Examples of entities include the following:

- IAM user in another account
- Application code running on an EC2 instance that needs to perform actions on AWS resources
- An AWS service that needs to act on resources in your account to provide its features
- Users from a corporate directory who use identity federation with SAML

IAM roles issue keys that are valid for short durations, making them a more secure way to grant access.

Additional resources:

- [IAM Roles FAQ](#)
- [IAM Roles Documentation](#)
- [Tutorial: Setting Up Cross Account Access](#)
- [Common Scenarios for Roles](#)

Create role

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

CloudFront

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



▼ Attached permissions policies

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

Filter policies ▾		Search	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	×

- 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 Edit
Role description	Allows CodeDeploy to read S3 objects, invoke Lambda functions, publish to SNS topics, and update ECS services on your beh
Instance Profile ARNs	Edit
Path	/
Creation time	2019-04-23 13:29 MDT
Maximum CLI/API session duration	1 hour Edit

Permissions Trust relationships Tags Access Advisor 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](#)

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.

[Cancel and Exit](#)

tomcat

Quick Start (0)

My AMIs (0)

AWS Marketplace (30)

Community AMIs (277)

Categories

- All Categories
- Infrastructure Software (24)
- DevOps (7)
- Business Applications (8)

Operating System

Clear Filter

All Marketplace

Tomcat Certified by Bitnami

★★★★★ (7) | 8.5.39-3 on Ubuntu 16.04 | By Bitnami

\$0.00/hr for software + AWS usage fees

Linux/Unix, Ubuntu 16.04 | 64-bit (x86) Amazon Machine Image (AMI) | Updated: 4/12/19

Tomcat is an open source Java servlet container and one of the most widely adopted application servers in the world. The Bitnami Tomcat stack greatly simplifies the development and ...

Product highlights:

- Completely configured and ready to run out of the box.
- Includes an Apache webserver, MySQL, and Java, as well as all other required libraries dependencies.
- Regularly updates with the latest stable release of each component in the stack.

Tomcat is an open source Java servlet container and one of the most widely adopted application servers in the world. The Bitnami Tomcat stack greatly simplifies the development and deployment of applications based on Tomcat. It consists of ready-to-run versions of Apache, MySQL, Tomcat, Java, and all of the other software required to run each of those components. The Bitnami Tomcat Stack is completely integrated and configured, so you'll be ready to start developing your application as soon as the image is launched.

Why use Bitnami Certified Apps? Bitnami certifies that our images are secure, up-to-date, and packaged using industry best practices. With Bitnami you can trust what's in the app you're launching. We monitor all components and libraries for vulnerabilities, outdated components, and application updates. When one is reported, we update and release every affected listing within a couple days at most.

[Tomcat Certified by Bitnami product detail page on AWS Marketplace](#)

Select

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

Tomcat Certified by Bitnami



Free tier eligible

Tomcat Certified by Bitnami

Tomcat is an open source Java servlet container and one of the most widely adopted application servers in the world. The Bitnami Tomcat stack greatly simplifies the development and deployment of applications based on Tomcat. It consists of ready-to-run versions of Apache, MySQL, Tomcat, Java, and all of the other software required to run each of ...

[More info](#)

[View Additional Details in AWS Marketplace](#)

Product Details

By	Bitnami
Customer Rating	 (7)
Latest Version	8.5.39-3 on Ubuntu 16.04
Base Operating System	Linux/Unix, Ubuntu 16.04
Delivery Method	64-bit (x86) Amazon Machine Image (AMI)
License Agreement	End User License Agreement
On Marketplace Since	9/16/14
AWS Services Required	Amazon EC2, Amazon EBS

Highlights

- Completely configured and ready to run out of the box.
- Includes an Apache webserver, MySQL, and Java, as well as all other required libraries dependencies.

Pricing Details

Hourly Fees

Instance Type	Software	EC2	Total
t2.micro	\$0.00	\$0.012	\$0.012/hr
t2.small	\$0.00	\$0.023	\$0.023/hr
t2.medium	\$0.00	\$0.046	\$0.046/hr
t2.large	\$0.00	\$0.093	\$0.093/hr
t2.xlarge	\$0.00	\$0.186	\$0.186/hr
t2.2xlarge	\$0.00	\$0.371	\$0.371/hr
t3.micro	\$0.00	\$0.01	\$0.01/hr
t3.small	\$0.00	\$0.021	\$0.021/hr
t3.medium	\$0.00	\$0.042	\$0.042/hr
t3.large	\$0.00	\$0.083	\$0.083/hr
t3.xlarge	\$0.00	\$0.166	\$0.166/hr
t3.2xlarge	\$0.00	\$0.333	\$0.333/hr
m4.large	\$0.00	\$0.10	\$0.10/hr
m4.xlarge	\$0.00	\$0.20	\$0.20/hr
m4.2xlarge	\$0.00	\$0.40	\$0.40/hr
m4.4xlarge	\$0.00	\$0.80	\$0.80/hr
m4.10xlarge	\$0.00	\$2.00	\$2.00/hr
m4.16xlarge	\$0.00	\$3.20	\$3.20/hr
c5.large	\$0.00	\$0.085	\$0.085/hr
c5.xlarge	\$0.00	\$0.17	\$0.17/hr
c5.2xlarge	\$0.00	\$0.34	\$0.34/hr

[Cancel](#) [Continue](#)

- 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 various features such as pre-installed software, network connectivity, and storage options.

[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 <small>Free tier eligible</small>	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 your Student VPC
 - Subnet: Select either subnet
 - Auto-Assign public IP: Select Enable (or leave the default if it is Enable)
 - **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

The screenshot shows the AWS EC2 Launch Instance Wizard at Step 3: Configure Instance Details. The URL in the browser is https://us-east-2.console.aws.amazon.com/ec2/v2/home?region=us-east-2#LaunchInstanceWizard:3. The top navigation bar includes the AWS logo, Services dropdown, Resource Groups dropdown, and a search icon. Below the navigation, a breadcrumb trail shows the steps: 1. Choose AMI, 2. Choose Instance Type, 3. Configure Instance (which is highlighted in orange), 4. Add Storage, 5. Add Tags, 6. Configure Security Group, and 7. Review.

Step 3: Configure Instance Details

Subnet: subnet-04d355bb02dc06049 | tst1 | us-east-2a (dropdown) [Create new subnet](#)

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

Placement group: Add instance to placement group

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

IAM role: EC2_Plus_S3 [C Create new IAM role](#)

Shutdown behavior: Stop (dropdown)

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-04d355b1 ▾	Auto-assign	Add IP	Add IP

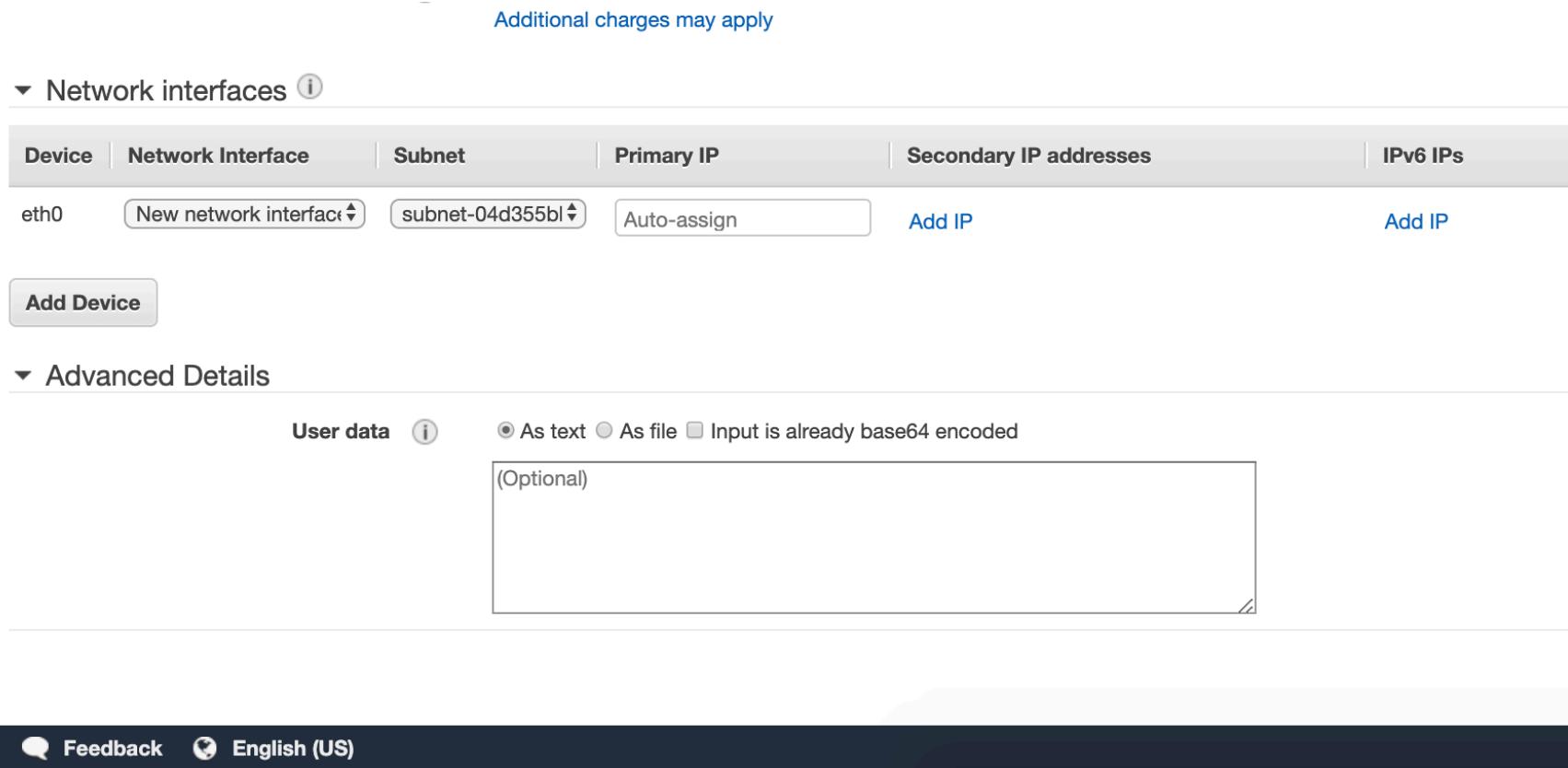
Add Device

▼ 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"/>	
Add another tag (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

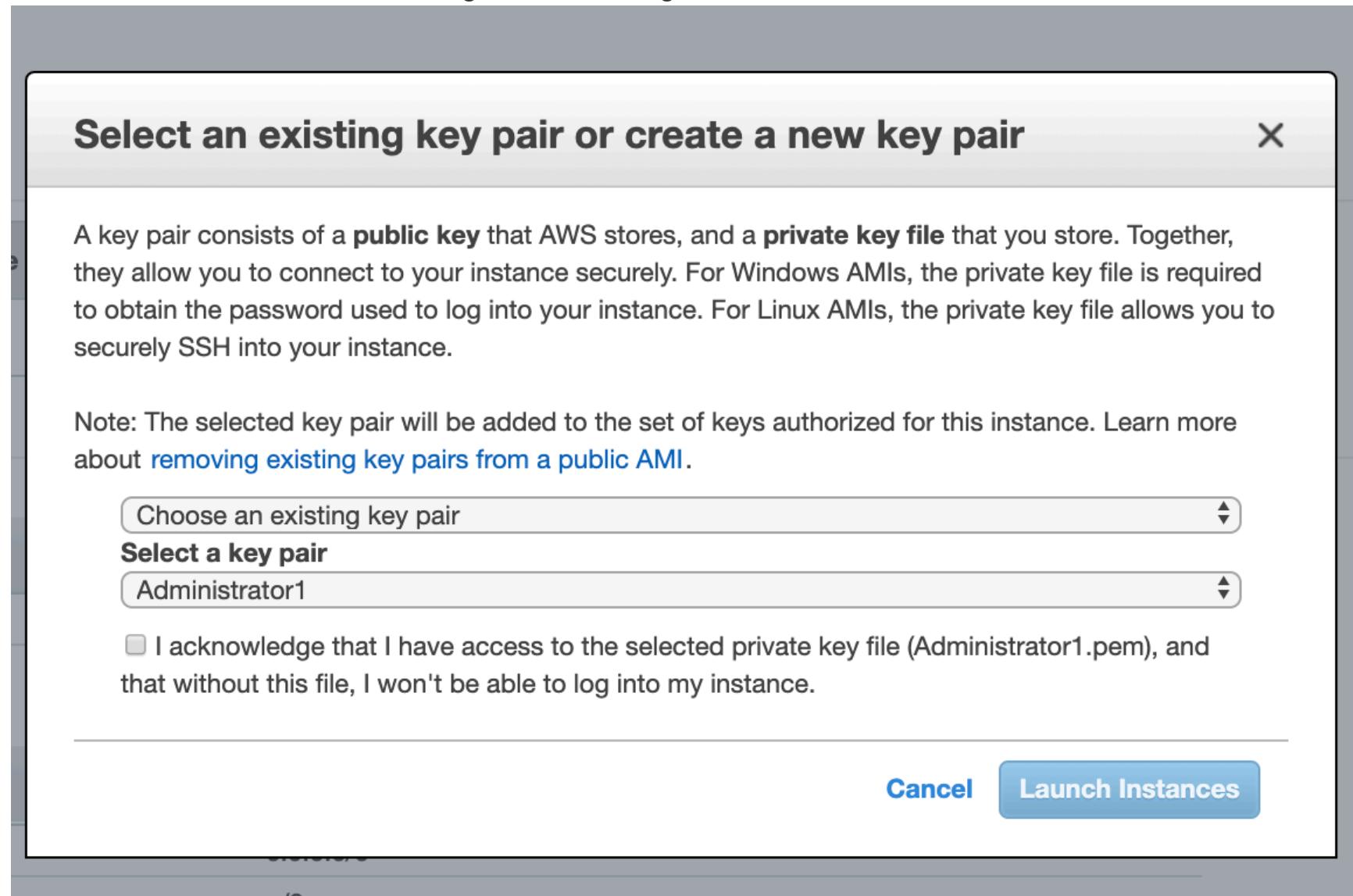
Select an existing security group

Security Group ID	Name	Description
<input checked="" type="checkbox"/> sg-0e1a263967d2471f2	default	default VPC security group

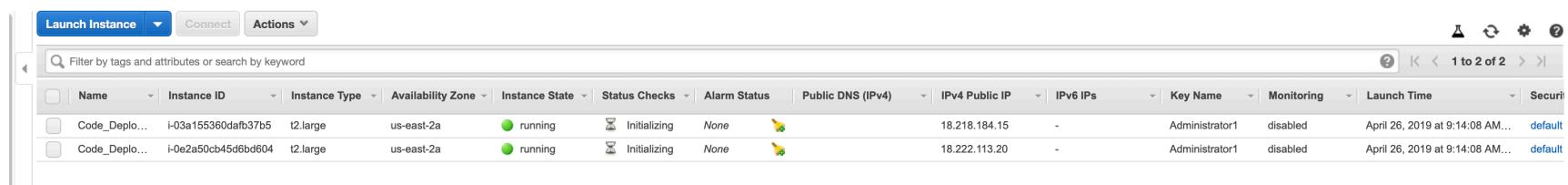
Inbound rules for sg-0e1a263967d2471f2 (Selected security groups: sg-0e1a263967d2471f2)

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



The screenshot shows the AWS EC2 Instances page. At the top, there are buttons for 'Launch Instance', 'Connect', and 'Actions'. Below the header is a search bar with the placeholder 'Filter by tags and attributes or search by keyword'. The main area is a table with the following columns: 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, and Security Group. There are two rows of data:

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 Group
Code_Deplo...	i-03a155360dabf37b5	t2.large	us-east-2a	running	Initializing	None	ec2-18-218-184-15.eca3.compute.amazonaws.com	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.eca3.compute.amazonaws.com	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 i	ec2betclinic	⚠ Load balancer names must only contain alphanumeric characters or hyphens, and not start with a hyphen.
Scheme i	<input checked="" type="radio"/> internet-facing <input type="radio"/> internal	
IP address type i	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
HTTP	8080

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

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 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-0e1a263967d2471f2	default	default VPC security group

- Step 4, Create a new Target Group
 - Name: create a memorable name (e.g. StudentCodeDeployTG8080)
 - Port: 80
 - Path: /index.html
- 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 i New target group ▼

Name i TGcodedeployec2petclinic ⚠ TargetGroup name cannot contain characters that are not letters, numbers, or underscores.

Target type i Instance IP Lambda function

Protocol i HTTP ▼

Port i 8080

Health checks

Protocol i HTTP ▼

Path i /petclinic

▼ Advanced health check settings

Port i traffic port override

Healthy threshold	<input type="text" value="5"/>
Unhealthy threshold	<input type="text" value="2"/>
Timeout	<input type="text" value="5"/> seconds
Interval	<input type="text" value="30"/> seconds
Success codes	<input type="text" value="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.

<input type="checkbox"/>	Instance	Name	Port	State	Security groups	Zone
<input type="checkbox"/>	i-03a155360dafb37b5	Code_Deploy_Instances	8080	● running	default	us-east-2a
<input type="checkbox"/>	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.

Add to registered	on port 8080						
<input style="width: 100%;" type="text" value="Search Instances"/>							
<input type="checkbox"/>	Instance	Name	State	Security groups	Zone	Subnet ID	Subnet CIDR
<input checked="" type="checkbox"/>	i-03a155360dafb37b5	Code_Deploy_Instances	● running	default	us-east-2a	subnet-04d355bb02dc06049	10.0.0.0/24
<input checked="" type="checkbox"/>	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

[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

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

Manage public bucket policies for this bucket

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

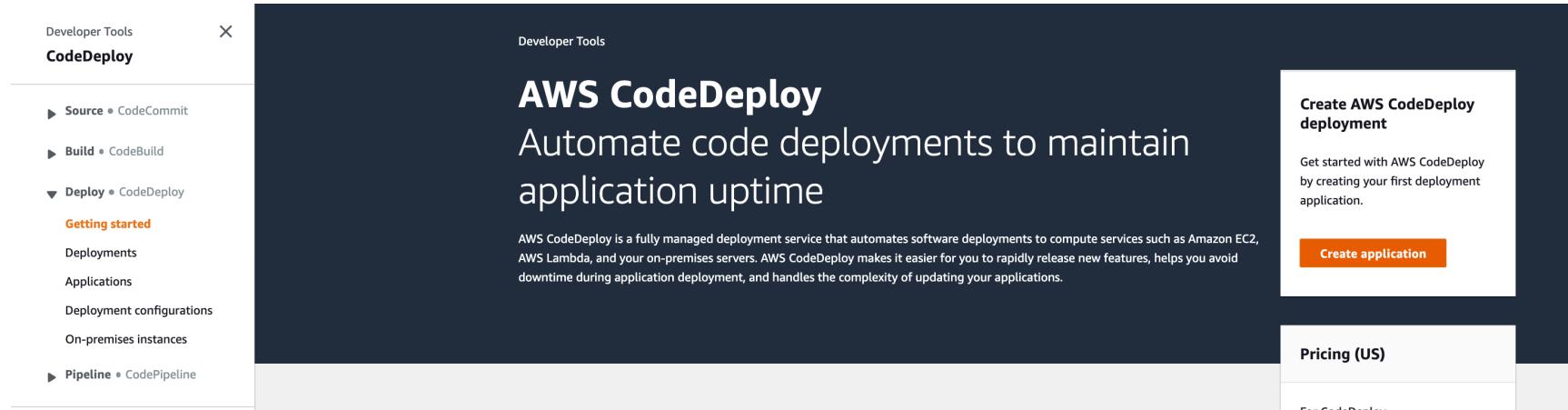
Manage system permissions

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 bar indicates the path: Developer Tools > CodeDeploy > Applications > Create application. Below the breadcrumb, the title 'Create application' is displayed. A section titled 'Application configuration' contains fields for 'Application name' (with placeholder 'Enter an application name' and value 'petclinic') and 'Compute platform' (with placeholder 'Choose a compute platform' and value 'EC2/On-premises'). At the bottom right, there are 'Cancel' and 'Create application' buttons.

Application name
Enter an application name
petclinic
100 character limit

Compute platform
Choose a compute platform
EC2/On-premises

Cancel **Create application**

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

The screenshot shows the 'Application' configuration page. On the left, a sidebar menu has 'Application' selected. The main content area displays the word 'Application'. This is likely a preview or confirmation step after creating the 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

Blue/green

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

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

Name



Code_Deploy_Instances



Remove tag

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 'Developer Tools' at the top, followed by 'CodeDeploy'. Under 'CodeDeploy', there are several options: 'Source' (CodeCommit), 'Build' (CodeBuild), 'Deploy' (CodeDeploy), 'Getting started', 'Deployments', 'Applications' (which is expanded to show 'Application'), 'Deployment configurations', 'On-premises instances', 'Pipeline' (CodePipeline), 'Feedback', and 'Return to the old experience'. The 'Application' section is highlighted in orange. In the main content area, a green banner at the top says 'Success Deployment group created'. Below it, the breadcrumb trail is 'Developer Tools > CodeDeploy > Applications > petclinic > petclinicDepGrp'. The title 'petclinicDepGrp' is displayed, with 'Edit', 'Delete', and 'Create deployment' buttons to its right. A section titled 'Deployment group details' contains the following information:

Deployment group name	Application name	Compute platform
petclinicDepGrp	petclinic	EC2/On-premises
Deployment type	Service role ARN	Deployment configuration
In-place	arn:aws:iam::443007076818:role/CodeDeployServiceRole	CodeDeployDefault.OneAtATime
Rollback enabled		
True		

Below this is a section titled 'Environment configuration: Amazon EC2 instances' with a table:

Key	Value
Name	Code_Deploy_Instances

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-microcosmsignin.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



A screenshot of the AWS IAM User Management console. At the top, there are two buttons: 'Add user' (blue) and 'Delete user' (red). Below them is a search bar with the placeholder 'Find users by username or access key'. The main area shows a table with two columns: 'User name' and 'Groups'. The first row lists 'Administrator' under 'User name' and 'Administrators' under 'Groups'. There are checkboxes next to each column header and each row item.

