

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](#)

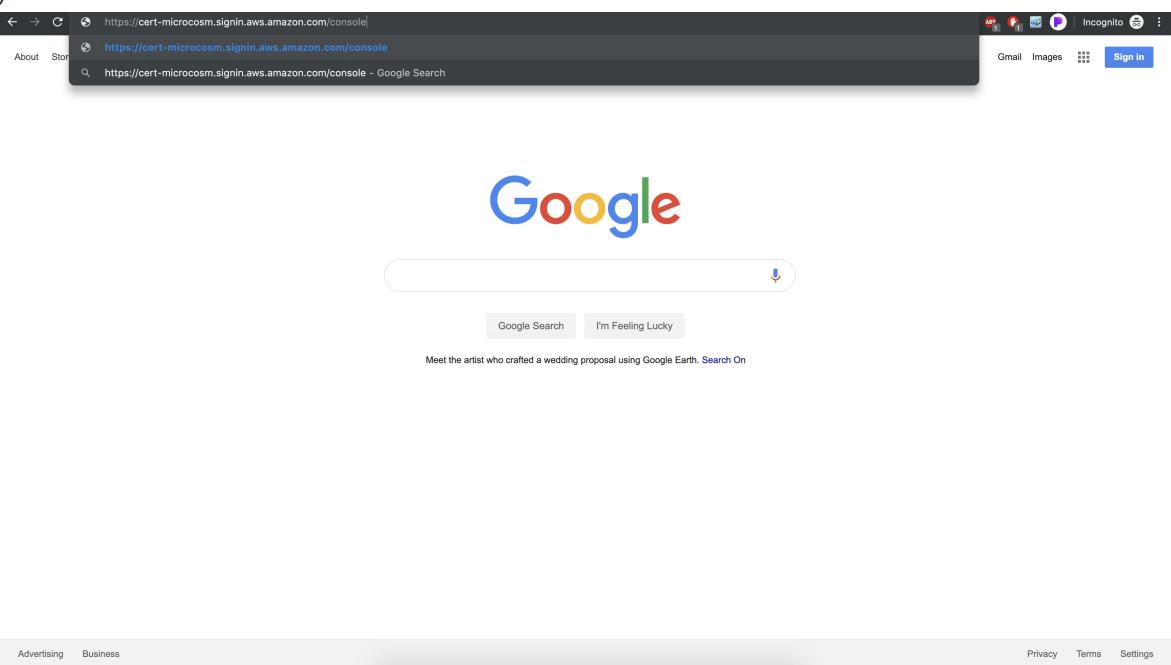
Introduction

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](#)