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

The screenshot shows the AWS IAM User Summary page for the user 'Administrator'. The left sidebar has a 'Users' tab selected. The main content area displays the User ARN (arn:aws:iam::443007076818:user/Administrator), Path (/), and Creation time (2019-04-10 10:04 MDT). Below this, tabs for 'Permissions', 'Groups (1)', 'Tags (1)', 'Security credentials' (which is selected), and 'Access Advisor' are visible. The 'Sign-in credentials' section shows a summary with a console sign-in link (https://cert-microcosm.signin.aws.amazon.com/console) and indicates the password is enabled (last signed in Today). It also shows that no MFA device is assigned and no signing certificates are present. The 'Access keys' section provides instructions for using access keys to make secure requests to AWS services. A 'Create access key' button is available. A single access key entry is listed: Access key ID AKIAWOKHRHJCDYDAKNE, Created 2019-04-10 10:04 MDT, and Last used 2019-04-26 20:32 MDT with codedeploy in us-east-2.

Access key ID	Created	Last used
AKIAWOKHRHJCDYDAKNE	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. The dialog has a green header bar with the text 'Success'. Below it, a message states: '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.' A 'Download .csv file' button is available. The table displays the generated access key details:

Access key ID	Secret access key
AKIAWOJKHRHJASUYR35N	7ar8nUSG1C2gstsL9x3H6qBf46ta202Man6/wRXF Hide

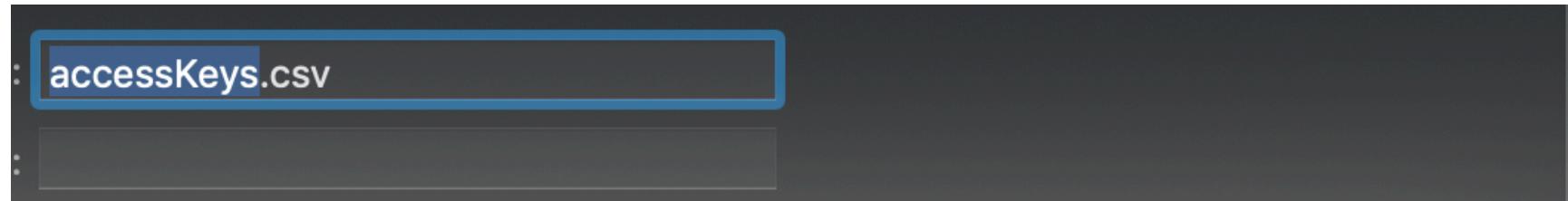
A 'Close' button is located at the bottom right of the dialog.

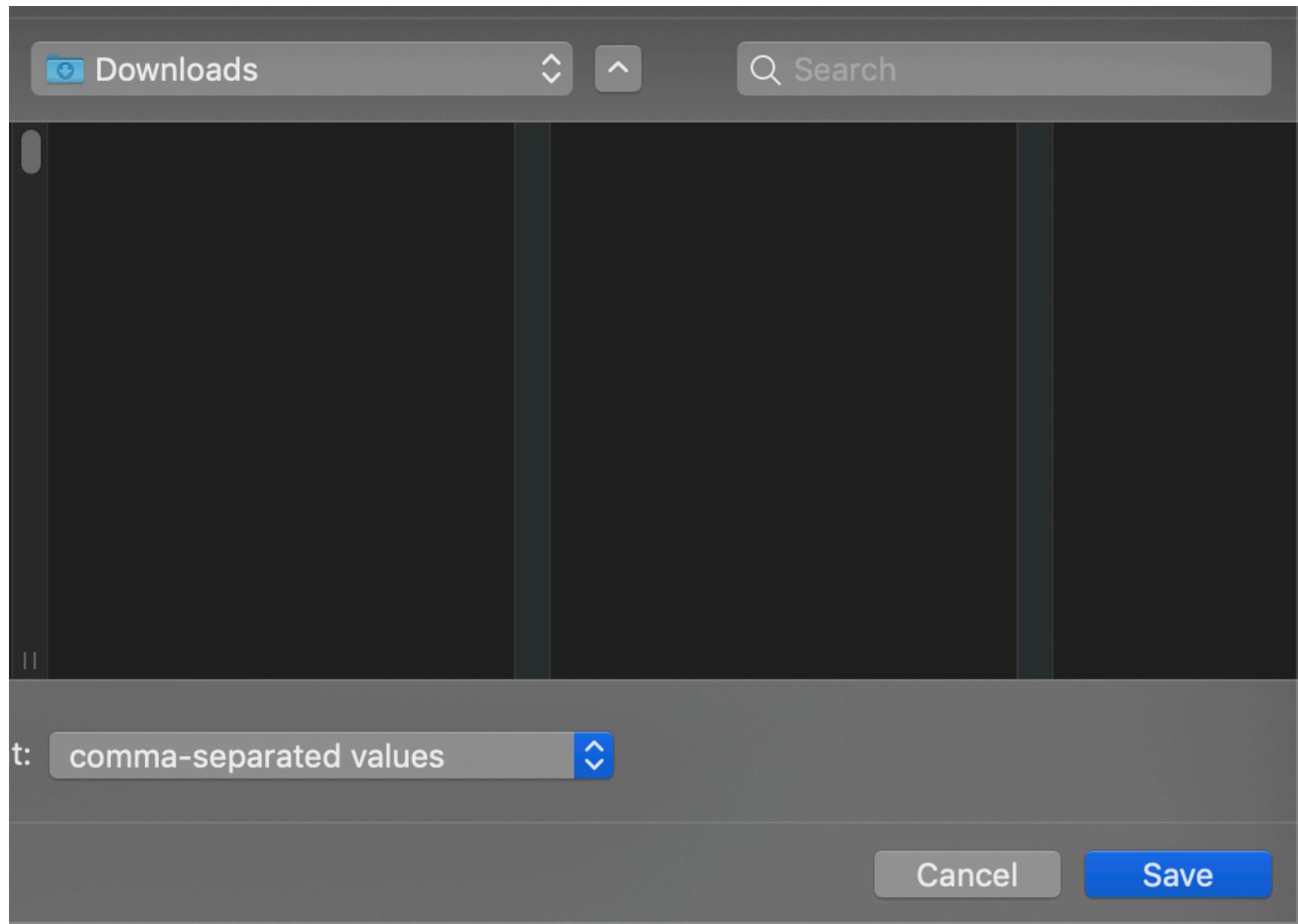
Access keys
Use access keys to make secure REST or HTTP Query protocol requests to AWS service APIs. For your protection, you should never share your secret keys with anyone. As rotation. [Learn more](#)

[Create access key](#)

Access key ID	Created	Last used
AKIAWOJKHRHJASUYR35N	2019-04-27 07:27 MDT	N/A

- 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
AKIAWOJKHRHJASUYR35N	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 to 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

Build Settings **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	

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.

AWS Access Key ID
AKIAWOJKHRHJCDYDAKNE

AWS Secret Key
.....

Save Apply

Build and Deploy

- From your Project Page in Jenkins select Build

Jenkins ➤ petclinic ➤

 Back to Dashboard

 Status

 Changes

 Workspace

 Build Now

 Delete Maven project

[Delete Maven project](#)[Configure](#)[Modules](#)[Favorite](#)[SonarQube](#)[Open Blue Ocean](#)[Rename](#)[Embeddable Build Status](#)**Build History**[trend ▾](#)

TINA

X



#43

Apr 26, 2019 3:35 PM



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

Amazon S3 > petclinicdeploy > target

Overview



Type a prefix and press Enter to search. Press ESC



Upload



Create folder

Download



	Name ▼
<input type="checkbox"/>	 #43-3400450200434229806.zip
<input type="checkbox"/>	 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 a deployment named **d-9GMQAH25Y**. The deployment status indicates that 0 of 2 instances have been updated, and the process is currently **In progress**. Deployment details show the application is **petclinic**, the deployment ID is **d-9GMQAH25Y**, and the deployment group is **petclinicDepGrp**. Revision details provide the revision location (**s3://petclinicdeploy/target/#54-8254645117549694950.zip?eTag=beff67edee49a1f6645a1beefb4d5fb**), the time of creation (**1 minute ago**), and a description stating the application revision was registered via Jenkins. The deployment lifecycle events table lists two instances: **i-03a155360dafb37b5** and **i-0e2a50cb45d6bd604**, both of which are currently in progress with the most recent event being **BeforeBlockTraffic**.

- 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 interface. On the left, there's a navigation sidebar with links for Source (CodeCommit), Build (CodeBuild), Deploy (CodeDeploy), Getting started, Deployments, Deployment (selected), Applications, Deployment configurations, On-premises instances, Pipeline (CodePipeline), Feedback, and a link to Return to the old experience.

The main content area displays deployment details for the application **petclinic**. It shows the Application ID as **d-9GMQAH25Y**, the Deployment configuration as **custom1**, and the Deployment group as **petclinicDepGrp**. The status is listed as **In progress**. Below this, the **Revision details** section shows the revision location as **s3://petclinicdeploy/target/#54-8254645117549694950.zip?** with an eTag of **eTag=beff67edee49a1f6645a1beeffb4d5fb**, and the revision was created 2 minutes ago.

A detailed table below lists deployment events and their statuses:

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	-	-	-

Below this, a summary table provides key deployment information:

Application	Deployment ID
petclinic	d-FRO04J65Y
Deployment configuration	Deployment group
CodeDeployDefault.AllAtOnce	petclinicDepGrp

The final section, **Revision details**, is currently empty.

Revision location	s3://petclinicdeploy/target/#62-2472083856417875777.zip? eTag=5c72ef1adb2c717b3611effbef97645	Revision created	33 minutes ago
Event	Duration	Status	Error code
BeforeBlockTraffic	less than one second	Succeeded	-
BlockTraffic	22 seconds	Succeeded	-
AfterBlockTraffic	less than one second	Succeeded	-
ApplicationStop	less than one second	Succeeded	-
DownloadBundle	less than one second	Succeeded	-
BeforeInstall	less than one second	Succeeded	-
Install	less than one second	Succeeded	-
AfterInstall	less than one second	Succeeded	-
ApplicationStart	less than one second	Succeeded	-
ValidateService	less than one second	Succeeded	-
BeforeAllowTraffic	less than one second	Succeeded	-
AllowTraffic	29 minutes 21 seconds	Succeeded	-
AfterAllowTraffic	less than one second	Succeeded	-

- 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 ✔ Succeeded
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	✔ Succeeded	AfterAllowTraffic	View events	Apr 26, 2019 3:51 PM	Apr 26, 2019 3:52 PM
i-0cfcb0d38fdf28923	1 minute 0 seconds	✔ Succeeded	AfterAllowTraffic	View events	Apr 26, 2019 3:51 PM	Apr 26, 2019 3:52 PM

The screenshot shows the Pet Clinic application's home page. The header features a green bar with the text "Pet Clinic" and "A Spring Framework Demonstration". To the right of the header is a large "Spring" logo with a leaf icon. Below the header is a navigation bar with five items: "Home", "Find owners", "Who-am-I", "Error", and "Help". The main content area has a large "Welcome" heading and a photograph of a brown puppy and a white cat with blue eyes.

Home Find owners Who-am-I Error Help

Welcome



[Return to Table of Contents](#)

AWS ECS Based Code Deploy

Additional Task Definitions

Task/Service Name	Container Name	Image	Ports	Volumes	Mount Points	Environment Variables
petclinic	tomcat8	tomcat:8-jre8	8080	/webapps	[TODO path to apache 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 Ballancer

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

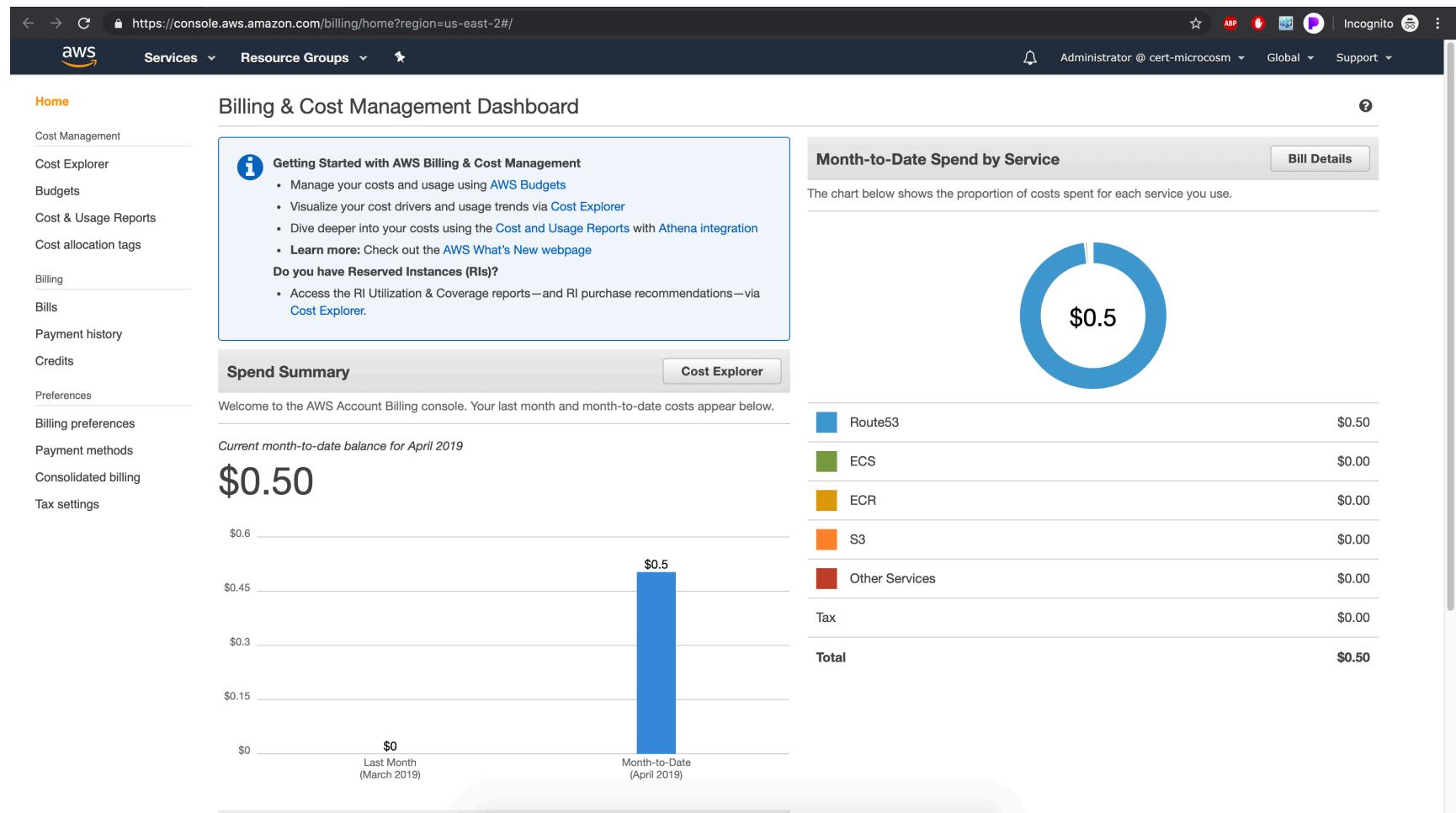
The screenshot shows the AWS Management Console homepage. At the top right, there is a user menu with the following items:

- IAM User: Administrator
- Account: cert-microcosm

Below the user menu, there is a "Access resources" section with a "Billing Dashboard" link highlighted in orange. Other links in this section include "My Account", "My Organization", "My Security Credentials", "Switch Role", and "Sign Out".

The main content area has sections for "AWS services", "Build a solution", and "Explore AWS". The "AWS services" section includes a "Find Services" search bar and a "Recently visited services" list with icons for Billing, IAM, EKS, ECR, and ECS. The "Build a solution" section features three cards: "Launch a virtual machine" (With EC2, 2-3 minutes), "Build a web app" (With Elastic Beanstalk, 6 minutes), and "Build using virtual servers" (With Lightsail, 1-2 minutes). The "Explore AWS" section includes cards for "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 IQ 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.

 Add Credentials

Domain Global credentials (unrestricted)

Kind Username with password

Scope Global (Jenkins, nodes, items, all child items, etc)

Username admin

Password

ID nexusadmin|

Description

- 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`)
 - 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	<input type="text" value="Nexus"/>
Server ID	<input type="text" value="hello1234567890bobmyfriend"/> ?
Server URL	<input type="text" value="http://18.221.105.75:8081/nexus"/>
Credentials	<input type="text" value="admin/*****"/> ▼ Add ▾

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

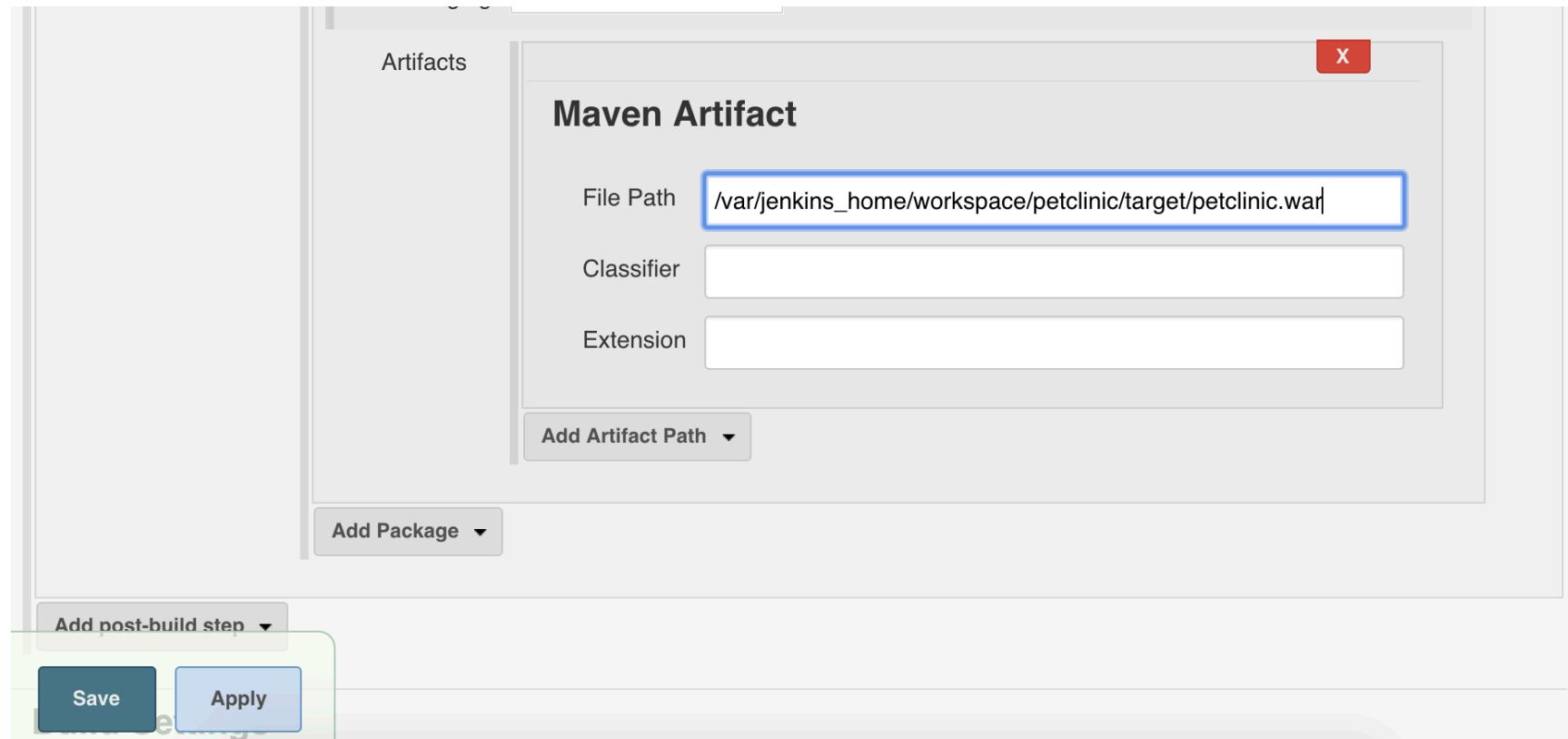
- Return to the PetClinic build configuration page
- Select the Post Steps Tab and click Add post-build step
- Select Nexus Repository Manager Publisher

- 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	<table border="1"><tr><td>Group</td><td>petclinic-main</td><td>X</td></tr><tr><td>Artifact</td><td>petclinic.war</td><td></td></tr><tr><td>Version</td><td>2.3</td><td></td></tr><tr><td>Packaging</td><td>war</td><td></td></tr></table>	Group	petclinic-main	X	Artifact	petclinic.war		Version	2.3		Packaging	war		
Group	petclinic-main	X												
Artifact	petclinic.war													
Version	2.3													
Packaging	war													



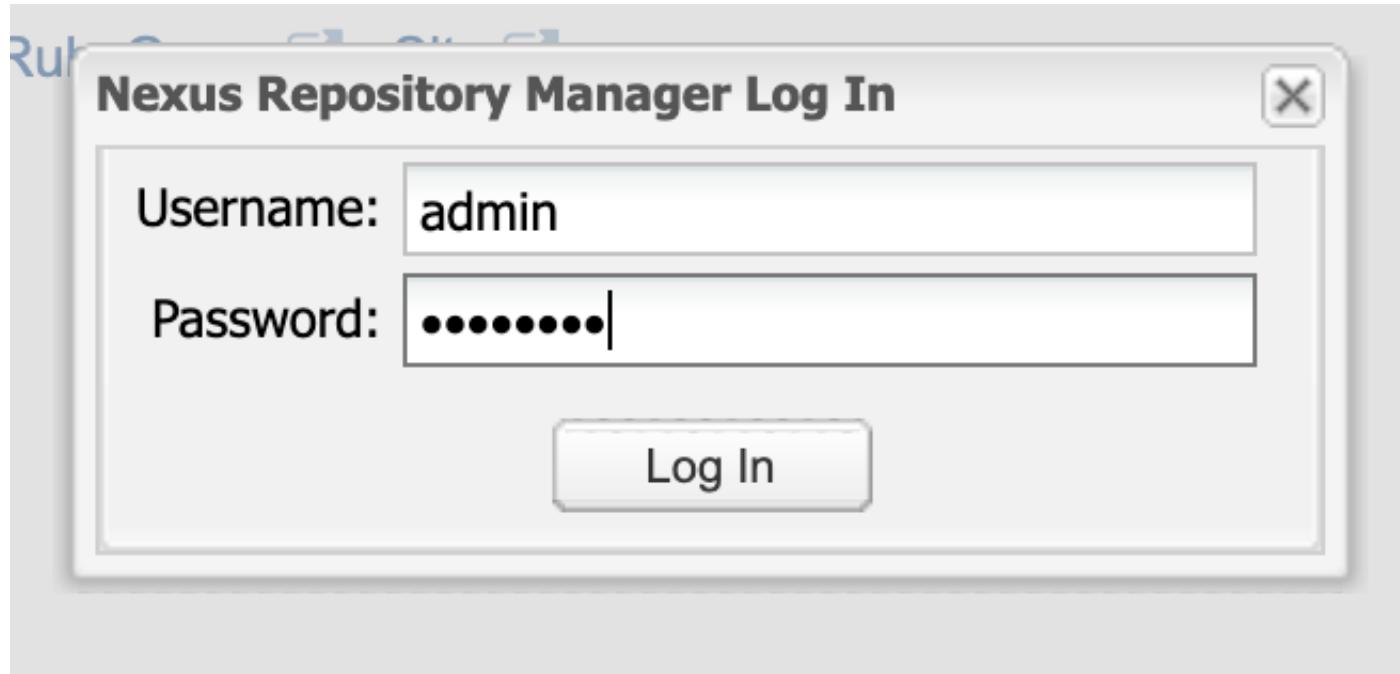
- After a successful Jenkins build, if you look in the build console output, you should see, near the bottom of the page, a message indicating success, similar to:

```
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))

The screenshot shows the Nexus Repository Manager OSS interface. At the top right is a "Log In" button and the text "Nexus Repository Manager OSS 2.14.12-02". The main header features a large green "N" icon followed by the text "Nexus Repository Manager OSS". Below the header is a search bar with placeholder text: "Type in the name of a project, class, or artifact into the text box below, and click Search. Use 'Advanced Search' on the left for more options." To the left of the search bar is a sidebar titled "Sonatype™" containing links for "Artifact Search", "Advanced Search", "Views.Repositories", "Repositories", and "Help". The central content area includes sections for "Get Started" (Configuration, Documentation, Community), "Repository Formats" (Maven, .NET/NuGet, Node/npm, OSGi, P2, RPM/YUM, RubyGems, Site), and a "Go Language Feedback" survey. A sidebar on the right asks, "Which of the following best describes your use of Go?".

- Login using the default credentials admin:admin123



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

Repository	Type	IQ Policy Violations	Health Check	Format	Policy	Repository Status	Repository Path
Public Repositories	group		<button>ANALYZE</button>	maven2		In Service	http://18.221.105.75:8081/nexus/content/groups/public
3rd party	hosted		<button>ANALYZE</button>	maven2	Release	In Service	http://18.221.105.75:8081/nexus/content/repositories/thirdparty
Apache Snapshots	proxy		<button>ANALYZE</button>	maven2	Snapshot	In Service	http://18.221.105.75:8081/nexus/content/repositories/apache-snapshots
Central	proxy		<button>ANALYZE</button>	maven2	Release	In Service	http://18.221.105.75:8081/nexus/content/repositories/central
Central M1 shadow	virtual		<button>ANALYZE</button>	maven1	Release	In Service	http://18.221.105.75:8081/nexus/content/shadows/central-m1
Releases	hosted		<button>ANALYZE</button>	maven2	Release	In Service	http://18.221.105.75:8081/nexus/content/repositories/releases
Snapshots	hosted		<button>ANALYZE</button>	maven2	Snapshot	In Service	http://18.221.105.75:8081/nexus/content/repositories/snapshots

[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

The screenshot shows the AWS Identity and Access Management (IAM) console home page. The left sidebar has a 'Dashboard' section with links for Groups, Users, Roles, Policies, Identity providers, Account settings, Credential report, and Encryption keys. The main content area displays the following:

- 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
 - Customer Managed Policies: 0
 - Roles: 9
 - Identity Providers: 1
- 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 (video thumbnail)
- 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

The screenshot shows the AWS IAM (Identity and Access Management) service in the AWS Management Console. The URL is https://console.aws.amazon.com/iam/home#/users. The left sidebar has a 'Users' section selected, which is highlighted with an orange border. The main content area displays a table with one row of data. The columns are: User name (smorley), Groups (admin), Access key age (5 days), Password age (5 days), Last activity (5 days), and MFA (Not enabled). There are 'Add user' and 'Delete user' buttons at the top of the table. A search bar at the top says 'Find users by username or access key'. The status bar at the bottom indicates 'Showing 1 result'.

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

Add user

Set user details

You can add multiple users at once with the same access type and permissions. [Learn more](#)

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 and existing user, if applicable)
- If the desired group does not yet exist, select Create Group

The screenshot shows the AWS IAM 'Add user' wizard at Step 2: Set permissions. The 'Add user to group' section is active, showing a search results table for the 'admin' group. The table has columns for Group and Attached policies. One result is shown: 'admin' with 'AdministratorAccess' attached.

- 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

The screenshot shows the AWS IAM 'Add user' wizard, Step 2: Create group. A modal window is open for creating a new group named 'Administrators'. The 'Create group' button is visible at the bottom right of the modal. Inside the modal, a table lists various AWS managed policies. The policy 'AdministratorAccess' is selected (indicated by a checked checkbox). Other policies listed include AmazonAPIGatewayAdministrator, AWSAppSyncAdministrator, AWSCloud9Administrator, AWSSSDirectoryAdministrator, AWSSSOMasterAccountAdministrator, AWSSSOMemberAccountAdministrator, DatabaseAdministrator, NetworkAdministrator, and SystemAdministrator. The table columns are Policy name, Type, Used as, and Description.

	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"/>	AWSSSDirectoryAdministrator	AWS managed	None	Administrator access for SSO Directory
<input type="checkbox"/>	AWSSSOMasterAccountAdministrator	AWS managed	None	Provides access within AWS SSO to manage AWS Organizations master and member acco...
<input type="checkbox"/>	AWSSSOMemberAccountAdministrator	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

The screenshot shows the AWS IAM 'Add user' wizard at Step 2: Set permissions. The 'Add user to group' section is active, showing a list of groups. The 'Administrators' group is selected, indicated by a checked checkbox and highlighted with a green border. The 'Attached policies' column shows 'AdministratorAccess' for the selected group. Other groups like 'admin' are listed but not selected.

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

- Add tags if desired
- Click Review

Add user

1 2 3 4 5

Add tags (optional)

IAM tags are key-value pairs you can add to your user. 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 user. [Learn more](#)

Key	Value (optional)	Remove
administrators		x
Add new key		

You can add 49 more tags.

Cancel Previous Next: Review

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

- After reviewing, click Create User

Add user

Review

User details

User name	Administrator
AWS access type	Programmatic access and AWS Management Console access
Console password type	Custom
Require password reset	Yes
Permissions boundary	Permissions boundary is not set

Permissions summary

The user shown above will be added to the following groups.

Type	Name
Group	Administrators

Tags

The new user will receive the following tag

Key	Value
administrators	(empty)

Cancel Previous 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

The screenshot shows the AWS IAM 'Add user' interface. At the top, there are five numbered steps: 1, 2, 3, 4, and 5, with step 5 being the active one. A 'Success' message box is displayed, stating: '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.' Below this message, a link 'Users with AWS Management Console access can sign-in at: <https://cert-microcosm.signin.aws.amazon.com/console>' is provided. A 'Download .csv' button is located above a table. The table has columns: User, Access key ID, Secret access key, and Email login instructions. It lists a single user named 'Administrator'. The 'Access key ID' is 'AKIAWOJKHRHJCODYDAKNE', and the 'Secret access key' is 'ZNHgDgrWBqnBR+K2AxGn csul7m2SIH0Pjk4yqit5'. There are 'Send email' and 'Hide' buttons next to the secret key. A 'Close' button is at the bottom right of the table area. At the very bottom of the page, there are links for 'Feedback', 'English (US)', '© 2008 - 2019, Amazon Web Services, Inc. or its affiliates. All rights reserved.', 'Privacy Policy', and 'Terms of Use'.

	User	Access key ID	Secret access key	Email login instructions
▶	Administrator	AKIAWOJKHRHJCODYDAKNE	ZNHgDgrWBqnBR+K2AxGn csul7m2SIH0Pjk4yqit5	Send email Hide

- 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.

The screenshot shows the AWS IAM User Summary page for a user named 'Administrator'. The left sidebar navigation includes 'Dashboard', 'Groups', 'Users' (which is selected), 'Roles', 'Policies', 'Identity providers', 'Account settings', 'Credential report', 'Encryption keys', and 'Feedback'. The main content area displays the user's ARN (arn:aws:iam::443007076818:user/Administrator), Path (/), and Creation time (2019-04-10 10:04 MDT). Below this, tabs for 'Permissions', 'Groups (1)', 'Tags (1)', 'Security credentials' (which is selected), and 'Access Advisor' are visible. The 'Sign-in credentials' section shows a summary with a console sign-in link. The 'Assigned MFA device' section indicates 'Not assigned'. The 'Signing certificates' section shows 'None'. The 'Access keys' section contains a table with one row for access key ID 'AKIAWOKJKHRHJCDYDAKNE', created on '2019-04-10 10:04 MDT', last used on '2019-04-15 16:13 MDT with servicediscovery in us-east-2', and a status of 'Active'. The 'SSH keys for AWS CodeCommit' section allows for uploading SSH public keys. The bottom navigation bar includes links for 'Feedback', 'English (US)', 'Privacy Policy', and 'Terms of Use'.

[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 dashboard with the 'Key Pairs' section selected. The left sidebar lists various EC2 services: EC2 Dashboard, Events, Tags, Reports, Limits, Instances, AMIs, Elastic Block Store, Network & Security, Load Balancing, and a feedback link. The 'Key Pairs' link is highlighted with an orange border. The main content area displays a message: 'You do not have any Key Pairs in this region.' Below it is a sub-instruction: 'Click the "Create Key Pair" button to create your first Key Pair.' A prominent blue 'Create Key Pair' button is centered below the message. At the bottom of the page, there are links for Feedback, English (US), and a copyright notice: '© 2008 - 2019, Amazon Web Services, Inc. or its affiliates. All rights reserved. Privacy Policy 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 Key Pairs page. The left sidebar lists various EC2 services: EC2 Dashboard, Events, Tags, Reports, Limits, Instances (Instances, Launch Templates, Spot Requests, Reserved Instances, Dedicated Hosts, Capacity Reservations), Images (AMIs, Bundle Tasks), Elastic Block Store (Volumes, Snapshots, Lifecycle Manager), Network & Security (Security Groups, Elastic IPs, Placement Groups, Key Pairs, Network Interfaces), and Load Balancing (Load Balancers, Target Groups). The 'Key Pairs' option is selected and highlighted in orange. The main content area displays a message: "You do not have any Key Pairs in this region. Click the 'Create Key Pair' button to create your first Key Pair." Below this is a blue 'Create Key Pair' button. A modal window titled 'Create Key Pair' is open in the center, containing a text input field with the placeholder 'Key pair name:' and the value 'Administrator1'. At the bottom of the modal are two buttons: 'Cancel' and 'Create'.

- View key pair details

The screenshot shows the AWS EC2 Key Pairs page. The left sidebar includes links for EC2 Dashboard, Events, Tags, Reports, Limits, Instances, AMIs, ELASTIC BLOCK STORE, NETWORK & SECURITY, Key Pairs (which is selected and highlighted in orange), and LOAD BALANCING. The main content area displays a table with one row for 'Administrator1'. The table columns are 'Key pair name' and 'Fingerprint'. The 'Key pair name' column contains 'Administrator1' and the 'Fingerprint' column contains 'fa:c0:90:4b:a7:bf:82:9f:3d:13:1c:73:dd:83:eb:b4:bd:74:f2:26'. Below the table, a section titled 'Key Pair: Administrator1' shows the same information. At the bottom of the page, there are links for Feedback, English (US), a download link for 'Administrator1.pem', and buttons for Show All and Close.

[Return to Table of Contents](#)

Creating Roles

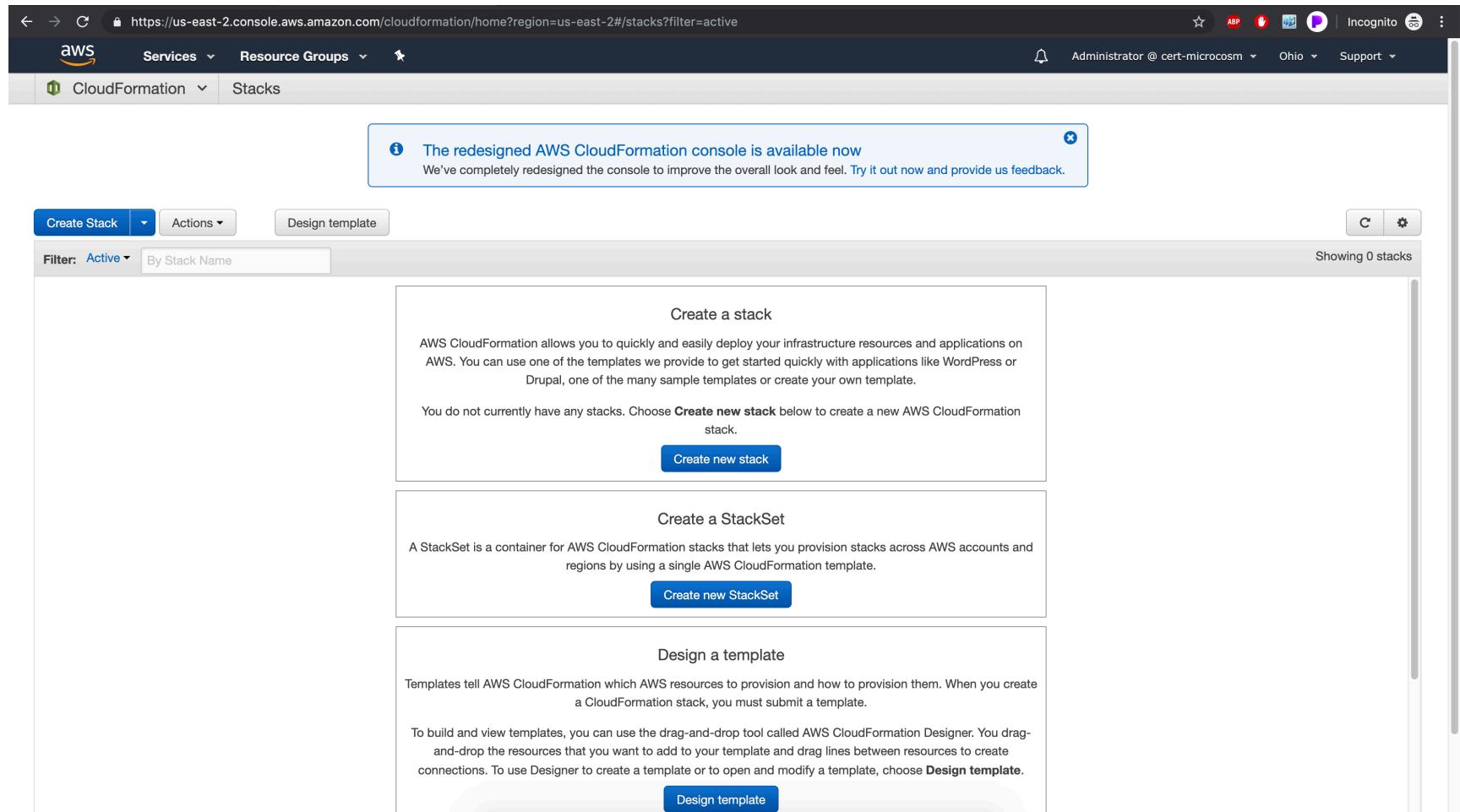
See the instructions in the [Create IAM Role for EC2 Based CodeDeploy](#) section.

[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

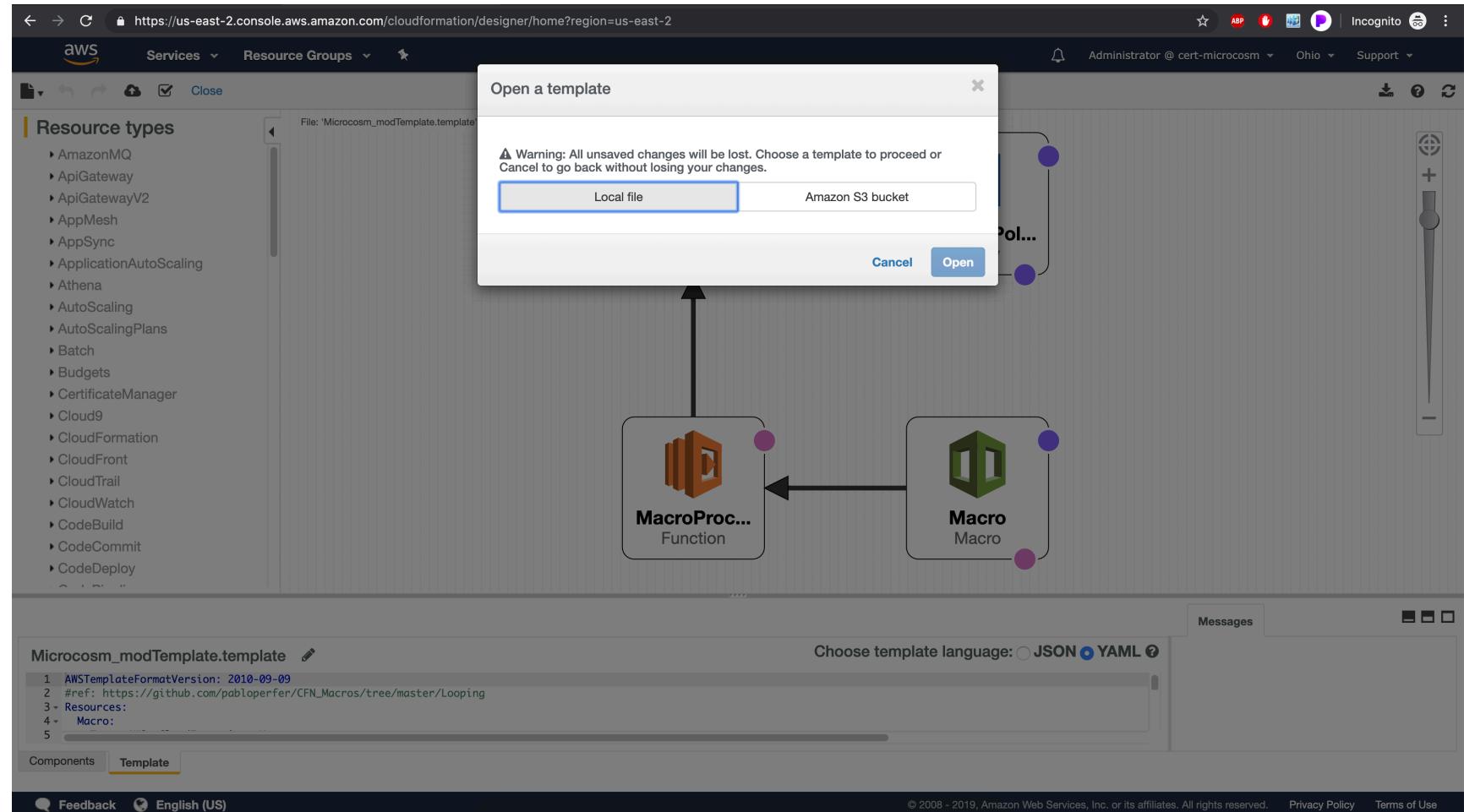
The screenshot shows the 'Create stack' wizard in the AWS CloudFormation console. The current step is 'Select Template'. On the left, a sidebar lists steps: 'Select Template' (which is active and highlighted in orange), 'Specify Details', 'Options', and 'Review'. The main content area is titled 'Select Template' and contains instructions: '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.' Below this, there are two sections: 'Design a template' (using AWS CloudFormation Designer) and 'Choose a template' (using a JSON/YAML template file). Under 'Choose a template', the 'Select a sample template' option is selected, and a dropdown menu is open, showing a list of available templates. Other options include 'Upload a template to Amazon S3' (with a 'Choose File' button) and 'Specify an Amazon S3 template URL'. At the bottom right of the content area are 'Cancel' and 'Next' buttons. The footer of the page includes links for 'Feedback', 'English (US)', 'Privacy Policy', and 'Terms of Use'.

- 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)

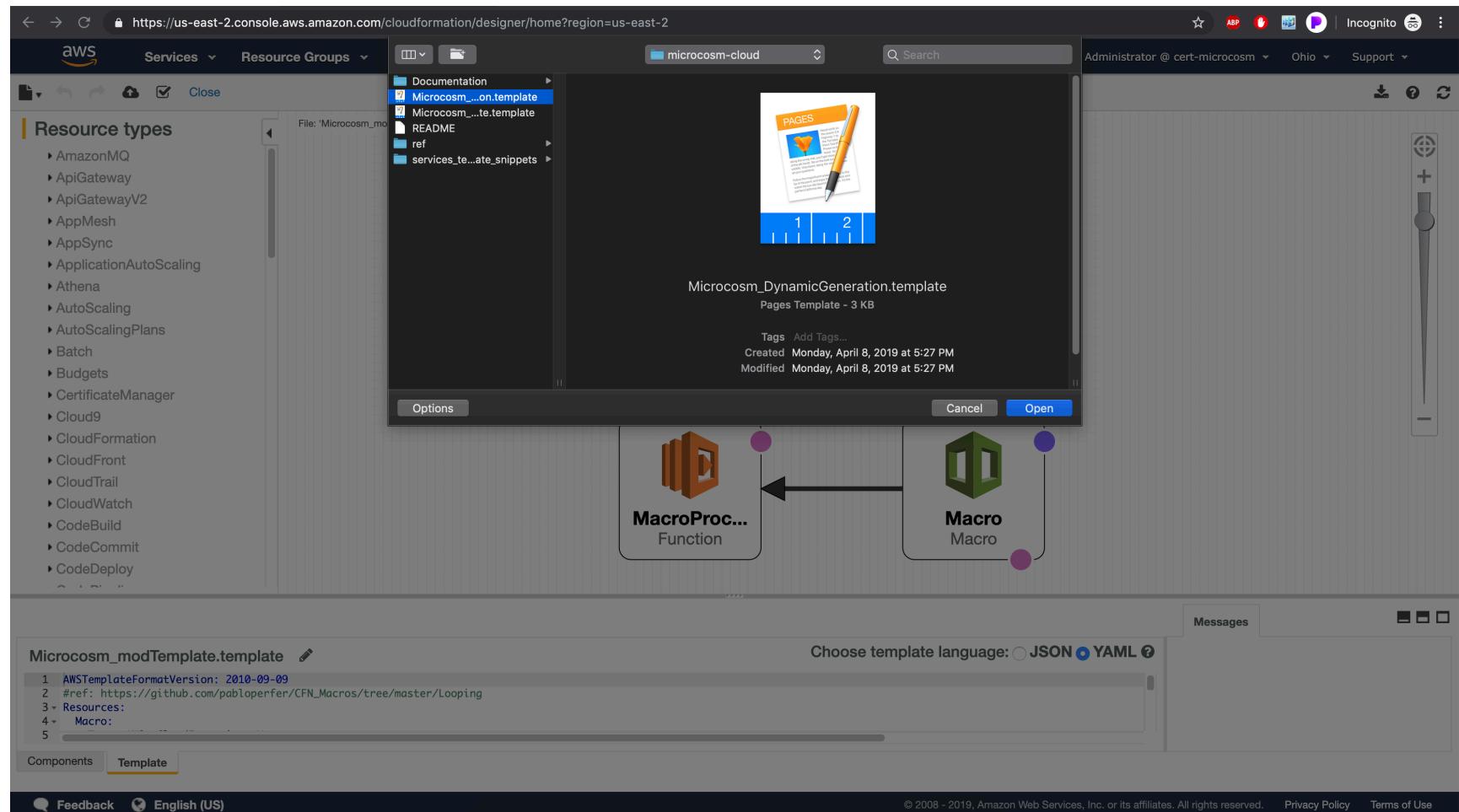
The screenshot shows the AWS CloudFormation Designer interface. On the left, a sidebar lists various AWS services with 'CloudFormation' selected. The main area displays a diagram of a CloudFormation stack. At the top is a 'LambdaExecutionRole' (represented by a hard hat icon) which has a blue circular handle at the bottom right. To its right is a 'LambdaPolicy' (represented by a document icon) which also has a blue circular handle. Below them is a 'MacroProcessorFunction' (represented by a server icon) and a 'Macro' (represented by a green cube icon). Arrows point from the policy and macro to the execution role, indicating dependencies. The bottom half of the screen shows the raw CloudFormation template code in YAML format:

```
AWSTemplateFormatVersion: 2010-09-09
#ref: https://github.com/pablopfer/CFN_Macros/tree/master/Looping
Resources:
  Macro:
```

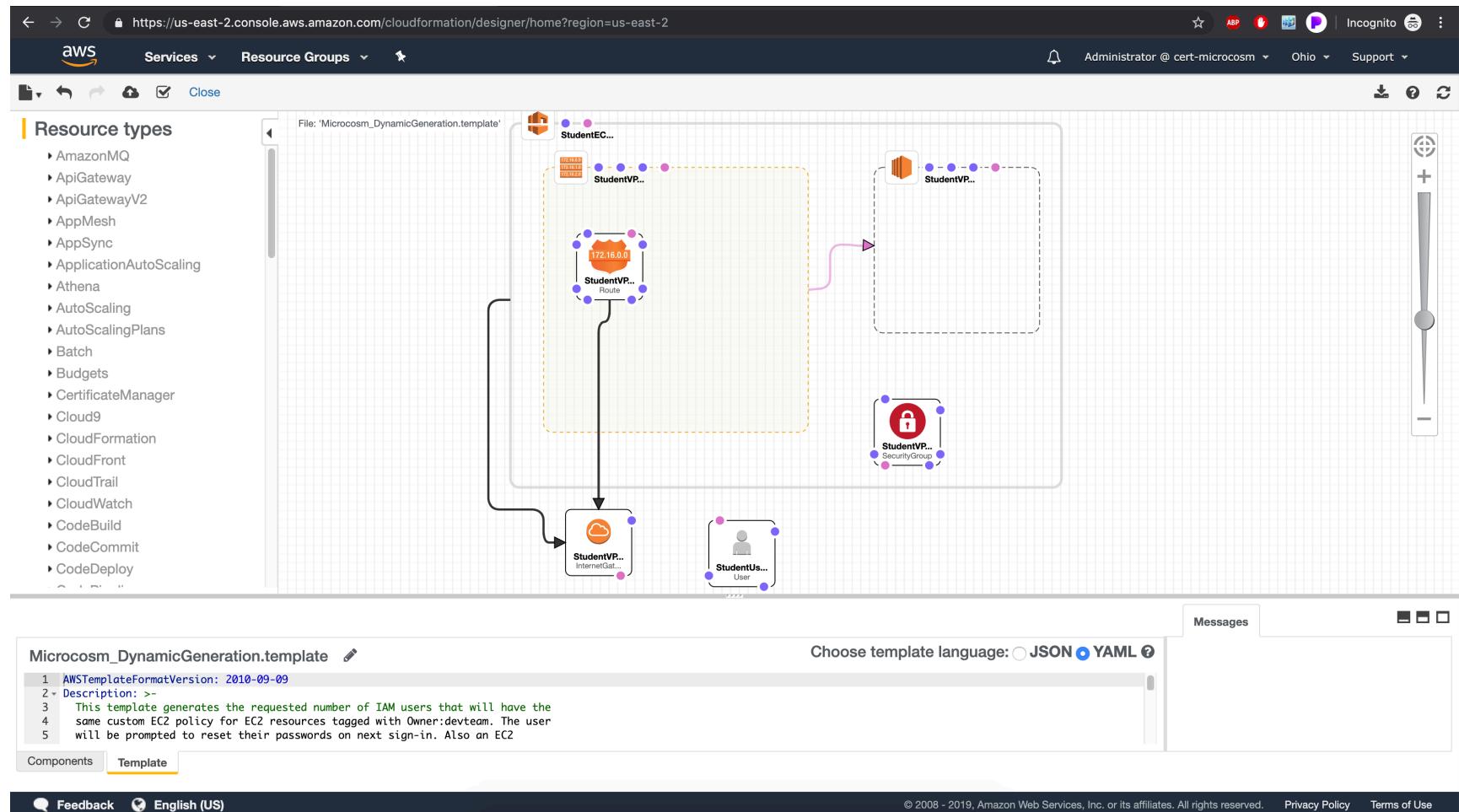
- Select the source of your template. It is good practice to name your templates with the extension `.template` or some variation thereof



- 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).



- 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 current step is 'Select Template'. The left sidebar has tabs for 'Select Template' (which is active), 'Specify Details', 'Options', and 'Review'. The main content area is titled 'Select Template' and contains the following information:

- 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.](#)

Under 'Choose a template', there are three options:

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

At the bottom right of the main form are 'Cancel' and 'Next' buttons. The footer of the page includes links for Feedback, English (US), Copyright notice (© 2008 - 2019, Amazon Web Services, Inc. or its affiliates. All rights reserved.), Privacy Policy, and Terms of Use.

- **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

The screenshot shows the AWS CloudFormation 'Create stack' wizard. The current step is 'Specify Details'. The 'Stack name' field contains the value 'Macro'. At the bottom right of the screen, there are three buttons: 'Cancel', 'Previous', and a blue 'Next' button.

- Press Next

The screenshot shows the AWS CloudFormation 'Create New Stack' wizard. The current step is 'Permissions'. It includes fields for selecting an IAM role ('Choose a role (optional)') or entering a role ARN. Below this is the 'Rollback Triggers' section, which is currently collapsed. It contains a 'Monitoring Time' input set to 0-180 minutes, with a note about the range. A table lists one trigger: 'AWS::CloudWatch::Alarm' with an empty ARN field and a '+' button to add more. The 'Advanced' section is expanded, showing options for notifications and stack policies, with a note about learning more. At the bottom are standard navigation buttons: 'Cancel', 'Previous', and '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

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 Minutes

Minimum value of 0. Maximum value of 180.

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

► Advanced

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

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

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

- Select the acknowledgement and press Create

The screenshot shows the AWS CloudFormation 'Create Stack' wizard. The 'Details' tab is selected, displaying the following information:

- Stack name:** Macro
- Tags:** No tags provided.
- Rollback Triggers:** No monitoring time provided, No rollback triggers provided.
- Advanced:**
 - Notification:** None
 - Termination Protection:** Disabled
 - Timeout:** none
 - Rollback on failure:** Yes

Capabilities:

Info: 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

The screenshot shows the AWS CloudFormation console interface. At the top, there's a navigation bar with the AWS logo, 'Services', 'Resource Groups', and a dropdown for the current user ('Administrator @ cert-microcosm'). Below the navigation is a header bar with 'CloudFormation' and 'Stacks'. A sub-header bar includes 'Create Stack', 'Actions', and 'Design template' buttons. A filter dropdown set to 'Active' and a search input field 'By Stack Name' are also present. The main content area displays a table titled 'Showing 1 stack' with one row for the stack 'Macro'. The table columns are 'Stack Name', 'Created Time', 'Status', 'Drift Status', and 'Description'. The 'Status' column shows 'CREATE_COMPLETE' for the 'Macro' stack. Below this, there's a tab-based navigation bar with 'Overview', 'Outputs', 'Resources', 'Events' (which is selected), 'Template', 'Parameters', 'Tags', 'Stack Policy', 'Change Sets', and 'Rollback Triggers'. Under the 'Events' tab, a table lists various creation events for the 'Macro' stack on April 11, 2019, at 10:23 UTC. The table columns are 'Timestamp', 'Status', 'Type', 'Logical ID', and 'Status Reason'. Events include 'CREATE_COMPLETE' for the stack itself and its components like Lambda functions and IAM policies.

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

- Repeat the Stack creation process for the Microcosm Dynamic Template

The screenshot shows the 'Create stack' wizard in the AWS CloudFormation console. The current step is 'Select Template'. The left sidebar shows navigation steps: 'Select Template' (highlighted), 'Specify Details', 'Options', and 'Review'. The main content area has two sections: 'Design a template' (using AWS CloudFormation Designer) and 'Choose a template' (using a JSON/YAML template file). Under 'Choose a template', the 'Upload a template to Amazon S3' option is selected, with a file named 'Microcosm_Dy...on.template' chosen. Below it, the 'Specify an Amazon S3 template URL' option is available. At the bottom right of the main area are 'Cancel' and 'Next' buttons. The footer includes links for 'Feedback', 'English (US)', and legal notices.

- 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 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.

Create stack

Select Template

Specify Details

Options

Review

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 WebServer EC2 instance type

NumberOfUsers Enter the number of users to create for the Dev Team

SSHLocation Lockdown SSH access to the bastion host (default can be accessed from anywhere)

Cancel Previous Next

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

- Press Next

The screenshot shows the AWS CloudFormation 'Create New Stack' wizard, Step 1: Set Stack Options. The URL is <https://us-east-2.console.aws.amazon.com/cloudformation/home?region=us-east-2#/stacks/new>.

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.

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

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

The screenshot shows the AWS CloudFormation 'Create New Stack' wizard. In the 'Advanced' section, there are three settings: 'Termination Protection' set to 'Disabled', 'Timeout' set to 'none', and 'Rollback on failure' set to 'Yes'. Below this is the 'Capabilities' section, which contains a note about transforms requiring IAM access and two checkboxes for acknowledging IAM resource creation. The 'Transforms' section notes that the stack includes a transform and provides a 'Create Change Set' button. At the bottom, there are links for Feedback, English (US), and legal notices.

No rollback triggers provided

Advanced

Notification

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

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

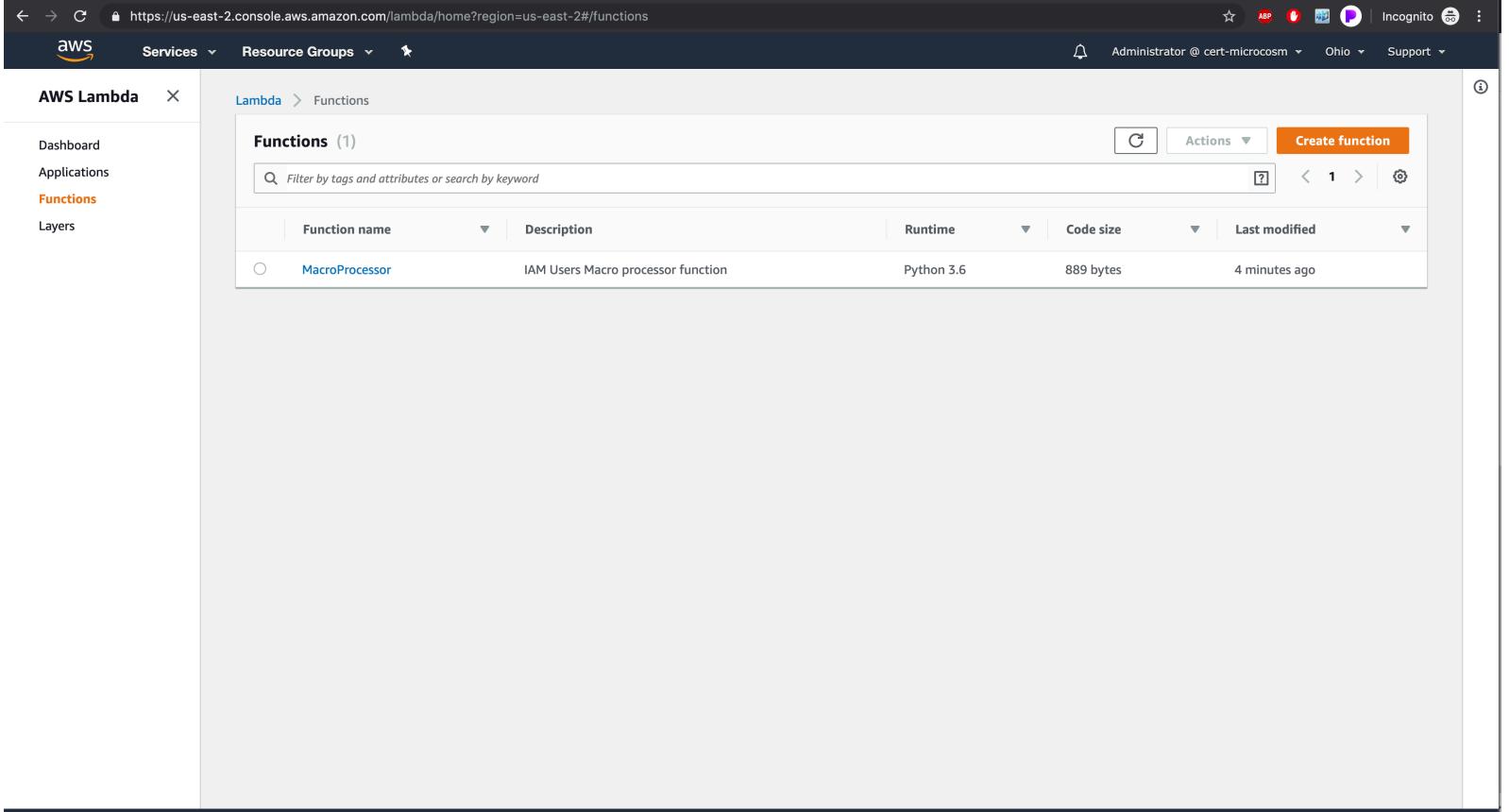