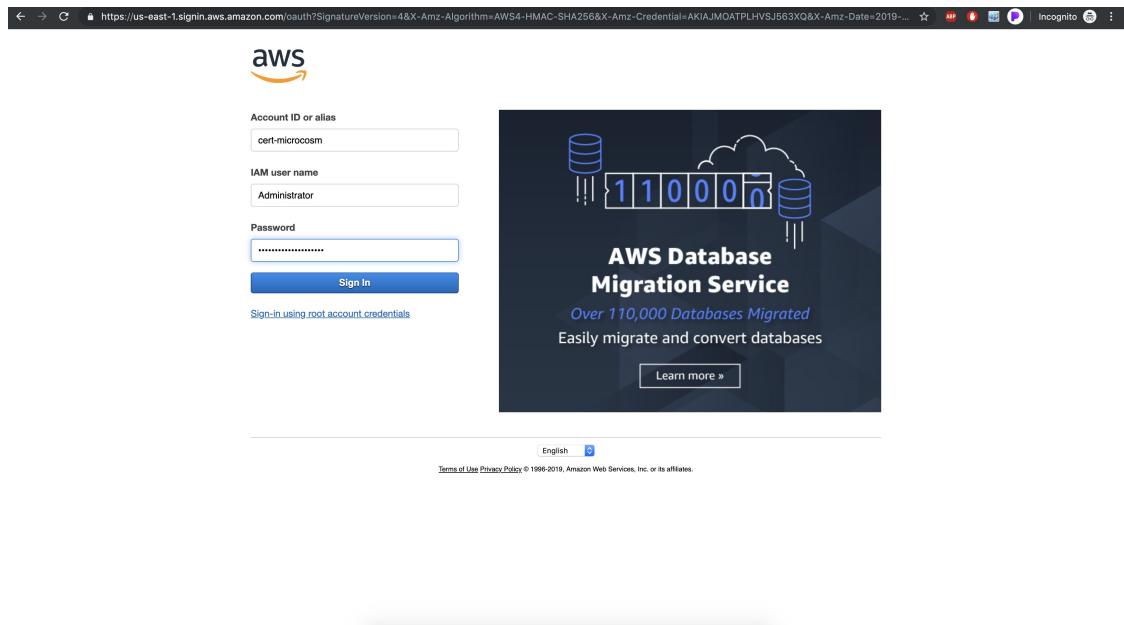
Login

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

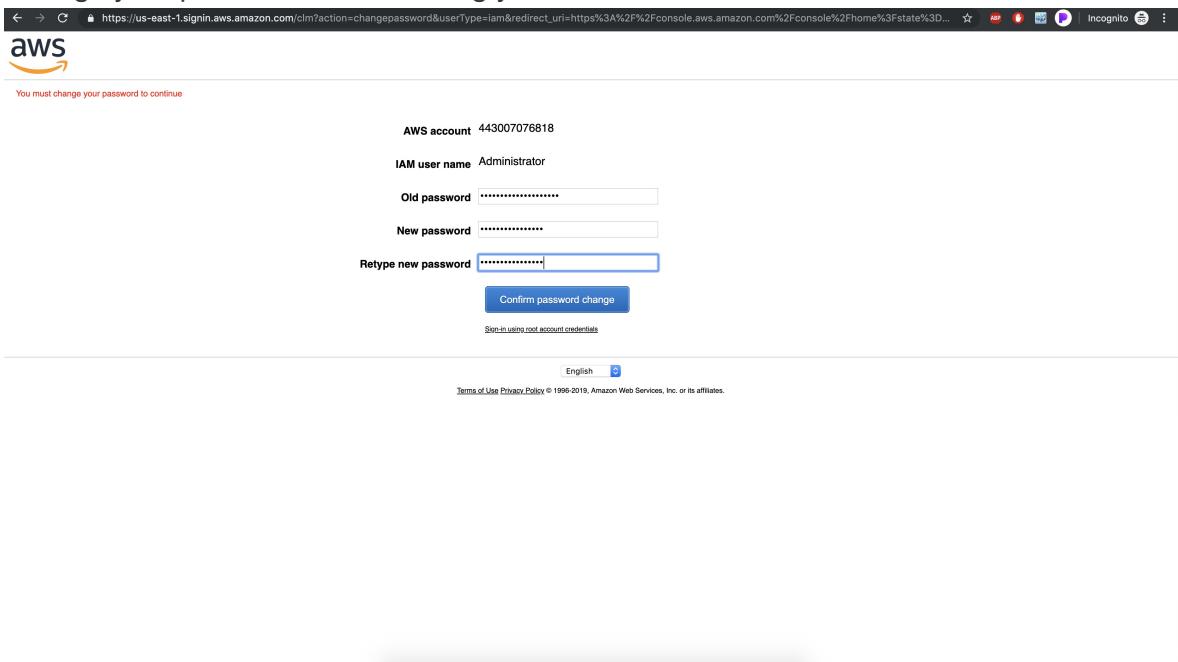


- Enter the credentials received

- User:[stackname]-StudentUser[number]-[hash]
- pwd: \$bzq\$UrFLxw9HFtB-49eRtf!



- Change your password to something you will remember



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

The screenshot shows the AWS Management Console homepage. On the left, there's a search bar with placeholder text "Example: Relational Database Service, database, RDS". Below it, a "Build a solution" section lists six quick start options with icons and descriptions:

- Launch a virtual machine** (With EC2, 2-3 minutes) - icon: server
- Build a web app** (With Elastic Beanstalk, 6 minutes) - icon: cloud with code
- Build using virtual servers** (With Lightsail, 1-2 minutes) - icon: server with circular arrows
- Connect an IoT device** (With AWS IoT, 5 minutes) - icon: IoT device
- Start a development project** (With CodeStar, 5 minutes) - icon: developer tools
- Register a domain** (With Route 53, 3 minutes) - icon: shield with 53

On the right, there are two sections: "Access resources on the go" (with a mobile phone icon) and "Explore AWS" (listing 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, there's a sidebar with links for Amazon ECS, Clusters, Task Definitions, Amazon EKS, Amazon ECR, Repositories, AWS Marketplace, Discover software, and Subscriptions. The main content area is titled 'Clusters' and contains a message about the new ARN and resource ID format. Below this is a 'Create Cluster' button and a 'Get Started' button. A table lists one cluster: 'microcosm' with 5 services, 6 running tasks, 0 pending tasks, and 0 container instances. At the bottom, there are links for Feedback, English (US), and a copyright notice.

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

The screenshot shows the 'Create Cluster' wizard, Step 1: Select cluster template. It has three options: 'Networking only', 'EC2 Linux + Networking', and 'EC2 Windows + Networking'. The 'Networking only' option is selected. Each option shows the resources to be created: 'Cluster', 'VPC (optional)', and 'Subnets (optional)' for the first; 'Cluster', 'VPC', and 'Subnets' for the second; and 'Cluster', 'VPC', 'Subnets', and 'Auto Scaling group with Linux AMI' for the third. At the bottom, there are 'Required' 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.



Create Cluster

Step 1: Select cluster template

Step 2: Configure cluster

Configure cluster

Cluster name* ⓘ

Networking

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

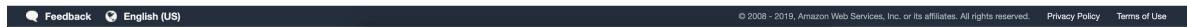
Create VPC Create a new VPC for this cluster

Tags

Key	Value
Add key	Add value

*Required

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



- Once the cluster is created successfully select View Cluster



Launch status

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

[Back](#) [View Cluster](#)

ECS status - 1 of 1 complete StudentCluster

ECS cluster
ECS Cluster StudentCluster successfully created



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

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

Task Definitions

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

Microcosm Task Definitions

Use the following table to create the task definitions necessary to stand up the Microcosm DevOps Pipeline

Important: for the Hubot task, replace the bracketed portion with the hubot key when generated

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

[Return to Table of Contents](#)

Define a task

Repeat the Following for each desired task definition

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

The screenshot shows the AWS ECS Task Definitions page. On the left, there's a sidebar with links for Amazon ECS, Clusters, Task Definitions (which is selected and highlighted in orange), Amazon EKS, Amazon ECR, Repositories, AWS Marketplace, Discover software, and Subscriptions. The main content area has a title "Task Definitions" and a sub-instruction: "Task definitions specify the container information for your application, such as how many containers are part of your task, what resources they will use, how they are linked together, and which host ports they will use." Below this is a "Learn more" link. At the top right, there are buttons for "Create new Task Definition", "Create new revision", and "Actions". A status filter is set to "ACTIVE". The table below lists task definitions with their latest revision status:

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

At the bottom, there are "Feedback" and "English (US)" buttons, along with copyright and legal links.

- Select the Fargate Type and press Next Step

The screenshot shows the "Create new Task Definition" step 1: "Select launch type compatibility". It has two options: "FARGATE" and "EC2".

FARGATE
Icon: A yellow cube with a smaller cube on top.
Price based on task size
Requires network mode awsvpc
AWS-managed infrastructure, no Amazon EC2 instances to manage

EC2
Icon: Three yellow squares stacked vertically.
Price based on resource usage
Multiple network modes available
Self-managed infrastructure using Amazon EC2 instances

*Required Cancel Next step

At the bottom, there are "Feedback" and "English (US)" buttons, along with copyright and legal links.

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

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* ⓘ

Requires Compatibilities* FARGATE

Task Role ⓘ

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 ⓘ

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 ⓘ

Task size

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

Add volume

Name ⓘ

Specify a volume driver
This feature is not supported when the Fargate launch type is used.