The screenshot shows the AWS CloudFormation 'Create New Stack' wizard at the 'Transforms' step. At the top, there are two checked checkboxes: 'I acknowledge that AWS CloudFormation might create IAM resources.' and 'I acknowledge that AWS CloudFormation might create IAM resources with custom names.' Below this, a message box says: 'Check the following transforms: ["443007076818::Macro"]'. It explains that a change set must be created because the stack includes transforms, and provides a link to 'View change set details'. The main 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::VPCCGatewayAttachment	
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 table area, there are three buttons: 'Cancel', 'Previous', and a blue 'Execute' button.

At the very bottom of the page, there are links for 'Feedback', 'English (US)', '© 2008 - 2019, Amazon Web Services, Inc. or its affiliates. All rights reserved.', 'Privacy Policy', and 'Terms of Use'.

- 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

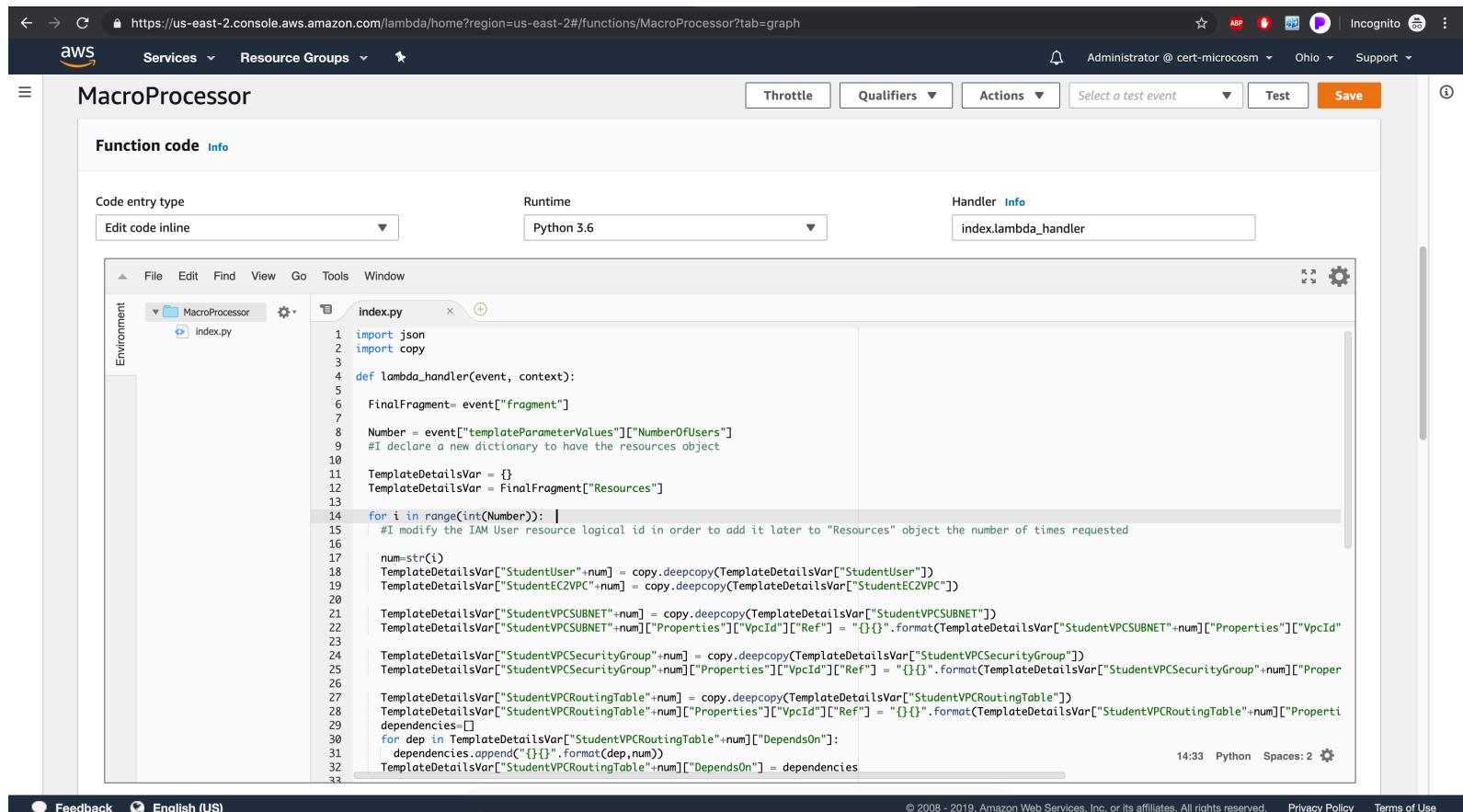


The screenshot shows the AWS Lambda Functions page. The left sidebar has 'AWS Lambda' selected under 'Functions'. The main area shows a table with one row:

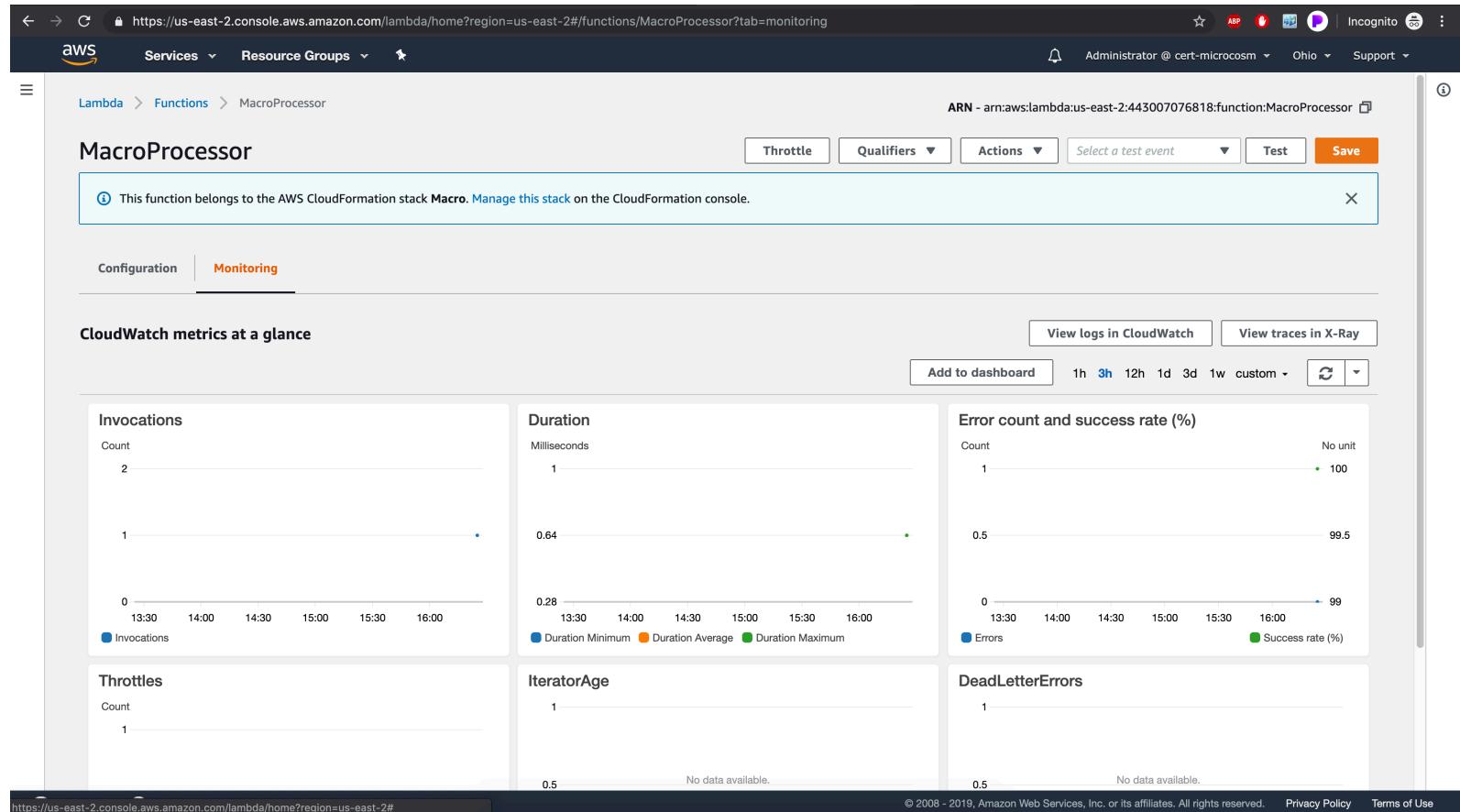
Function name	Description	Runtime	Code size	Last modified
MacroProcessor	IAM Users Macro processor function	Python 3.6	889 bytes	4 minutes ago

At the bottom, there are links 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)



- 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 Log Groups interface. The left sidebar navigation bar includes links for CloudWatch, Dashboards, Alarms, Billing, Events, Rules, Event Buses, Logs (which is selected), Insights, Metrics, and Favorites. The main content area displays a list of log streams under the path 'CloudWatch > Log Groups > Streams for /aws/lambda/MacroProcessor'. A search bar at the top allows filtering by 'Log Stream Name Prefix'. The list is sorted by 'Last Event Time'. Each log stream entry contains the prefix, a timestamp, and a small checkbox icon.

Log Stream Prefix	Last Event Time
2019/04/11[\$LATEST]c94293df5f8c42eea323d4e9e2a85f5e	2019-04-11 10:26 UTC-6
2019/04/09[\$LATEST]b725f8d2f2504450850af589cda4e586	2019-04-09 12:44 UTC-6
2019/04/09[\$LATEST]319574b1043d4b67b2249689e9bff710	2019-04-09 11:53 UTC-6
2019/04/08[\$LATEST]7f138ede3c3a7419a91e803e7a8ac7e3	2019-04-08 17:07 UTC-6
2019/04/08[\$LATEST]b4889a7392764dfa82369469a1e6c6ba	2019-04-08 17:03 UTC-6
2019/04/08[\$LATEST]1408f685438a420c93ced1138d841816	2019-04-08 16:55 UTC-6
2019/04/08[\$LATEST]484dd5b31ce143a98a5b168670e233ac	2019-04-08 16:46 UTC-6
2019/04/08[\$LATEST]da47eff4e1c04832a4fbfa7754594f6	2019-04-08 16:22 UTC-6
2019/04/08[\$LATEST]cd4d20fa0a404028a8cb48ef292a4fad	2019-04-08 16:12 UTC-6
2019/04/08[\$LATEST]9114ef93b1c4b4fb6d6d30a7a070eca7	2019-04-08 16:10 UTC-6
2019/04/08[\$LATEST]b0951a37d71348259459a6c6d9236113	2019-04-08 16:09 UTC-6
2019/04/08[\$LATEST]247811ab40d240869286e1c427c288c1	2019-04-08 16:07 UTC-6
2019/04/08[\$LATEST]c440b9c0602d43fca1f81746a1355945	2019-04-08 16:04 UTC-6
2019/04/08[\$LATEST]fe4621a210b046d8878178553167b624	2019-04-08 15:59 UTC-6
2019/04/08[\$LATEST]3266799eed464cabbd5704b7bc8c8356f	2019-04-08 15:50 UTC-6
2019/04/08[\$LATEST]3709987b856841f4ab3c3c076e9d5765	2019-04-08 15:48 UTC-6
2019/04/08[\$LATEST]70bbb01f65334d1d8d8ead362f1bc1c8	2019-04-08 15:34 UTC-6
2019/04/08[\$LATEST]je0bb102c82744946b0bc7d34d39c085a	2019-04-08 15:22 UTC-6
2019/04/08[\$LATEST]00f84282ad224295af4f30a902db6bd2	2019-04-08 15:02 UTC-6
2019/04/08[\$LATEST]42d5bf2944ef44c2bfc54db955a2162f	2019-04-08 14:47 UTC-6
2019/04/08[\$LATEST]ce50a33bd9bb4cb6917e9d054f6e7172	2019-04-08 14:28 UTC-6
2019/04/08[\$LATEST]4be3e1e5cab429083bf85e17116740a	2019-04-08 14:13 UTC-6
2019/04/08[\$LATEST]bd9ffc3cf59345dc8103e33350555777	2019-04-08 14:11 UTC-6
2019/04/08[\$LATEST]af60fbdd7f5f467994edb335ce664204	2019-04-08 13:29 UTC-6

- View the logs generated by the lambda function

The screenshot shows the AWS CloudWatch Log Event Viewer interface. The left sidebar navigation bar includes CloudWatch, Dashboards, Alarms, ALARM (0), INSUFFICIENT (0), OK (0), Billing, Events, Rules, Event Buses, Logs (selected), Insights, Metrics, and Favorites. The main content area displays log entries for the path /aws/lambda/MacroProcessor on 2019/04/11. The logs show the creation of an IAM user named 'StudentUser' with specific properties like Path, LoginProfile, and PasswordResetRequired. The logs also show the execution duration and memory usage.

Time (UTC +00:00)	Message
2019-04-11	No older events found at the moment. Retry .
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": {"AWS::TemplateFormatVersion": "2010-09-09", "Description": "This template generates the requested number of IAM users tha", "Properties": {"Path": "/", "LoginProfile": {"Password": "\$bzq\$UrFLxw9HFtB-49eRtf!", "PasswordResetRequired": true}, "Metadata": {"AWS::CloudForma", "StudentUser": {"Type": "AWS::IAM::User", "Properties": {"Path": "/", "LoginProfile": {"Password": "\$bzq\$UrFLxw9HFtB-49eRtf!", "PasswordResetRequired": true}}}, "Metadata": {"AWS::CloudForma", "END RequestId: f40b8d79-75d1-4182-a8eb-aed854299233 Duration: 0.64 ms Billed Duration: 100 ms Memory Size: 128 MB Max Memory Used: 40 MB
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

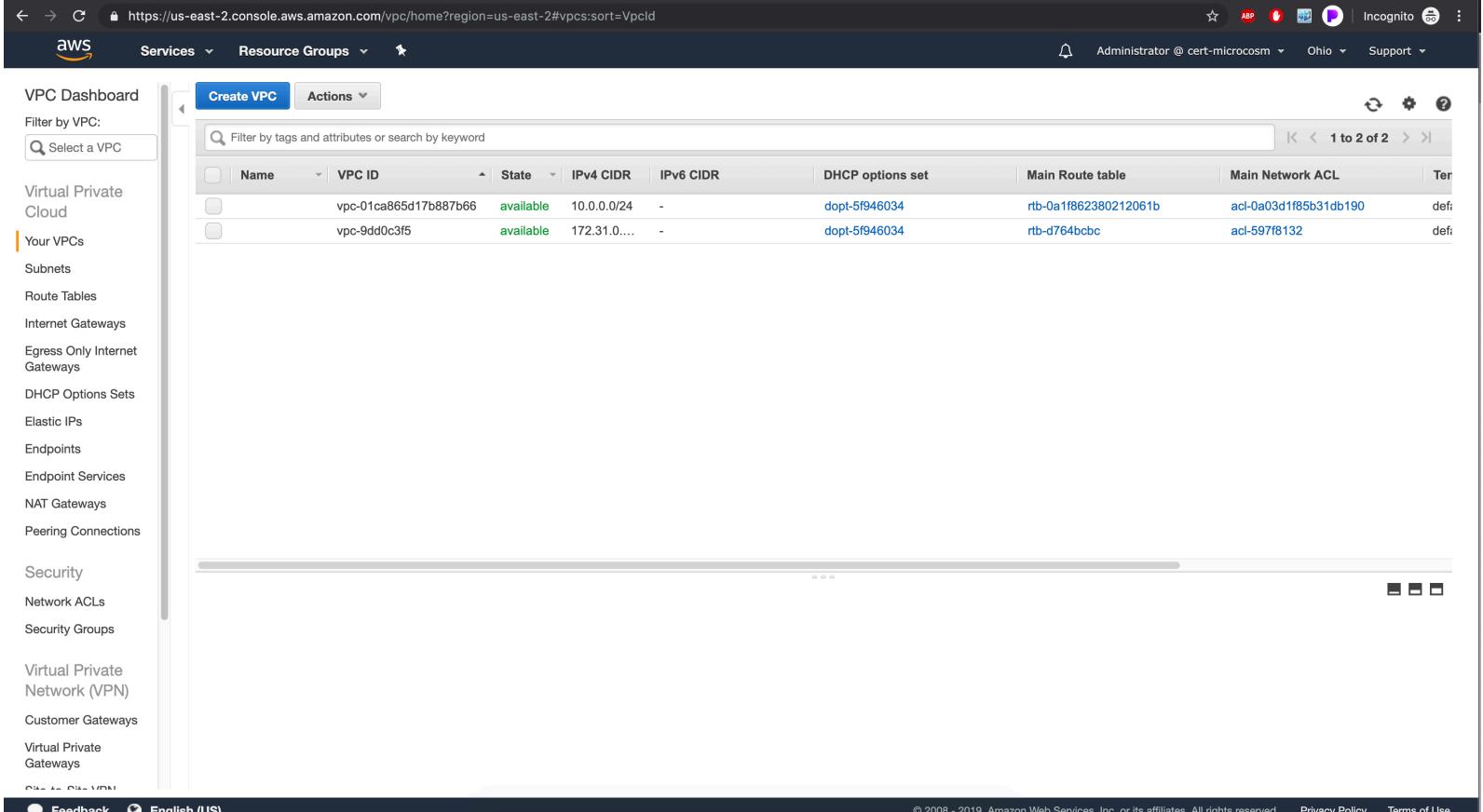
The screenshot shows the AWS IAM Users page. The left sidebar includes options like Dashboard, Groups, Roles, Policies, Identity providers, Account settings, Credential report, and Encryption keys. The main area displays a table of users with columns: User name, Groups, Access key age, Password age, Last activity, and MFA. The table shows three results:

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

A red box highlights the word "Text" in the bottom right corner of the main content area.

- **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. On the left, a sidebar lists various VPC components: Virtual Private Cloud, Your VPCs (selected), 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. At the top, there are buttons for 'Create VPC' and 'Actions'. A search bar at the top right allows filtering by tags and attributes or searching by keyword. The main area displays a table of VPCs with the following data:

Name	VPC ID	State	IPv4 CIDR	IPv6 CIDR	DHCP options set	Main Route table	Main Network ACL	Termination
vpc-01ca865d17b887b66	available	10.0.0.0/24	-	doptl-5f946034	rtb-0a1f862380212061b	acl-0a03d1f85b31db190	def	
vpc-9dd0c3f5	available	172.31.0....	-	doptl-5f946034	rtb-d764bcfc	acl-597fb132	def	

At the bottom, there are links for 'Feedback', 'English (US)', and copyright information: © 2008 - 2019, Amazon Web Services, Inc. or its affiliates. All rights reserved. Privacy Policy Terms of Use.

[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](#).

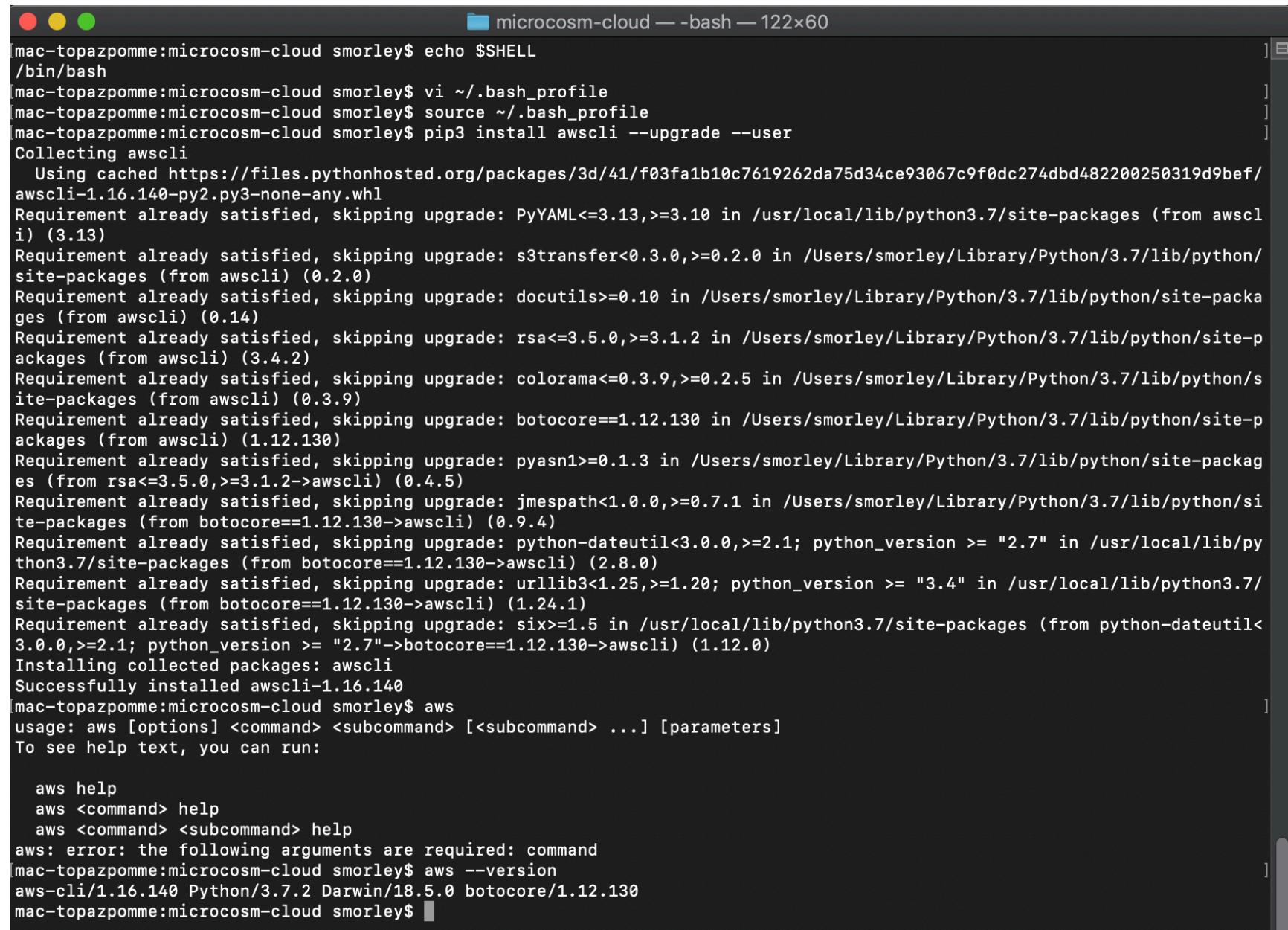
- Add the path to the python binaries to your `~/.bash_profile` in order to access the 'aws' after it is installed



microcosm-cloud — vi `~/.bash_profile` — 122x60