*Required

Cancel Add

Task size

The task size allows you to specify a fixed size for your task. Task size is required for tasks using the Fargate launch type and is optional for the EC2 launch type. Container-level memory settings are optional when task size is set. Task size is not supported for Windows containers.

Container Definitions

Add container

Container N...	Image	Hard/Soft m...	CPU Units	GPU	Essential
No results					

Volumes

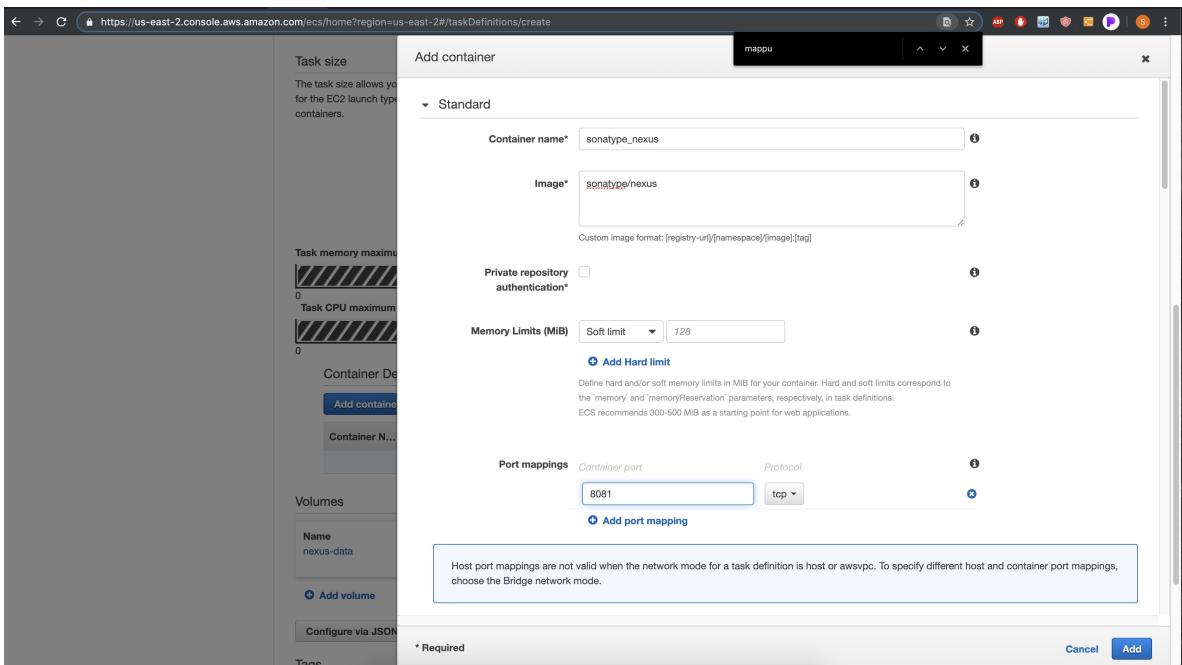
Add volume

Configure via JSON

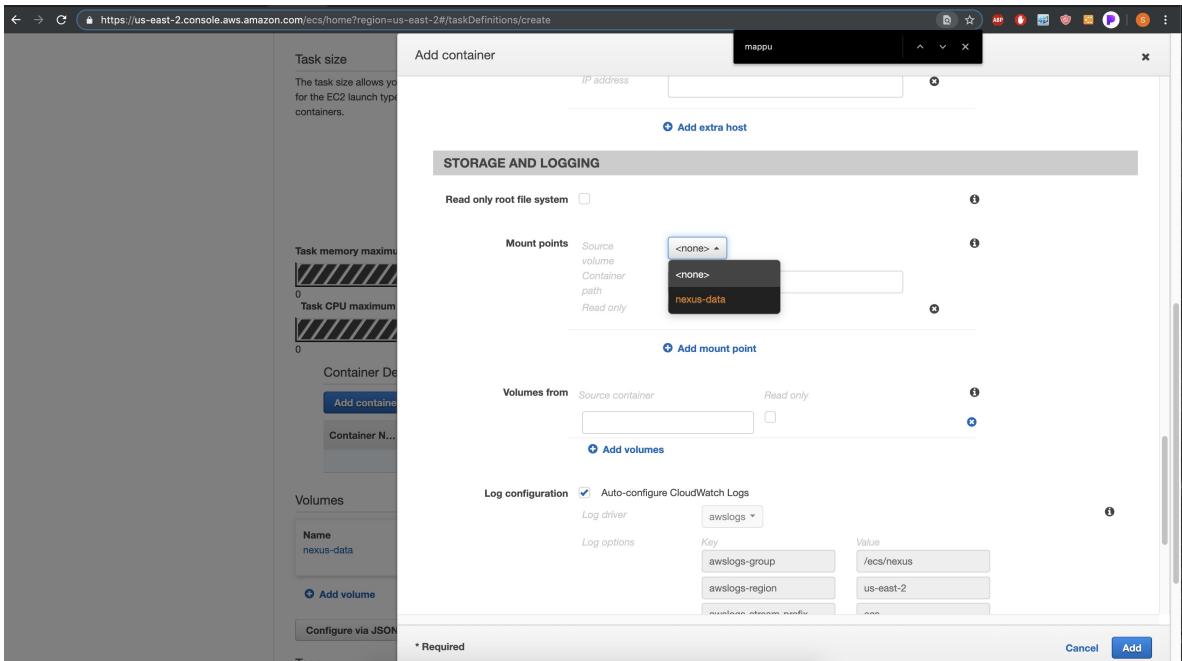
Tags

Add key Add value

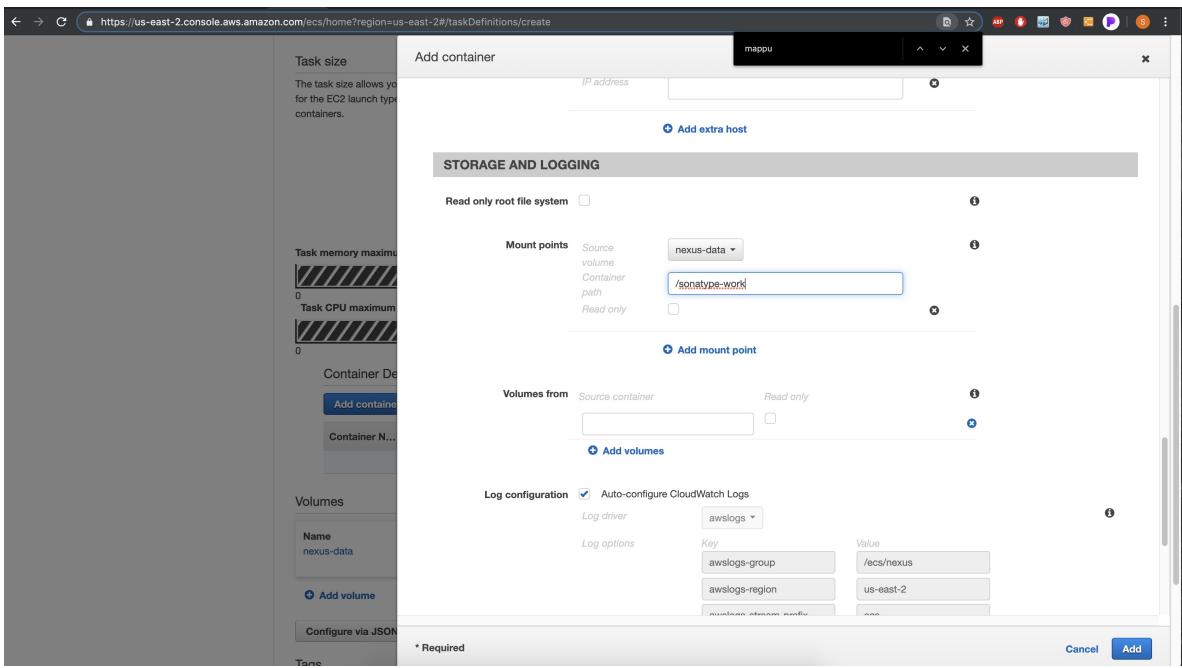
- Select Add Container.
- Enter the Container Name, Image, Port mappings number(s) (select Add port mapping to add additional ports)



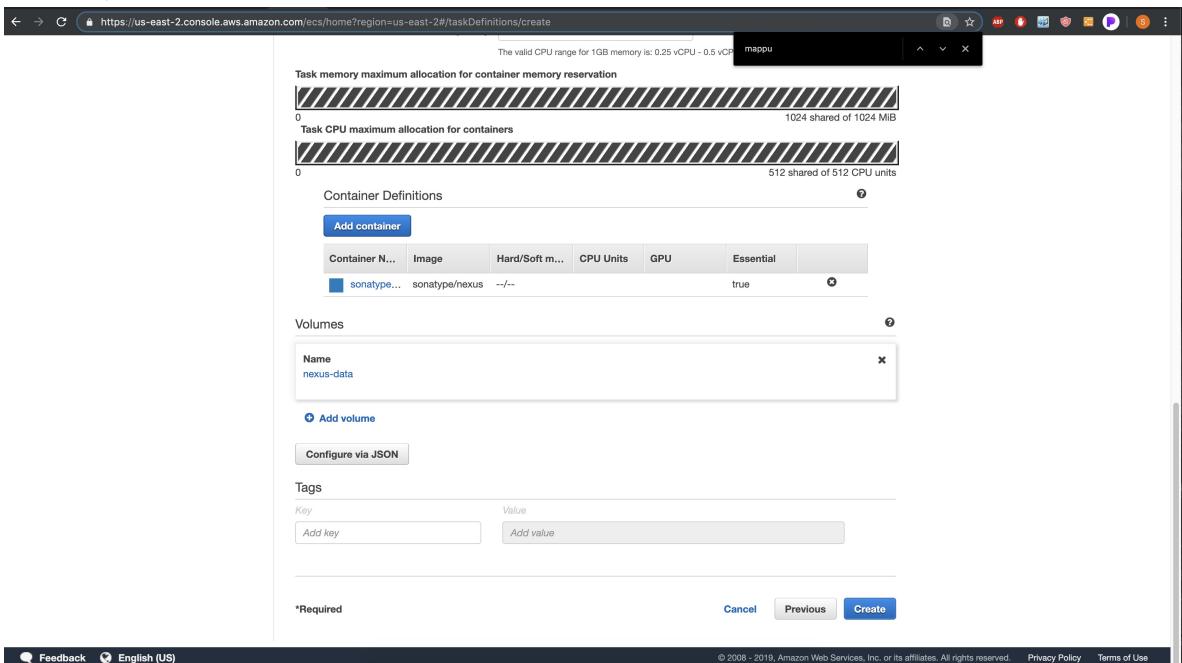
- Under Mount Points, if applicable, select a volume entered previously from the drop down



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



- Add Tags if desired and press create



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

The screenshot shows the AWS CloudWatch Log Group creation status. It displays two successful operations: 'Create Task Definition: nexus' (status: succeeded) and 'Create CloudWatch Log Group' (status: CloudWatch Log Group created). The URL is <https://us-east-2.console.aws.amazon.com/ecs/home?region=us-east-2#/taskDefinitions/create>.

- Review the task definition, if desired

The screenshot shows the AWS Task Definition configuration page for 'Task Definition: nexus:1'. The configuration includes:

- Task Definition Name: nexus
- Task Role: ecsTaskExecutionRole
- Network Mode: awsvpc (Note: 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.)
- Compatibilities: EC2, FARGATE
- Requires compatibility: FARGATE
- 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
- Task size: (empty)

The URL is <https://us-east-2.console.aws.amazon.com/ecs/home?region=us-east-2#/taskDefinitions/nexus/1>.

[Return to Table of Contents](#)

Services

Services are constructs within ECS that allow for the management, grouping, load balancing, fault tolerance, etc. of tasks. When a service is created based on a task definition, ECS will instantiate tasks (containers) automatically and re-start them up if they fail. It is possible to have multiple different task definitions, of the same or different types, within a single service. Note, it is possible to have tasks not managed by a service, these have no fault tolerance nor take advantage of any other service properties. In this example we will create a service based

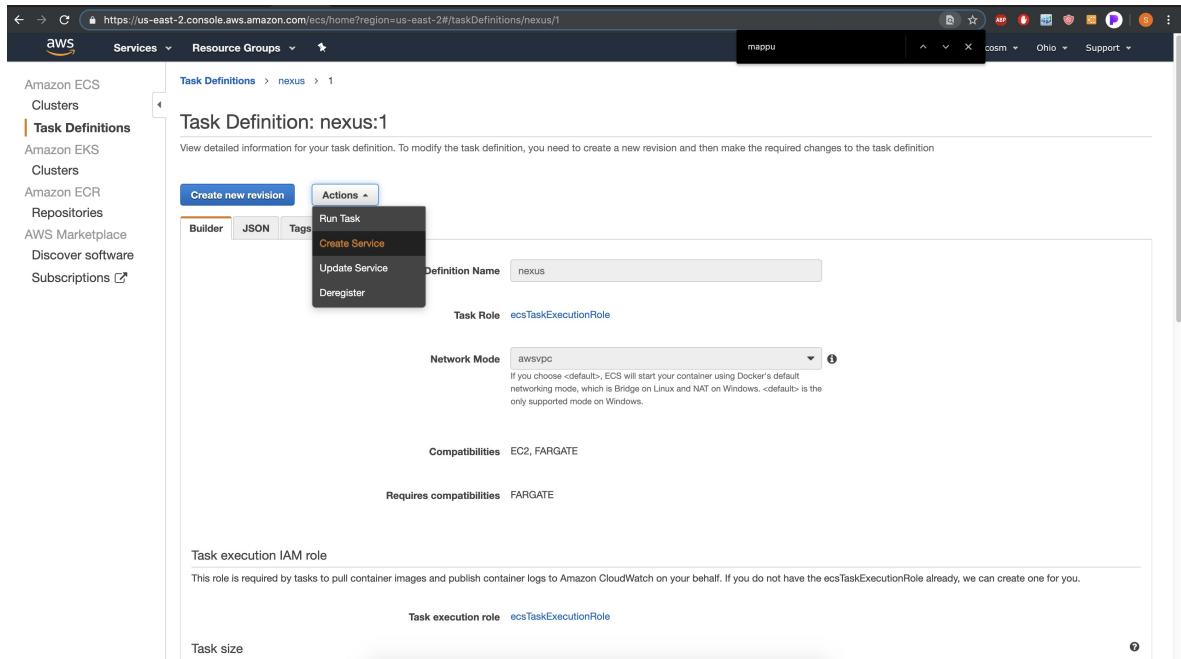
on a task definition and let ECS create the task for us. Note, before deleting a service, you must stop all running tasks of that service and set the number of tasks to 0, else the service will restart the tasks before the service can be deleted.

####Create a Service

Repeat the Following for each desired service definition

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

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



The screenshot shows the AWS ECS Task Definitions console. On the left, there's a sidebar with links like 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 area has a breadcrumb navigation: Task Definitions > nexus > 1. Below it, there's a title 'Task Definition: nexus:1' and a sub-instruction: 'View detailed information for your task definition. To modify the task definition, you need to create a new revision and then make the required changes to the task definition'. There are three tabs at the top: 'Create new revision' (blue), 'Actions' (dropdown menu), and 'Builder' (selected). The 'Actions' dropdown is open, showing options: Run Task, Create Service (which is highlighted in red), Update Service, and Deregister. Other fields include 'Definition Name' (nexus), 'Task Role' (ecsTaskExecutionRole), 'Network Mode' (awsvpc), 'Compatibilities' (EC2, FARGATE), and 'Requires compatibilities' (FARGATE). At the bottom, there's a section for 'Task execution IAM role' with a note: '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.' A 'Task execution role' field shows 'ecsTaskExecutionRole'. At the very bottom, there's a 'Task size' section.

- 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

Step 1: Configure service

- Step 2: Configure network
- Step 3: Set Auto Scaling (optional)
- Step 4: Review

Configure service

A service lets you specify how many copies of your task definition to run and maintain in a cluster. You can optionally use an Elastic Load Balancing load balancer to distribute incoming traffic to containers in your service. Amazon ECS maintains that number of tasks and coordinates task scheduling with the load balancer. You can also optionally use Service Auto Scaling to adjust the number of tasks in your service.

Launch type FARGATE EC2

Task Definition Family: **NEXUS**
Revision: **1**

Platform version **LATEST**

Cluster **jenkinsdefault**

Service name **sonatype_nexus**

Service type* **REPLICAS**

Number of tasks **1**

Minimum healthy percent **100**

Maximum percent **200**

Deployments

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

Create Service

Step 1: Configure service

Step 2: Configure network

Step 3: Set Auto Scaling (optional)

Step 4: Review

Configure network

VPC and security groups

VPC and security groups are configurable when your task definition uses the awsvpc network mode.

Cluster VPC* **vpc-0a8f63ffd28e39ced (10.0.0.0/24)**

Subnets* **subnet-0c12a48c3055e9708 (10.0.0.0/24) ECS jenkinsdefault - Public Subnet 1 - us-east-2a assign ipv6 on creation: Disabled**

Security groups* **security-group-019a463695e9708 (10.0.0.0/24) ECS jenkinsdefault - Public Subnet 1 - us-east-2a assign ipv6 on creation: Disabled**

Auto-assign public IP **Public Subnet 1 - us-east-2a**

Health check grace period If your service's tasks take a while to start and respond to ELB health checks, you can specify a health check grace period of up to 2,147,483,647 seconds during which the ECS service scheduler will ignore ELB health check status. This grace period can prevent the ECS service scheduler from marking tasks as unhealthy and stopping them before they have time to come up. This is only valid if your service is configured to use a load balancer.

Health check grace period requires a load balancer.

Load balancing An Elastic Load Balancing load balancer distributes incoming traffic across the tasks running in your service. Choose an existing load balancer, or create a new one in the [Amazon EC2 console](#).