```
export PATH=$PATH:/Users/smorley/Library/Python/3.7/bin
```

- Install the AWS CLI



```
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: urllib3<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 eyJwYXlsb2FkIjoiKzhsTWTCL2krRjFoL3p1VzFVcjBlajNvbWhPUGpMWW50K01XNVJLUUpXR1pHTGFDQTMzQ2VBY3hYZ29XT0R
SS1YzKpIdGdvZ0KvaWxHzZ4Y2lEanJTQXFcaVZJemdXWVNScEx4Ymg3NDBvbW1JQTFHQs8wwWdmRnZJN1dBblFQa0pwS1Nza0hPZkR5N0VuSwtVYVBjWxOrZ
1EzWDNBc1cydnF5bjY0WDhaU014b3F2bkdoV09hKzM4bz1PNWdBVWtMwmVsMXhweWpBQncyNFBVDDlcldKeG5SbGt2enVrSXhmQjFhZ20zek5sMkZhRTFydnh
ibEw3NnI2ZGQvNzNGeTU4c3FRa1g1SG1MdFlGY0ozT1RmMkJKckQ1cj1UUkR6TG1xZ2ZMRF14eG1PMVVIdjAwaUovb2hEa1VtOfJMcDM3SGhabzJ1MnpjTkRVC
3gvbvNROW5EM2dvZ1NtL1YwUFhZcm1pdkkvNFJERERxYnA5eXZmcHVSazV2OUF6MGZucl3NnJUSS9IZHJCZzFxeVVnc3hnSmo0RXY5TuTa0WZzb0pvRF1XS1h
6N2ZndWpIRXM0Y3AyM2dUSEhodUjrN09FVkl1NVgvV3NIMEFMaHBhYWI2UXMzbUJVa1Rvd1pMd1M0Uk5zL01WMThXb0hRRk4vUU1iSW9qawxNRGxnMHFCQkorV
GxaYTNYOFVrckZMNE5BU01mdFNSL0NNc2hTSud3aV1KemZIWcYVDZZVWZiekowT3VtNDRPWTY3RDRzb3JaMTRuRWZ3bkJ1dUo5SUNPVEFyK0pJTms0aEVDUuF
JQ2x3SFU1S0p5REZOYX1yRFEzeGQ5cU5GUEJxR314bURxeFYwd0o0MkorUy8vSE1DTTc3QWERl0UzNFBNaEt1N2E5a3k5YzVPYXQzUGNxeUFEUKJaaFYzYzQyd
TRkQ1VQUwtDZUuyUDV4WEFLUnJNOUZ4RFJwT315by9IUXVHNG9iWEQxZ0htUVhnRXc30Uh1WnN5aEVXdnp3bnVHaC9tajVUUU1SVjNRL1Yzbw9LMGZ0VW9HWm9
FWmIxRHRjUjBHSU9SUTZoZGMzWGxrZS9zNU9uRF11SzJvSkk3bE1yVzdCM09jMXdoNG1oUhpVNXNpYUhjK0loNFpOb252Z1VrMTN40GxMRRWeHFLN0I1Q1g5N
zdDeERvTjdnbksvZS9Dd1JHVVhHS0NIOUsxbW1Lb1VSL1R5V1Zzek1MVHU30Tk5MXBGaU5CK2pmeVVQU0o4ZXVmNkQwaHc2cmw5bkhkbW1BYUZhNHFuUW55bUQ
xbmcxUWVwa0liiUUh6aURZZm5tNFZGOWNmaVZSY21DWGJXM0FMWPaZXBjUFJWM0FKcW9VUE1kd3NTd2NyChNsZHBCU0x5YkJ1UVdvRmZxanRvT2YzWGpObzVke
G8zU3RZbmxoSkVpMu9yUGkxTVh2bUFLciszMkd4an1CMWY3K29DMzRil3orVkuwVUdFNmJTR2ZFR0pTVkxMckJ3PT0iLCJkYXRha2V5IjoiQVFFQkFIakI3L21
nd01nNE5Qd2F1cnhTSV14NEhmbnh1R2MvNDhiRHd2d0RwT11XWmdBQUFINhdmQV1KS29aSwH2Y05BUWNhb0c4d2JRSUJBREJvQmdrcWhraUc5dzBCQndFd0hnW
UpZSVpJQVdVREJBRXVNQkVFRExRYKViQzNwY0htR3c2bk5BSUJFSUE3eUUveWI1Mkp00X15MnMzeTRUd3JiNm9nRGV0d1ArNCswOVIwcm1kM0dPL11vai96TU9
BNEpiNVprUHM4QWFqWnRNM09vL1IxY1FLNmJsdz0iLCJ2ZXJzaW9uIjoiMiIsInR5cGUi0iJEQVRBX0tFWSIsImV4cGlyYXRpb24i0jE1NTQ5NjQ40DF9 http
s://443007076818.dkr.ecr.us-east-2.amazonaws.com
```

- Docker login using received credentials

```
mac-topazpomme:devops-microcosm smorley$ docker login -u AWS -p eyJwYXlsb2FkIjoiKzsTWTCL2krRjFoL3p1VzFVcjBlajNvbWhPUGpMWW50K01XNVJLUUpXR1pHTGFDQTMzQ2VBY3hYZ29XT0RSS1YzYkpIdGdvZ0kvaWxHZZ4Y21EanJTQXFcaVZJemdxWVNSeEx4Ymg3NDBvbW1JQTFHQs8wWWdmRnZJN1dBblFQa0pwS1Nza0hPZkR5N0VuSwtVYVBJWXorZ1EzWDNBc1cydnF5bjY0WDHaU014b3F2bkdoV09hKzM4bz1PNWdBVWtMWmVsMXhweWpBQncyNFBXVDdlcl dKeG5SbGt2enVrSXhmQjFhZ20zek5sMkZhRTFydnhibEw3NnI2ZGQvNzNGeTU4c3FRa1g1SG1MdFlGY0eZt1RmMkJKckQ1cjlUUkR6TG1xZ2ZMRf1eG1PMVVI djAwaUovb2hEa1Vt0FJMcDM3SGhabzJ1MnpjTkRvc3gvbVNROW5EM2dvZ1NtL1YwUFhzcm1pdkkvNFJERERxYnA5eXZmcHVSazV20UF6MGZucl3NnJUSS9IZHJCZzFxeVVnc3hnSmo0RXY5TUta0WZzb0pvRF1XS1h6N2ZndWpIRXM0Y3AyM2dUSEhodUJrN09FVkl1NVgvV3NIMEFMaHBhYWI2UXMzbUJVa1Rvd1pMd1M0Uk5zL01WMThXb0hRRk4vUU1iSw9qaWxNRGxnMHFCQkorVGxaYTNYOFVrckZMNE5BU01mdFNSL0NNc2hTSUD3aV1KemZIZWcYVDZZVWZiekowT3VtNDRPWTY3RDRzb3JaMTruRWZ3bkJ1dUo5SUNPVEFyK0pJTms0aEVDDnfJQ2x3SFU1S0p5REZOYX1yRFEzeGQ5cU5GUEJxR314bURxeFYwd0o0MkorUy8vSE1DTTc3QWErl0UzNFBNaEt1N2E5a3k5YzVPYXQzUGNxuUFEUkJaaFYzYzQydTRkQ1VQUWtDZUuyUDV4WEFLUnJNOUZ4RFJwT315by9IUXVHNG9iWEQxZ0htUVhnRXc30Uh1WnN5aEVXdn p3bnVHaC9tajVUUU1SVjNRL1Yzbw9LMGZ0VW9HWm9FWmIxRHRjUjBHSU9SUTz0ZGMzWGXrzS9zNU9uRF1lSzJvSkk3bE1yVzdCM09jMXdoNG1oUHpVNXNpYuhiK0loNFpOb252Z1VrMTN40GxMRVweHFLN0I1Q1g5NzdDeERvTjdnbksvZS9Dd1JHVVhHS0NI0UsxbW1Lb1VSL1R5V1Zzek1MVHU30Tk5MXBGaU5CK2pmeVVQU0 o4ZXVmNkQwaHc2cmw5bkhkbW1BYUZhNHFuUW55bUQxbmcxUWVwa01iUUh6aURZzm5tNFZGOWNmaVZSY21DWGJXM0FMWwpazXBjUFJWM0FKcw9VUE1kd3NTd2Ny cHnsZHBCU0x5YkJ1UVdvRmZxanRvT2YzWg0bzVkeG8zU3RZbxmoSkVpMU9yUGkxTvH2bUFLciszMkd4an1CMWY3K29DMzRiL3orVkuwVudFnMJTR2ZFR0pTVk xMckJ3PT0iLCJkYXRha2V5IjoiQVFQkFIakI3L21nd01nNE5Qd2F1cnhTSV14NEhmbnh1R2MvNdhIRhd2d0RwT1lXWmdBQUFINHdmQV1KS29aSWh2Y05BUWNhb0c4d2JRSUJBREJvQmdrcWhraUc5dzBCQndFd0hnWUpZSvpJQVdVREJBRXVNQkvFRExRYkViQzNwY0htR3c2bk5BSUJFSUE3eUuveW1MkpoOX15MmMzeTRUd3JiNm9nRGV0d1ArNCswOViwcmlkM0dPL1lvai96TU9BNEpiNvprUHM4QWFqWnRNM09vL1IxY1FLNmJsdz0iLCJ2ZXJzaW9uIjoiMiIsInR5cGUi0iJEQVRBX0tFWSIsImV4cGlyYXRpb24i0jE1NTQ5NjQ40DF9 https://443007076818.dkr.ecr.us-east-2.amazonaws.com
WARNING! Using --password via the CLI is insecure. Use --password-stdin.
Login Succeeded
[mac-topazpomme:devops-microcosm smorley$ docker container ls
CONTAINER ID        IMAGE               COMMAND             CREATED            STATUS              PORTS               NAMES
MES
[mac-topazpomme:devops-microcosm smorley$ docker container ls --all
CONTAINER ID        IMAGE               COMMAND             CREATED            STATUS              PORTS               NAMES
      PORTS
      NAMES
b57f36821ce5      sonarqube:lts      "./bin/run.sh"       7 days ago        Exited (0) 42 minutes ago
          sonarqube
73a4061cd9f0      gillax/hubot-slack-jenkins   "./bin/hubot -a slack"  7 days ago        Exited (0) 37 minutes ago
          hubot
ba06c7de2e42      gitlab/gitlab-ce     "/assets/wrapper"    7 days ago        Exited (0) 37 minutes ago
          gitlab
3193c8f8470b      sonatype/nexus      "/bin/sh -c 'java ...'" 7 days ago        Exited (143) 37 minutes ago
          sonatype_nexus
013b8b77e566      h1kkkan/jenkins-docker:lts  "/sbin/tini -- /usr/..." 7 days ago        Exited (143) 37 minutes ago
          jenkins
3fd4643668ec      owasp/zap2docker-stable  "zap-webswing.sh"     7 days ago        Exited (0) 44 minutes ago
          owaspzap
[mac-topazpomme:devops-microcosm smorley$ docker tag sonarqube:lts sonarqube:lts0419
]
```

- 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 sonarqube
{
    "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
079758d1bbbb7: Pushed
e11f2ab9e2f4: Pushed
f7d12d471667: Pushed
f350d0146bb3: Pushed
e38df31d449c: Pushed
af5ae4841776: Pushed
b17cc31e431b: Pushed
12cb127eee44: Pushed
604829a174eb: Pushed
fb641a8b943: Pushed
lts: digest: sha256:8cbd208b264ab1404bce2ab16bcd6ba9a31a35e5b3e800c0d72295d719c087e8 size:
2839
```

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

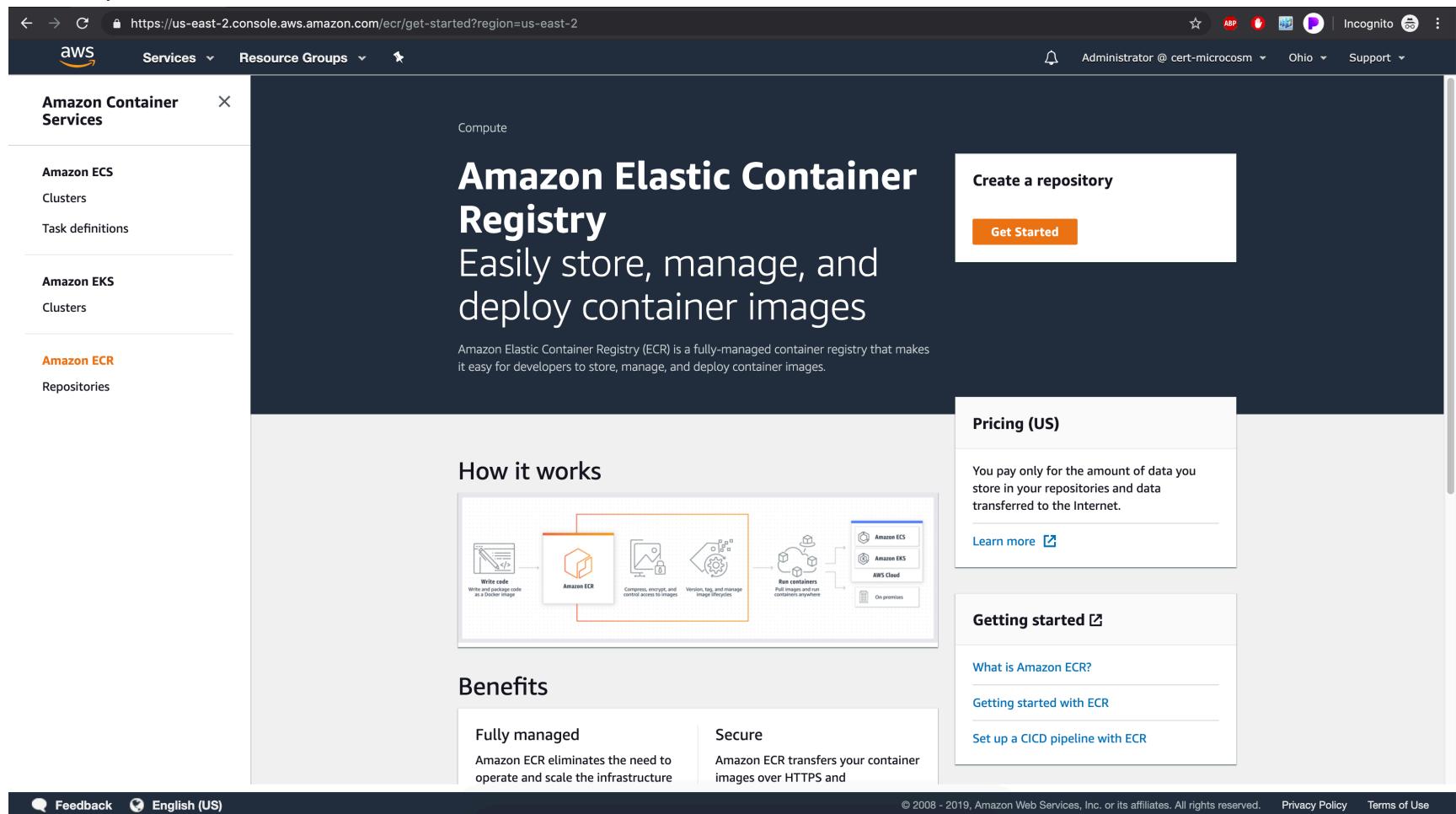
```
mac-topazpomme:devops-microcosm smorley$ aws ecr create-repository --repository-name h]
1kkan/jenkins-docker
{
```

```
"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 44300707] 6818.dkr.ecr.us-east-2.amazonaws.com/h1kkan/jenkins-docker:lts
[mac-topazpomme:devops-microcosm smorley$ docker push 443007076818.dkr.ecr.us-east-2.am] azonaws.com/h1kkan/jenkins-docker:lts
The push refers to repository [443007076818.dkr.ecr.us-east-2.amazonaws.com/h1kkan/jen
kins-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
50-cc641975: Pushed
```

```
f3a38968d075: Pushed  
a327787b3c73: Pushed  
5bb0785f2eee: Pushed  
lts: digest: sha256:a2a91ba9249a31a33ab9877bab42c7dea49955b30740c7c7beb6ffa124bc77ca size: 4925
```

- View repositories and their details in teh AWS ECR Console



The screenshot shows the AWS ECR Get Started page. The left sidebar has a navigation menu under 'Amazon Container Services' with options for Amazon ECS, Amazon EKS, and Amazon ECR. The 'Amazon ECR' section is currently selected, showing 'Repositories' as the sub-option. The main content area features a large title 'Amazon Elastic Container Registry' with the subtitle 'Easily store, manage, and deploy container images'. Below this, a description states: 'Amazon Elastic Container Registry (ECR) is a fully-managed container registry that makes it easy for developers to store, manage, and deploy container images.' A diagram titled 'How it works' illustrates the process: Write code and package code into a Docker image, which is then pushed to the Amazon ECR repository. From there, the image can be compressed, encrypted, and controlled via access keys. It can also be versioned, tagged, and managed. Finally, it can be pulled and run as containers anywhere, including Amazon ECS, Amazon EKS, AWS Cloud, or On premises. To the right, there are sections for 'Create a repository', 'Pricing (US)', and 'Getting started'. The 'Pricing (US)' section notes that users pay only for stored data and internet transferred data. The 'Getting started' section links to 'What is Amazon ECR?', 'Getting started with ECR', and 'Set up a CI/CD pipeline with ECR'. At the bottom, there are links for 'Feedback', 'English (US)', and legal notices.

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' (Clusters, Task definitions), 'Amazon EKS' (Clusters), and 'Amazon ECR' (Repositories). The 'Repositories' section is currently selected. The main content area is titled 'Repositories (6)' and contains a table with the following data:

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

At the top right of the main content area, there are buttons for 'View push commands', 'Delete', and 'Create repository'. Below the table, there are navigation controls (back, forward, search, etc.). The bottom of the page includes a URL bar with the address 'https://us-east-2.console.aws.amazon.com/ecr/repositories?region=us-east-2', a copyright notice '© 2008 - 2019, Amazon Web Services, Inc. or its affiliates. All rights reserved.', and links for 'Privacy Policy' and 'Terms of Use'.

The screenshot shows the AWS ECR console interface. On the left, there's a sidebar for 'Amazon Container Services' with sections for 'Amazon ECS' (Clusters, Task definitions), 'Amazon EKS' (Clusters), and 'Amazon ECR' (Repositories, Images, Permissions, Lifecycle Policy, Tags). The 'Images' section is currently selected. In the main content area, the repository 'gillax/hubot-slack-jenkins' is displayed. A table lists the image details:

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

At the bottom of the page, there are links for Feedback, English (US), and footer links for Privacy Policy and Terms of Use.

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

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

AWS Services Resource Groups

Administrator @ cert-microcosm Ohio Support

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.

How it works

The diagram illustrates the ECR workflow:

- Write code and package it as a Docker image.
- The image is uploaded to Amazon ECR.
- The image is compressed, encrypted, and its access is controlled.
- The image is versioned, tagged, and its lifecycle is managed.
- The image is pulled and run in containers, which can be run anywhere (AWS Cloud or On premises).

Benefits

- Fully managed
- Secure
- Launchpad

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.

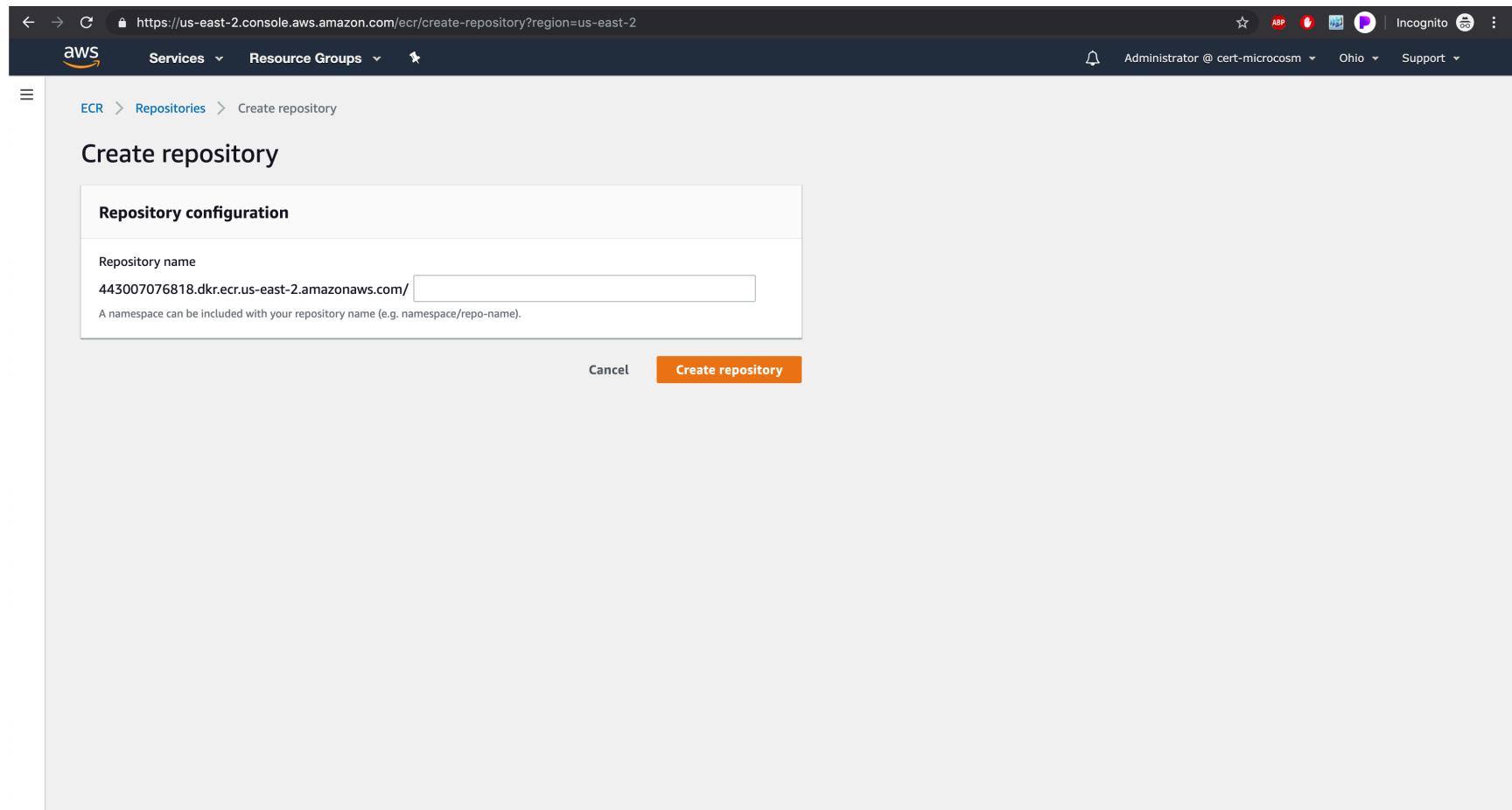
[Learn more](#)