Load balancer None

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

Configure security groups

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

Assigned security groups

- Create new security group
- Select existing security group

Existing security groups

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

1 selected					< 0-0 >
	Security group ID	Name	Description	Actions	
<input checked="" type="checkbox"/>	sg-016e9fba44d930d92	Dynamic2-StudentVPCSecurityGro...	Student VPC Security Group	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

Create Service

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

Set Auto Scaling (optional)

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

Service Auto Scaling

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

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

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

- Review and press Create Service

The screenshot shows the AWS ECS Service creation wizard. It consists of three main configuration sections:

- Configure network:** Set to VPC ID vpc-0a8f63ffd28e39ced, Subnets subnet-0c12a48c3055e9708, Create new security group sonaty-6743, and Auto assign IP ENABLED.
- Configure service discovery:** Set to Namespace ns-iz26fljy45cjaqgg, Service discovery name sonatype_nexus, Enable ECS task health propagation true, and DNS record type and TTL A 60.
- Set Auto Scaling (optional):** Set to not configured.

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

- Once the service is successfully created, select View Service

The screenshot shows the AWS ECS Service status page, indicating 4 of 4 completed steps:

- Configure Task Networking:** Step Create security group succeeded sg-08677ef9f87f9e049.
- Set inbound rules:** Step Set inbound rules succeeded sg-08677ef9f87f9e049.
- Configure Service Discovery:** Step Create service discovery service succeeded am:aws:servicediscovery:us-east-2:443007076818:service/srv-bcfmzbuvvuxelnz created.
- Create Service:** Step Create service: sonatype_nexus succeeded. Service created. Tasks will start momentarily. View: sonatype_nexus.

Additional integrations you can connect to your ECS service are listed below.

- 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 for a task named 'Task : 325b875c-9e8f-4686-872a-937c986973d7'. The task is part of the 'jenkinsdefault' cluster, using FARGATE launch type, Platform version 1.3.0, and Task definition 'nexus:1'. It has a Group of 'service:sonatype_nexus' and a Task role of 'ecsTaskExecutionRole'. The task is ACTIVATING, Desired status is RUNNING, and it was Created at 2019-04-12 14:25:35 -0600. The Network mode is awsvpc, with ENI Id 'eni-070590cb8cb942258', Subnet Id 'subnet-0c12a48c305e9708', Private IP '10.0.0.86', Public IP '13.58.236.137', and Mac address '02:38:3b:e6:c3:da'. The Containers table shows one container named 'sonatype_nexus' with Container Id 'a7474130-6432-44a7-8105-d79153c33a3a', Status 'RUNNING', Image 'sonatype/nexus', CPU Units '0', Hard/Soft memory limits '---', and Essential status 'true'. The table was last updated on April 12, 2019, at 2:26:17 PM (0m ago).

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. It displays two clusters: 'jenkinsdefault' and 'microcosm'. The 'jenkinsdefault' cluster is selected, showing its details. It is a FARGATE cluster with 6 Services, 5 Running tasks, and 1 Pending task. Under EC2, there are 0 Services, 0 Running tasks, and 0 Pending tasks. There is also a section for CPUUtilization and MemoryUtilization, both showing 'No data'. The 'microcosm' cluster is also listed below, showing it is a FARGATE cluster with 0 services, 0 running tasks, and 0 pending tasks.

- 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 console interface for the 'jenkinsdefault' cluster. The left sidebar has 'Clusters' selected. The main area displays the cluster details: Status is ACTIVE, Registered container instances is 0, Pending tasks count is 0 Fargate, 0 EC2, Running tasks count is 6 Fargate, 0 EC2, Active service count is 6 Fargate, 0 EC2, and Draining service count is 0 Fargate, 0 EC2. Below this is a table of tasks:

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
325b675c-9e8f-4...	rexus: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
c4aa9911-accd-4...	gitlab:1	--	RUNNING	RUNNING	ecs-svc/9223370...	service:gitlab	FARGATE	1.3.0

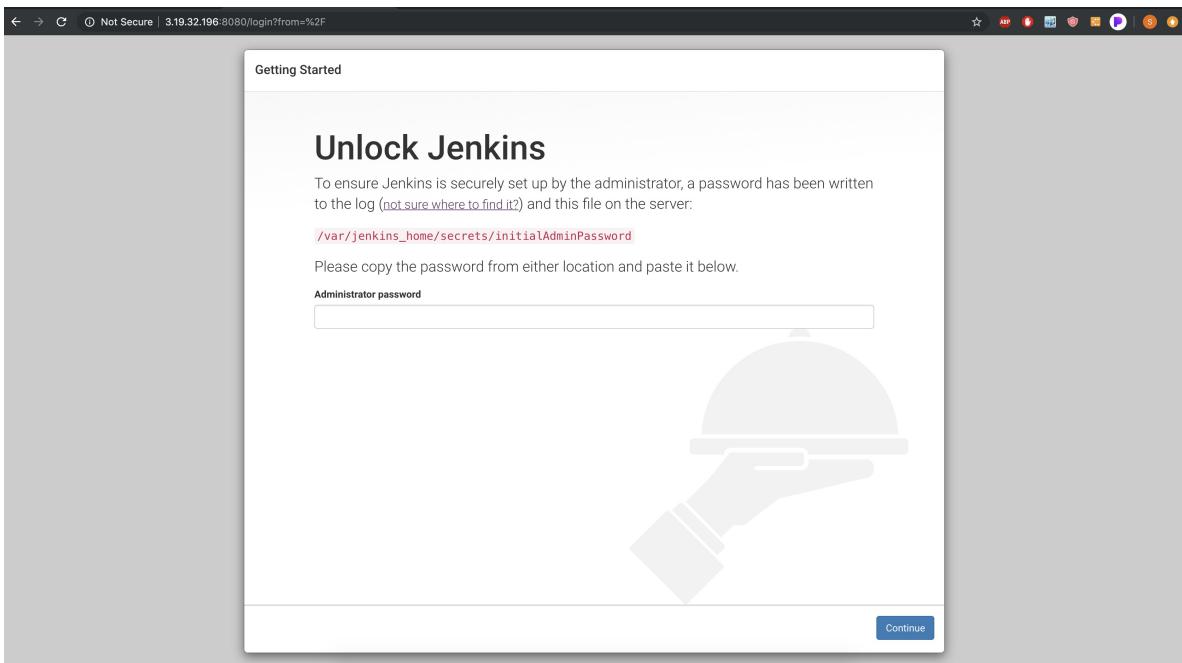
[Return to Table of Contents](#)