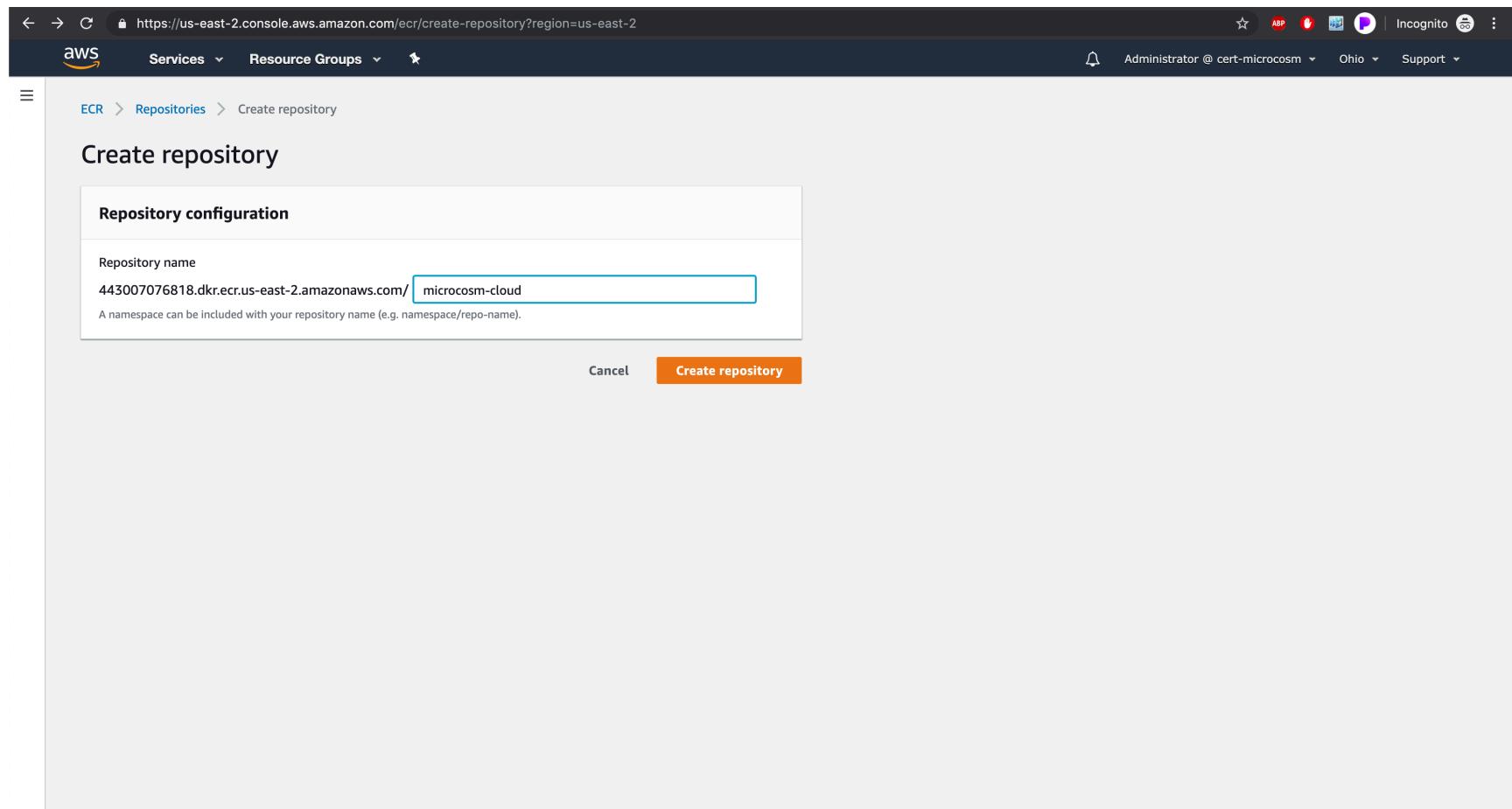
Getting started

[What is Amazon ECR?](#)

[Getting started with ECR](#)

[Set up a CI/CD pipeline with ECR](#)





The screenshot shows the AWS ECR (Amazon Container Registry) service in the AWS Management Console. The URL is https://us-east-2.console.aws.amazon.com/ecr/repositories?region=us-east-2. The left sidebar shows navigation for Amazon Container Services, Amazon ECS, Amazon EKS, and Amazon ECR. Under Amazon ECR, 'Repositories' is selected. A green success message at the top right says 'Successfully created repository'. Below it, the 'Repositories' section displays one item: 'microcosm-cloud' with URI 443007076818.dkr.ecr.us-east-2.amazonaws.com/microcosm-cloud, created on 04/10/19, 11:49:49 AM. There are buttons for '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

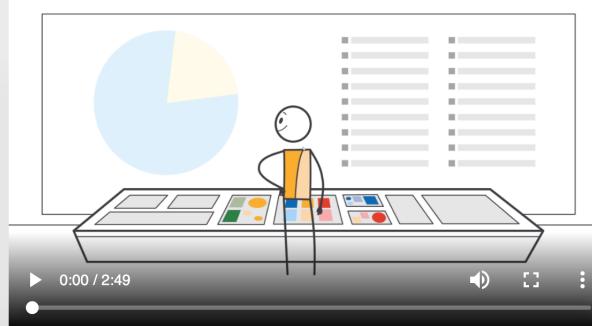
https://us-east-2.console.aws.amazon.com/ecs/home?region=us-east-2#/getStarted

AWS Services Resource Groups

Administrator @ cert-microcosm Ohio Support

Amazon ECS Clusters Task Definitions Amazon EKS Clusters Amazon ECR Repositories AWS Marketplace Discover software Subscriptions

Amazon Elastic Container Service (ECS)



0:00 / 2:49

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.

[Get started](#)

[Learn more about Amazon ECS](#)

 Run containers at scale

 Flexible container placement

 Integrated and extensible

The screenshot shows the AWS ECS Clusters page. The left sidebar includes links for Amazon ECS, Clusters (which is selected), Task Definitions, Amazon EKS, Clusters, Amazon ECR, Repositories, AWS Marketplace, Discover software, and Subscriptions. The main content area has a heading 'Clusters' and a note about ARN and resource ID format. It features 'Create Cluster' and 'Get Started' buttons. The 'View' dropdown is set to 'list'. Below it, a message says 'No clusters found' with a 'Get Started' button. The URL in the address bar is <https://us-east-2.console.aws.amazon.com/ecs/home?region=us-east-2#/clusters>.

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

```
graph TD; Container[Container definition] --- Task[Task definition]; Task --- Service[Service]; Service --- Cluster[Cluster]
```

Container definition

Choose an image for your container below to get started quickly or define the container image to use.

Edit

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)

custom

Configure

Edit

The screenshot shows the AWS ECR (Amazon Container Registry) interface. On the left, a sidebar for 'Amazon Container Services' lists 'Amazon ECS', 'Clusters', 'Task definitions', 'Amazon EKS', 'Clusters', and 'Amazon ECR'. Under 'Amazon ECR', 'Repositories' is selected, and 'Images' is highlighted. The main content area displays the 'h1kkan/jenkins-docker' repository. A search bar at the top says 'Find Images'. Below it, a table shows one image entry:

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

At the bottom, there are links for 'Feedback', 'English (US)', and legal notices: '© 2008 - 2019, Amazon Web Services, Inc. or its affiliates. All rights reserved.', 'Privacy Policy', and 'Terms of Use'.

The screenshot shows the AWS Elastic Container Service (ECS) console. On the left, there's a sidebar with navigation steps: Step 1: Container and Task, Step 2: Service, Step 3: Cluster, and Step 4: Review. Below this is a 'Diagram of ECS objects' and a 'Container definition' section listing two sample applications: 'sample-app' and 'tomcat-webserver'. The 'Task definition' section is currently active.

The main area is titled 'Edit container' under 'Standard' settings. It includes fields for 'Container name*' (set to 'jenkins'), 'Image*' (set to '443007076818.dkr.ecr.us-east-2.amazonaws.com/h1kkan/jenkins-docker:1ts'), and 'Memory Limits (MiB)' (set to 'Soft limit 128'). There's also a checkbox for 'Private repository authentication*' which is unchecked. A note below the memory limits says: 'Define hard and/or soft memory limits in MiB for your container. Hard and soft limits correspond to the `memory` and `memoryReservation` parameters, respectively, in task definitions. ECS recommends 300-500 MiB as a starting point for web applications.'

The 'Port mappings' section shows one mapping: 'Container port 8080' and 'Protocol tcp'. There's a link to 'Add port mapping'. A note in a box states: 'Host port mappings are not valid when the network mode for a task definition is host or aws:vcpc. To specify different host and container port mappings, choose the Bridge network mode.'

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

Getting Started with Amazon Elastic Container Service

Step 1: Container and Task

Step 2: Service

Step 3: Cluster

Step 4: Review

Diagram of ECS objects

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

Edit container

STORAGE AND LOGGING

Read only root file system

Mount points

Source volume: <none>

Container path: /var/jenkins_home

Read only

Add mount point

Volumes from

Source container:

Add volumes

Log configuration

Auto-configure CloudWatch Logs

Log driver: awslogs

Log options

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

Add key Add value

* Required

Cancel Update

The screenshot shows the AWS ECS Task Definition creation interface. At the top, there are four container definitions listed:

- 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)
- jenkins**: image: 443007076818.dkr.ecr.us-east-2.amazonaws.com/h1kkan/jenkins-docker:its, memory: 0.5GB (512), cpu: 0.25 vCPU (256). This container has a "Configure" button next to its name.

Task definition Edit

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 i

Network mode: awsvpc i

Task execution role: Create new i

Compatibilities: FARGATE i

Task memory: 0.5GB (512)

Task CPU: 0.25 vCPU (256)

*Required Cancel 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

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

The diagram illustrates the hierarchical structure of ECS objects. A dashed box encloses a 'Container definition' (represented by a yellow square containing a white rectangle) and a 'Task definition' (represented by a white rectangle). An arrow points from the 'Container definition' to the 'Service' object (represented by a black horizontal bar). Another arrow points from the 'Service' object to the 'Cluster' object (represented by a black horizontal bar at the bottom).

Define your service

Service name **jenkins-service** Edit

Number of desired tasks **1**

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

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

Step 1: Container and Task
Step 2: Service
Step 3: Cluster
Step 4: Review

Diagram of ECS objects and how they relate

```
graph LR; Container[Container definition] --- Task[Task definition]; Task --- Service[Service]; Service --- Cluster[Cluster]
```

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 Subnets

*Required Cancel Previous Next

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

The screenshot shows the 'Review' step of a task definition creation process in the AWS ECS console. The configuration includes:

- 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

At the bottom, there are buttons for *Required, Cancel, Previous, and Create.

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

Resource Type	Status
ECS resource creation	pending
Cluster microcosm	complete
Task definition first-run-task-definition:3	complete
Service	pending
Additional AWS service integrations	pending
Log group The log group [/ecs/first-run-task-definition] already exists	complete
CloudFormation stack	pending
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

Task	Status
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 \(US\)](#)

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

The screenshot shows the AWS ECS Service details page for a service named "jenkins-service" within a cluster called "microcosm". The service is currently active with a desired count of 1, a pending count of 1, and no running tasks. It uses a task definition named "first-run-task-definition:3" which is a replica type task using Fargate launch type and the latest platform version (1.3.0). The service role is "AWSServiceRoleForECS". The "Details" tab is selected, showing sections for Load Balancing (which is currently empty) and Network Access. Network access is configured to allow traffic from the VPC "vpc-0530acf15c751393" through subnets "subnet-0521ea54a64435dd4" and "subnet-0e50c4d05af5052a7", and security group "sg-06f07d8d65ef6b2fd". Auto-assign public IP is enabled. The page includes standard AWS navigation and footer links.

Clusters > microcosm > Service: jenkins-service

Service : jenkins-service

Cluster: microcosm

Status: ACTIVE

Task definition: first-run-task-definition:3

Service type: REPLICA

Launch type: FARGATE

Platform version: LATEST(1.3.0)

Service role: AWSServiceRoleForECS

Load Balancing

No load balancers

Network Access

Allowed VPC: vpc-0530acf15c751393

Allowed subnets: subnet-0521ea54a64435dd4, subnet-0e50c4d05af5052a7

Security groups*: sg-06f07d8d65ef6b2fd

Auto-assign public IP: ENABLED

The screenshot shows the AWS ECS Service Details page for the service `jenkins-service` within the cluster `microcosm`. The service is currently active with a desired count of 1, pending count of 0, and running count of 1. The service type is `REPLICA` and it uses the `FARGATE` launch type. The platform version is `LATEST(1.3.0)` and the service role is `AWSServiceRoleForECS`. The `Deployments` tab is selected, showing one deployment entry:

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

The screenshot shows the AWS ECS Service Details page for a service named "jenkins-service". The service is running in a cluster called "microcosm". Key details include:

- Cluster:** microcosm
- Status:** ACTIVE
- Task definition:** first-run-task-definition:3
- Service type:** REPLICA
- Launch type:** FARGATE
- Platform version:** LATEST(1.3.0)
- Service role:** AWSServiceRoleForECS

The "Tasks" tab is selected, showing one task in the "Running" state. The task details are:

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

Other tabs available include Details, Events, Auto Scaling, Deployments, Metrics, Tags, and Logs.

The screenshot shows the AWS ECS Task Details page for a task named `a92d17a2-260c-4fdb-b880-cc48376ea9ec`. The task is running in the `microcosm` cluster using the FARGATE launch type. The platform version is 1.3.0, and the task definition is `first-run-task-definition:3`. The task is currently `RUNNING`. It was created at 2019-04-11 08:46:48 -0600 and started at 2019-04-11 08:47:46 -0600. The task role is None. The network mode is `awsvpc`, and the ENI ID is `eni-03efb0fedd0d969f2`. The subnet ID is `subnet-0e50c4d05af5052a7`, private IP is `10.0.1.191`, and public IP is `18.191.252.84`. The Mac address is `06:5e:3e:bd:77:c4`. The container table shows one container named `jenkins` with Container ID `4ca0f74c-2b10-4f11-ad6f-fcab3ac20078`, Status `RUNNING`, and Image `443007076818.dkr.ecr.us-east-2.amazonaws.com/jenkins:latest`. The CPU Units are 0, and the Hard/Soft memory limit is `--/-`. The container is marked as Essential.

Clusters > microcosm > Task: a92d17a2-260c-4fdb-b880-cc48376ea9ec

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-03efb0fedd0d969f2
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.ecr.us-east-2.amazonaws.com/jenkins:latest	0	--/-	true

The screenshot shows the AWS ECS Task Definitions page. The left sidebar has 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 are 'Create new Task Definition', 'Create new revision', and 'Actions' dropdown buttons. To the right of these is a timestamp 'Last updated on April 11, 2019 8:49:12 AM (0m ago)' and a refresh icon. A status filter 'Status: ACTIVE INACTIVE' is shown. Below is a search bar with the placeholder 'Filter in this page'. On the right, there are navigation arrows ('< 1-1 >'), a 'Page size' dropdown set to 50, and a help icon. The main table has two columns: 'Task Definition' and 'Latest revision status'. One row is visible, showing 'first-run-task-definition' and 'ACTIVE'. The footer includes 'Feedback', 'English (US)', a copyright notice '© 2008 - 2019, Amazon Web Services, Inc. or its affiliates. All rights reserved.', and links for 'Privacy Policy' and 'Terms of Use'.

The screenshot shows the AWS ECS Clusters page. On the left, a sidebar lists various services: 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. It features a "Create Cluster" button and a "Get Started" button. Below this, there's a summary for the "microcosm > FARGATE" cluster. The summary shows:

EC2	Services	Running tasks	Pending tasks	CPUUtilization	MemoryUtilization	Container instances
1	1	1	0	No data	No data	0
Services	Running tasks	Pending tasks	CPUUtilization	MemoryUtilization	Container instances	

At the bottom, there are links for Feedback, English (US), and navigation icons.

Everything Created By Get Started Wizard

Follow the screen shots for reference

The screenshot shows the AWS CloudFormation console interface. At the top, there's a banner announcing the redesigned console. Below it, the main dashboard displays a table of stacks. A single stack, "EC2ContainerService-microcosm", is listed with the following details:

Stack Name	Created Time	Status	Drift Status	Description
EC2ContainerService-microcosm	2019-04-11 08:45:29 UTC-0600	CREATE_COMPLETE	NOT_CHECKED	AWS CloudFormation template to create a new ECS Fargate First Run stack

Below the table, there's a navigation bar with tabs: Overview, Outputs, Resources, Events, Template, Parameters, Tags, Stack Policy, Change Sets, and Rollback Triggers. The "Overview" tab is selected. A message "Select a stack" is displayed in the center of the page.

The screenshot shows the AWS CloudFormation Stack Detail page for the stack named "EC2ContainerService-microcosm". The stack is in a "CREATE_COMPLETE" status. Key details include:

- 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:** (None)
- Termination protection:** Disabled
- Drift status:** NOT_CHECKED [View details](#)
- Last drift check time:** (None)
- IAM role:** (None)
- 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	(None)
EcsElbName	(None)	Load Balancer for ECS Service	(None)

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::VPGatewayAttachment	NOT_CHECKED	CREATE_COMPLETE	(None)
EcsSecurityGroup	sg-06f07d8d65ef6b2fd	AWS::EC2::SecurityGroup	NOT_CHECKED	CREATE_COMPLETE	(None)
InternetGateway	igw-0db5c3b0358004693	AWS::EC2::InternetGateway	NOT_CHECKED	CREATE_COMPLETE	(None)

<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%2FEC2ContainerService-microcosm>

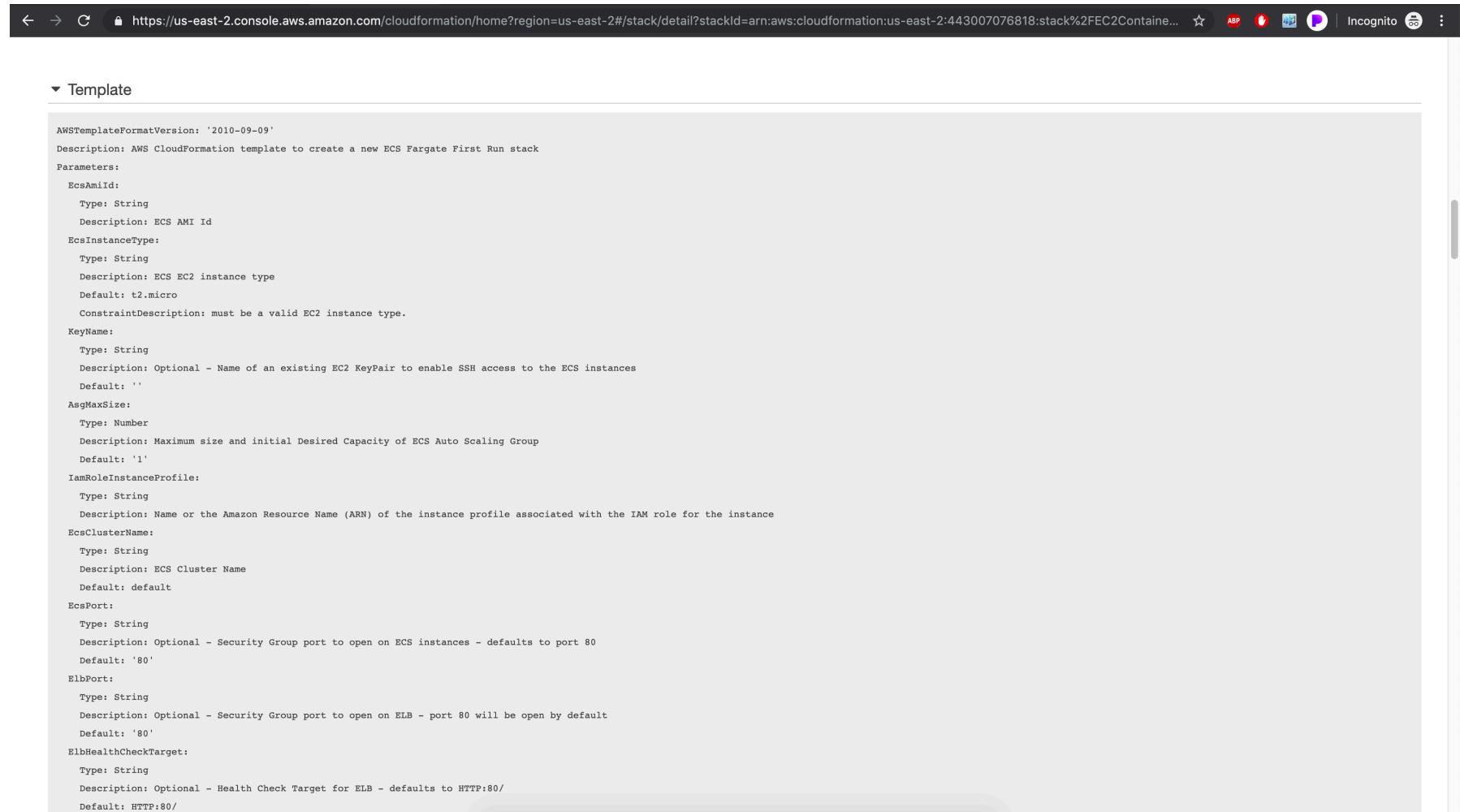
Resource Name	Logical ID	Type	Status	Reason
InternetGateway	igw-0db5c3b0358004693	AWS::EC2::InternetGateway	NOT_CHECKED	CREATE_COMPLETE
PublicRouteVialgw	EC2Co-Publi-1QU2JZOBMR836	AWS::EC2::Route	NOT_CHECKED	CREATE_COMPLETE
PublicSubnet1RouteTable...	rtbassoc-07dca01864197254f	AWS::EC2::SubnetRouteTableAssociation	NOT_CHECKED	CREATE_COMPLETE
PublicSubnet2RouteTable...	rtbassoc-05c00117481d09ba1	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-0f28d4e934944d0fc	AWS::EC2::RouteTable	NOT_CHECKED	CREATE_COMPLETE
Vpc	vpc-0530acf15c751393	AWS::EC2::VPC	NOT_CHECKED	CREATE_COMPLETE

▼ Events

Filter by:	Status	Search events		
2019-04-11	Status			
▶ 08:46:35 UTC-0600	CREATE_COMPLETE	AWS::CloudFormation::Stack	EC2ContainerService-microcosm	
▶ 08:46:30 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::SubnetRouteTableAssociation	PublicSubnet2RouteTableAssociation	
▶ 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	

The screenshot shows the AWS CloudFormation console with a stack named 'EC2ContainerService-microcosm' in the 'CREATE_IN_PROGRESS' state. The timestamp is 08:45:29 UTC-0600. The status message indicates 'User Initiated'. The 'Template' section is expanded, and the 'Parameters' section is collapsed. A table lists various parameters with their values and resolved values.