Setting up Jenkins for DevOps Pipeline (Microcosm)

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

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

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



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

Last Event Time	Log Stream
2019-04-16 13:27 UTC-6	ecs/jenkins/92c59a81-6a6a-45e0-bc7b-393298dfa9
2019-04-16 13:24 UTC-6	ecs/jenkins/cd06962c-ac89-49e5-b514-70cf1d84af06
2019-04-16 13:18 UTC-6	ecs/jenkins/234135b5-3ad6-40a5-a8bc-2c96a5977dd3
2019-04-16 13:12 UTC-6	ecs/jenkins/cb06a3e4-8354-4392-9abc-b577b00eacf
2019-04-16 13:06 UTC-6	ecs/jenkins/4c39ca9d-512e-43e9-a0f0-49f61b146d87
2019-04-16 13:01 UTC-6	ecs/jenkins/23da544d-46a1-4c1f-bcfe-59439fd398b3
2019-04-16 12:55 UTC-6	ecs/jenkins/1ba1470-602c-4503-8ec6-d2b7c934d8c6
2019-04-16 12:49 UTC-6	ecs/jenkins/67338b8-e9f6-462b-99ec-00c1ca93b0d4
2019-04-16 12:43 UTC-6	ecs/jenkins/ef11b1d9-3bcf-4c4e-9efe-1178adb17ed0
2019-04-16 12:38 UTC-6	ecs/jenkins/ab79c1e-377a-4b08-b3a4-479bece8e8a4
2019-04-16 12:36 UTC-6	ecs/jenkins/a0e70a0-ed61-4778-8a5d-e77fe096b7f5
2019-04-16 12:35 UTC-6	ecs/jenkins/30dd0d420-f7a4-4a1f-b81f-5ab001dde223
2019-04-16 12:34 UTC-6	ecs/jenkins/7ea768-2a5d-4170-a077-66a3ba95635d
2019-04-16 12:32 UTC-6	ecs/jenkins/1927ead4-fbb8-4db8-b942-870140cc4ced
2019-04-16 12:31 UTC-6	ecs/jenkins/12c54ba2-90d8-416d-ab59-c718a7624384
2019-04-16 12:30 UTC-6	ecs/jenkins/63329022-7683-488a-a1f9-4ac01cd0f3
2019-04-16 12:29 UTC-6	ecs/jenkins/02182562-c89e-4287-a4df-ae69b0f53a2c
2019-04-16 12:27 UTC-6	ecs/jenkins/f6f630-8228-403f-b4dd-c4916ba2d35
2019-04-16 12:26 UTC-6	ecs/jenkins/81ed7740-ea0e-4087-9954-55b447afea3
2019-04-16 12:25 UTC-6	ecs/jenkins/30da76c7-0dc8-4448-9e93-9ee15e385485
2019-04-16 12:10 UTC-6	ecs/jenkins/1542226d-fba0-4bf4-824b-a13b16e4d686
2019-04-16 11:53 UTC-6	ecs/jenkins/0771fa1c-47f5-43ba-a0c-8a5a5a61a720
2019-04-16 10:36 UTC-6	ecs/jenkins/c272e68f-c302-4bb1-8610-cee6cc63c6c3
2019-04-16 15:59 UTC-6	ecs/jenkins/0be7fe7b-eb7d-43c3-9076-9bad78e689f