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	



The screenshot shows a browser window displaying the AWS CloudFormation console at the URL <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...>. The page title is "Template". The template content is as follows:

```
AWSTemplateFormatVersion: '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/
```

The screenshot shows the AWS ECS Cluster details page for a cluster named 'microcosm'. The left sidebar navigation includes 'Clusters' (selected), 'Task Definitions', 'Amazon EKS', 'Clusters', 'Amazon ECR', 'Repositories', 'AWS Marketplace', 'Discover software', and 'Subscriptions'. The main content area displays cluster statistics: Status (ACTIVE), Registered container instances (0), Pending tasks count (0 Fargate, 0 EC2), Running tasks count (1 Fargate, 0 EC2), Active service count (1 Fargate, 0 EC2), and Draining service count (0 Fargate, 0 EC2). Below this, a table lists services, with one entry for 'jenkins-service'. The table columns include Service Name, Status, Service type, Task Definition, Desired tasks, Running tasks, Launch type, and Platform version. The 'jenkins-service' row shows ACTIVE status, REPLICA service type, first-run-task-de... task definition, 1 desired task, 1 running task, FARGATE launch type, and LATEST(1.3.0) platform version. Action buttons for Create, Update, Delete, and Actions are available at the top of the service table.

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

The screenshot shows the AWS VPC Dashboard for the US East (Ohio) region. On the left, a sidebar lists various VPC-related services. The main area displays 'Resources by Region' with counts for VPCs, NAT Gateways, 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. A note at the top states: 'Note: Your Instances will launch in the US East (Ohio) region.' To the right, the 'Service Health' section shows a green status for 'Amazon EC2 - US East (Ohio)' with the message 'Service is operating normally'. Below it, the 'Account Attributes' section includes links for 'Resource ID length management', 'VPC Documentation', 'All VPC Resources', 'Forums', and 'Report an Issue'. The 'Additional Information' section features a 'Create VPN Connection' button. The 'Site-to-Site VPN Connections' section explains the service's purpose and includes a 'See all regions' link.

VPC Dashboard

Filter by VPC:
Select a VPC

Virtual Private Cloud

Your VPCs

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

Virtual Private Gateways

Open to Site-to-Site VPN

Launch VPC Wizard Launch EC2 Instances

Note: Your Instances will launch in the US East (Ohio) region.

Resources by Region Refresh Resources

You are using the following Amazon VPC resources

VPCs See all regions	Ohio 2
NAT Gateways See all regions	Ohio 0
Subnets See all regions	Ohio 5
VPC Peering Connections See all regions	Ohio 0
Route Tables See all regions	Ohio 3
Network ACLs See all regions	Ohio 2
Internet Gateways See all regions	Ohio 2
Security Groups See all regions	Ohio 3
Egress-only Internet Gateways See all regions	Ohio 0
Customer Gateways See all regions	Ohio 0
DHCP options sets See all regions	Ohio 1
Virtual Private Gateways See all regions	Ohio 0
Elastic IPs See all regions	Ohio 0
Site-to-Site VPN Connections See all regions	Ohio 0
Endpoints See all regions	Ohio 0
Running Instances See all regions	Ohio 0
Endpoint Services	Ohio 0

Service Health

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

Account Attributes

Resource ID length management

VPC Documentation

All VPC Resources

Forums

Report an Issue

Additional Information

Create VPN Connection

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)

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

The screenshot shows the AWS VPC Dashboard. On the left, a sidebar lists various VPC-related services: VPC Dashboard, Virtual Private Cloud, Your VPCs, Subnets, Route Tables, Internet Gateways, Egress Only Internet Gateways, DHCP Options Sets, Elastic IPs, Endpoints, Endpoint Services, NAT Gateways, Peering Connections, Security, Network ACLs, and Security Groups. Below these are sections for Virtual Private Network (VPN) and Customer Gateways. At the bottom of the sidebar, there's a link to 'Create a VPC'.

The main content area displays a table of VPCs. The columns are: Name, VPC ID, State, IPv4 CIDR, IPv6 CIDR, DHCP options set, Main Route table, Main Network ACL, and Ter. There are two entries:

Name	VPC ID	State	IPv4 CIDR	IPv6 CIDR	DHCP options set	Main Route table	Main Network ACL	Ter
ECS microc...	vpc-0530acf15c751393	available	10.0.0.0/16	-	dopt-5f946034	rtb-089b8ba33139a46fa	acl-07f738971cf93937c	defi
	vpc-9dd0c3f5	available	172.31.0....	-	dopt-5f946034	rtb-d764bcfc	acl-597f8132	defi

At the bottom of the page, there are links for 'Feedback', 'English (US)', and 'Help'.

The screenshot shows the AWS VPC Dashboard. On the left, a sidebar lists various VPC-related services: VPC Dashboard, Virtual Private Cloud, Your VPCs, Subnets (which is selected), Route Tables, Internet Gateways, Egress Only Internet Gateways, DHCP Options Sets, Elastic IPs, Endpoints, Endpoint Services, NAT Gateways, Peering Connections, Security, Network ACLs, and Security Groups. Below these are sections for Virtual Private Network (VPN) and Customer Gateways. At the bottom of the sidebar, there's a link to 'Create a VPC'.

The main content area has tabs for 'Create subnet' and 'Actions'. A search bar at the top of the main table says 'Filter by tags and attributes or search by keyword'. The table displays five subnets:

Name	Subnet ID	State	VPC	IPv4 CIDR	Available IPv4	IPv6 CIDR	Availability Zone	Availability Zone ID	Route
ECS microc...	subnet-0521ea54a64435dd4	available	vpc-0530acf15c751393 ...	10.0.0.0/24	250	-	us-east-2a	use2-az1	rtb-0
ECS microc...	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

At the bottom of the page, there are links for 'Feedback', 'English (US)', 'Privacy Policy', and 'Terms of Use'. The footer contains the text '© 2008 - 2019, Amazon Web Services, Inc. or its affiliates. All rights reserved.'

The screenshot shows the AWS VPC Dashboard with the 'Route Tables' section selected. On the left sidebar, there is a navigation menu with various options like 'Virtual Private Cloud', 'Your VPCs', 'Subnets', 'Route Tables' (which is highlighted), 'Internet Gateways', 'Egress Only Internet Gateways' (in orange), '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 titled 'Route Tables' with three entries:

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-d764bcfc	rtb-d764bcfc	-	Yes	vpc-9dd0c3f5	443007076818

At the bottom of the page, there is a footer bar with the URL <https://us-east-2.console.aws.amazon.com/vpc/home?region=us-east-2#EgressOnlyInternetGateways>, a copyright notice '© 2008 - 2019, Amazon Web Services, Inc. or its affiliates. All rights reserved.', and links for 'Privacy Policy' and 'Terms of Use'.

The screenshot shows the AWS VPC Dashboard with the 'Internet Gateways' section selected. A table lists two existing Internet Gateways:

	Name	ID	State	VPC	Owner
<input type="checkbox"/>	ECS microc...	igw-0db5c3b0358...	attached	vpc-0530acf15c7...	443007076818
<input type="checkbox"/>		igw-d403bebc	attached	vpc-9dd0c3f5	443007076818

A message at the bottom left says "Select an internet gateway above". The navigation bar at the top includes links for Services, Resource Groups, and Actions.

The screenshot shows the AWS VPC Security Groups page. The left sidebar navigation includes: Internet Gateways, Egress Only Internet Gateways, DHCP Options Sets, Elastic IPs, Endpoints, Endpoint Services, NAT Gateways, Peering Connections, Security, Network ACLs, and **Security Groups** (which is selected). Other sections like Virtual Private Network (VPN), Customer Gateways, and Transit Gateways are also listed. The main content area displays a table of security groups:

Name	Group ID	Group Name	VPC ID	Type	Description	Owner
ECS microc...	sg-06f07d8d65ef6...	EC2ContainerSer...	vpc-0530acfc15c7...	EC2-VPC	ECS Allowed Ports	443007076818
	sg-0f84b4941885...	default	vpc-0530acfc15c7...	EC2-VPC	default VPC securi...	443007076818
	sg-1e087170	default	vpc-9dd0c3f5	EC2-VPC	default VPC securi...	443007076818

At the bottom, there are links for Feedback, English (US), and a footer note: © 2008 - 2019, Amazon Web Services, Inc. or its affiliates. All rights reserved. Privacy Policy Terms of Use.

The screenshot shows the AWS EC2 Dashboard for the US East (Ohio) region. The left sidebar contains navigation links for EC2 Dashboard, Events, Tags, Reports, Limits, Instances, Launch Templates, Spot Requests, Reserved Instances, Dedicated Hosts, Capacity Reservations, Images, AMIs, Bundle Tasks, Elastic Block Store, Volumes, Snapshots, Lifecycle Manager, Network & Security, Security Groups, Elastic IPs, Placement Groups, Key Pairs, Network Interfaces, Load Balancing, Load Balancers, Target Groups, and Auto Scaling, Launch Configurations.

Resources

You are using the following Amazon EC2 resources in the US East (Ohio) region:

0 Running Instances	0 Elastic IPs
0 Dedicated Hosts	0 Snapshots
0 Volumes	0 Load Balancers
0 Key Pairs	3 Security Groups
0 Placement Groups	

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

Create Instance

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

[Launch Instance](#)

Note: Your instances will launch in the US East (Ohio) region

Service Health

Service Status:

- US East (Ohio): ✓ (Operating normally)

Availability Zone Status:

- us-east-2a: Availability zone is operating normally
- us-east-2b: Availability zone is operating normally
- us-east-2c: Availability zone is operating normally

[Service Health Dashboard](#)

Scheduled Events

US East (Ohio):
No events

AWS Marketplace

Find free software trial products in the AWS Marketplace from the [EC2 Launch Wizard](#). Or try these popular AMIs:

Barracuda CloudGen Firewall for AWS - PAYG
By Barracuda Networks, Inc.
Rating ★★★★★
Starting from \$0.60/hr or from \$4,599/yr (12% savings)
for software + AWS usage fees
[View all Infrastructure Software](#)

Matillion ETL for Amazon Redshift
By Matillion
Rating ★★★★★
Starting from \$1.37/hr or from \$9,950/yr (17% savings)
for software + AWS usage fees
[View all Business Software](#)

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

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

The screenshot shows the AWS ECS Task Definitions page. The left sidebar has 'Task Definitions' selected. The main content area is titled 'Task Definitions' and contains a table with one row. The table has two columns: 'Task Definition' and 'Latest revision status'. The first row shows 'first-run-task-definition' with 'ACTIVE' status. There are buttons for 'Create new Task Definition', 'Create new revision', and 'Actions'.

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

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 wh

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

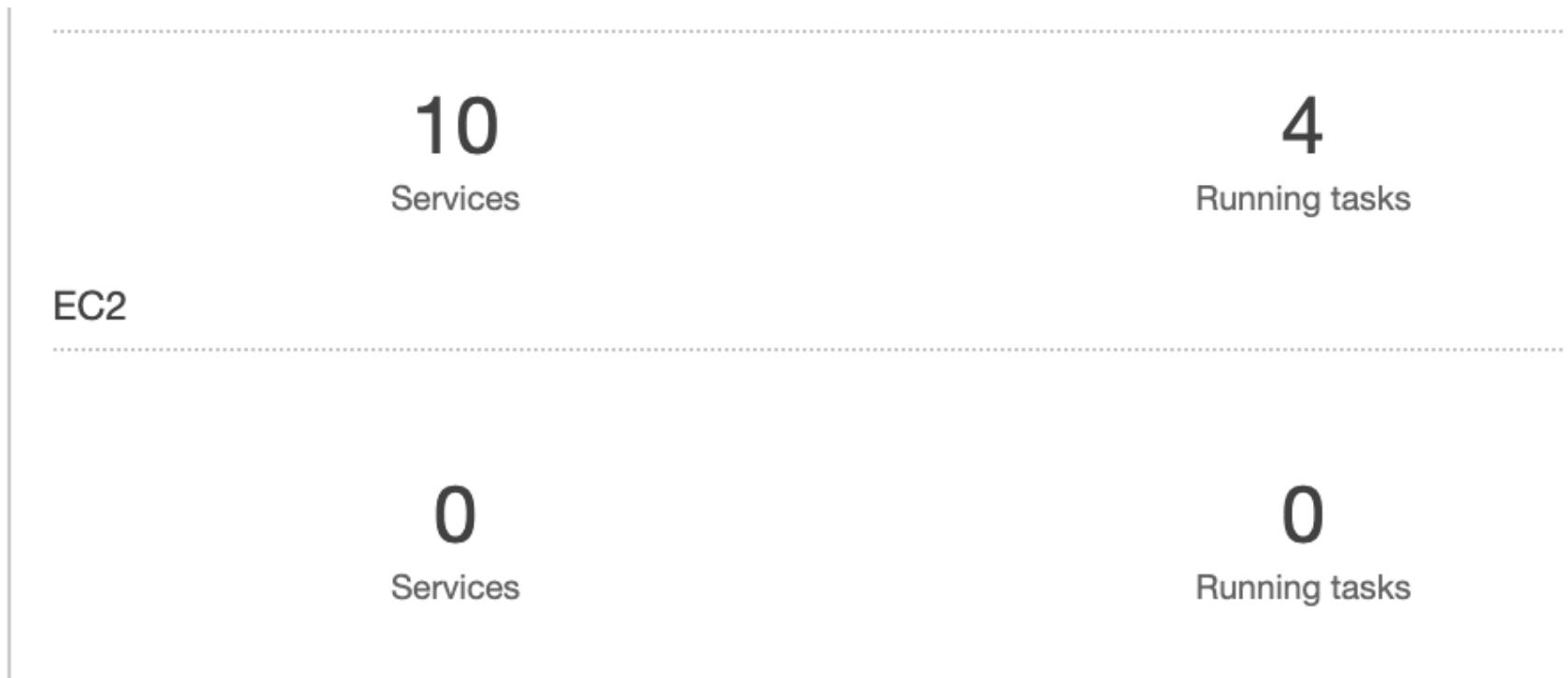


Opt in to the new ARN and resource ID format

Amazon ECS has introduced a new format for ARNs and resource IDs. The A

[Configure ECS ARN setting](#)

[Create Cluster](#)[Get Started](#)[View](#) [list](#) [card](#)[microcosm >](#)[FARGATE](#)



- 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**

Create **Update** **Delete** **Actions ▾**

Filter in this page Launch type: ALL Service type: AL

<input type="checkbox"/>	Service Name	Status
<input type="checkbox"/>	sonarqube	ACTIVE
<input type="checkbox"/>	gitlab	ACTIVE
<input type="checkbox"/>	hubitat	ACTIVE

<input type="checkbox"/>	nuvola	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)

Clusters > microcosm > Service: cloud-discovery

Service : cloud-discovery

Cluster: microcosm Desired count: 1
Status: ACTIVE Pending count: 0
Task definition: cloud-discovery:1 Running count: 1

Update Delete

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

Number of tasks i

Minimum healthy percent i

Maximum percent i

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

- Press Update service and the View Service
-

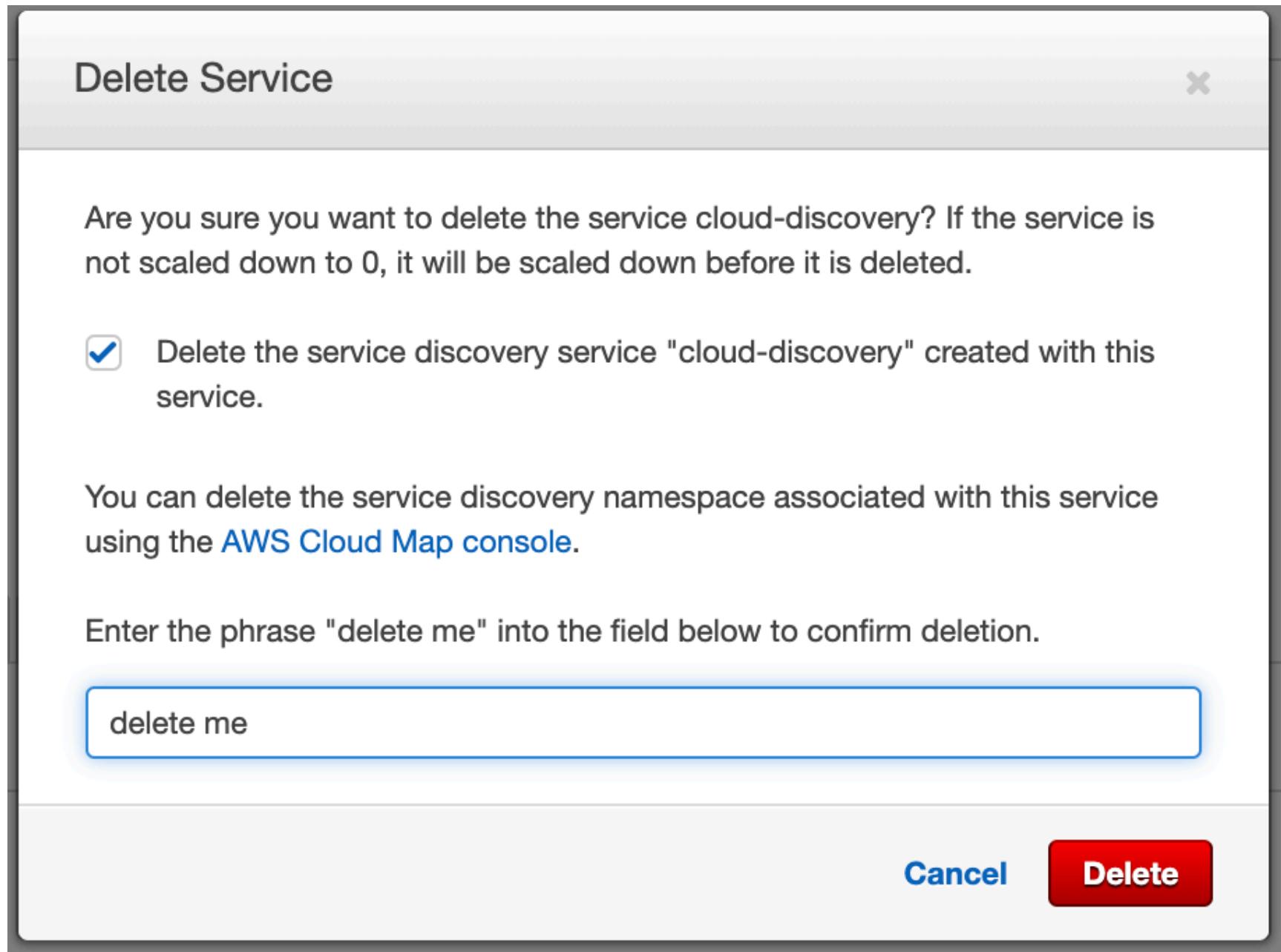
[Cancel](#)[Previous](#)[Update Service](#)

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

Deployment						
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

A blue rectangular button with a white border and rounded corners, containing the word "Update" in white capital letters.A grey rectangular button with a white border and rounded corners, containing the word "Delete" in black capital letters.

- Press Delete in the top right corner
- 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.

Launch Instance ▾ Connect Actions ▾

Filter by tags and attributes or search

	Name	Inst...	Availability Zone	Instance State	Status Checks
<input type="checkbox"/>	tomcat_codedeployTST	i-01	us-east-2a	● running	✓ 2/2 checks ...
<input type="checkbox"/>	tomcat_codedeploy	i-08			
<input type="checkbox"/>	tomcat_codedeploy	i-0c			

Connect
Get Windows Password
Create Template From Instance
Launch More Like This

Instance State ▾
Instance Settings ▾
Image ▾
Networking ▾
CloudWatch Monitoring ▾

Add/Edit Tags
Attach to Auto Scaling Group
Attach/Replace IAM Role
Change Instance Type
Change Termination Protection
View/Change User Data
Change Shutdown Behavior
Change T2/T3 Unlimited
Get System Log
Get Instance Screenshot
Modify Instance Placement
Modify Capacity Reservation Settings

AMI ID	Amazon Linux 2018.03.0 - Amazon Linux 2018.03.0 AMI - AWS Marketplace 2fcfbcb81353-ami-06be36ea8b6c786a7.4 (ami-0885e0cd6c51e6b89)
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

Role ARN	arn:aws:iam::443007076818:role/EC2PlusS3RoleTST
Role description	Edit
Instance Profile ARNs	arn:aws:iam::443007076818:instance-profile/EC2PlusS3RoleTST
Path	/

Creation time

2019-05-06 14:30 MDT

Maximum CLI/API session duration1 hour [Edit](#)**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* EC2PlusS3RoleTST [▼](#) [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](#)