- 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 Event Viewer interface. The left sidebar includes links for CloudWatch, Dashboards, Alarms, Events, Rules, Event Buses, Logs (which is selected), Insights, Metrics, and Favorites. The main pane displays log events for the date 2019-04-16. The logs show the startup of a Spring Framework application context, the creation of an admin user, and the download of the latest update center data file.

```

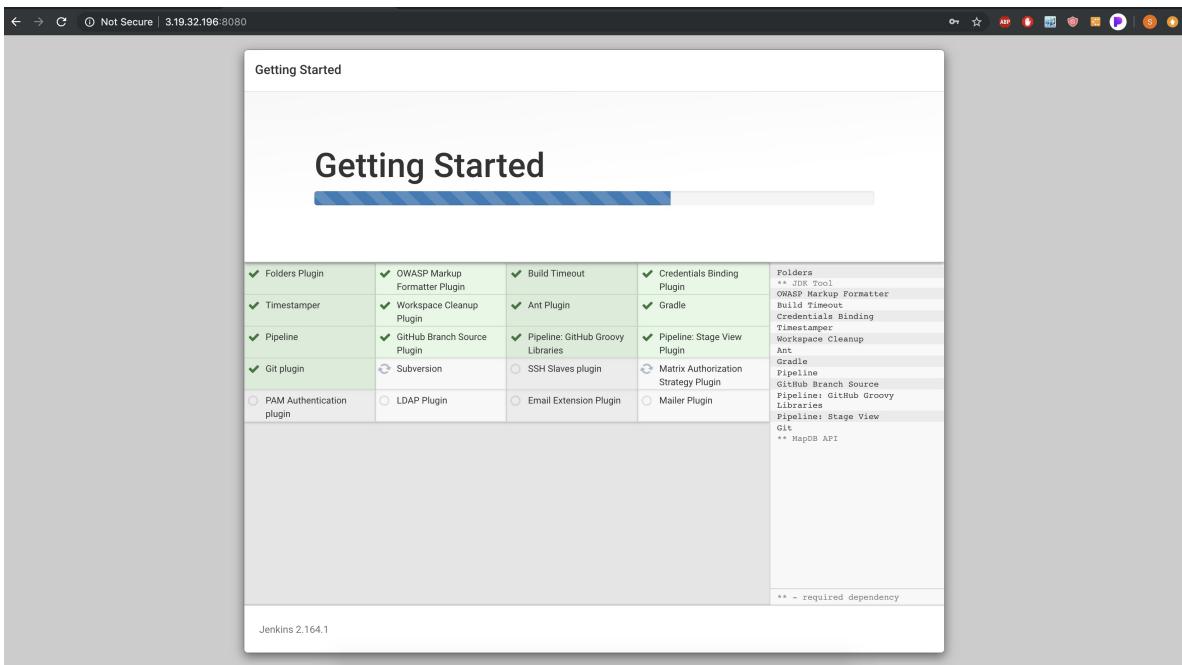
Time (UTC +0:00) Message
2019-04-16
19:28:31 Apr 16, 2019 7:28:31 PM org.springframework.context.support.AbstractApplicationContext prepareRefresh
19:28:31 INFO: Refreshing org.springframework.web.context.support.StaticWebApplicationContext@1b118f44; display name [Root WebApplicationContext]; startup date [Tue Apr 16 19:28:31 UTC 2019]
19:28:31 Apr 16, 2019 7:28:31 PM org.springframework.context.support.AbstractApplicationContext obtainFreshBeanFactory
19:28:31 INFO: Bean factory for application context [org.springframework.web.context.support.StaticWebApplicationContext@1b118f44]: org.springframework.beans.factory.support.DefaultListableBeanFactory
19:28:31 Apr 16, 2019 7:28:31 PM org.springframework.beans.factory.support.DefaultListableBeanFactory preInstantiateSingletons
19:28:31 INFO: Pre-instantiating singletons in org.springframework.beans.factory.support.DefaultListableBeanFactory@4a15ce02; defining beans [authenticationManager]; root of factory hierarchy
19:28:32 Apr 16, 2019 7:28:32 PM org.springframework.context.support.AbstractApplicationContext prepareRefresh
19:28:32 INFO: Refreshing org.springframework.web.context.support.StaticWebApplicationContext@33803935; display name [Root WebApplicationContext]; startup date [Tue Apr 16 19:28:32 UTC 2019]
19:28:32 Apr 16, 2019 7:28:32 PM org.springframework.context.support.AbstractApplicationContext obtainFreshBeanFactory
19:28:32 INFO: Bean factory for application context [org.springframework.web.context.support.StaticWebApplicationContext@33803935]: org.springframework.beans.factory.support.DefaultListableBeanFactory
19:28:32 Apr 16, 2019 7:28:32 PM org.springframework.beans.factory.support.DefaultListableBeanFactory preInstantiateSingletons
19:28:33 INFO: Pre-instantiating singletons in org.springframework.beans.factory.support.DefaultListableBeanFactory@271503c9; defining beans [filter,legacy]; root of factory hierarchy
19:28:33 Apr 16, 2019 7:28:33 PM jenkins.install.SetupWizard init
19:28:33 INFO:
19:28:33 ****
19:28:33 ****
19:28:33 Jenkins initial setup is required. An admin user has been created and a password generated.
19:28:33 Please use the following password to proceed to installation:
19:28:33 47dbd8eaae244d55b030d2cdad78db
19:28:33
19:28:33 This may also be found at: /var/jenkins_home/secrets/initialAdminPassword
19:28:33 ****
19:28:33 ****
19:28:33 ****
19:28:33 ****
19:28:40 --> setting agent port for http...
19:28:41 --> setting agent port for http... done
19:28:48 Apr 16, 2019 7:28:48 PM hudson.model.UpdateSite updateData
19:28:48 INFO: Obtained the latest update center data file for UpdateSource default
19:28:48

```

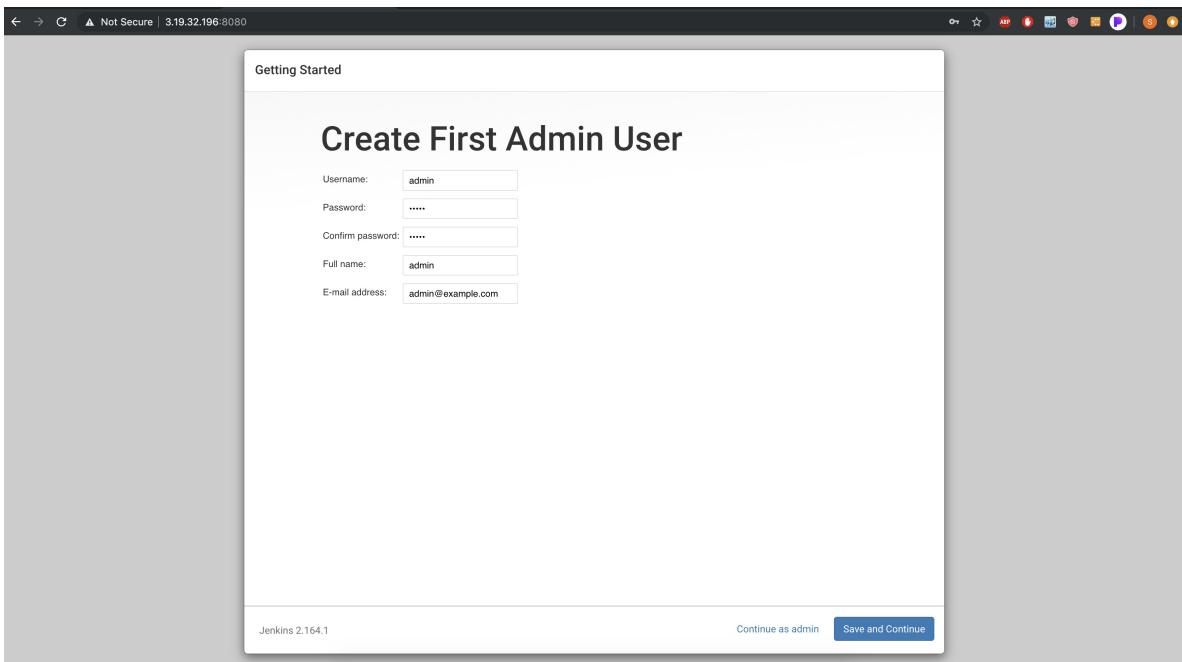
- Click Install suggested plugins

The screenshot shows the Jenkins 'Getting Started' dialog box. It contains two main options: 'Install suggested plugins' and 'Select plugins to install'. Both options are described as ways to customize Jenkins with additional features. The dialog box also includes a note about the Jenkins version (2.164.1).

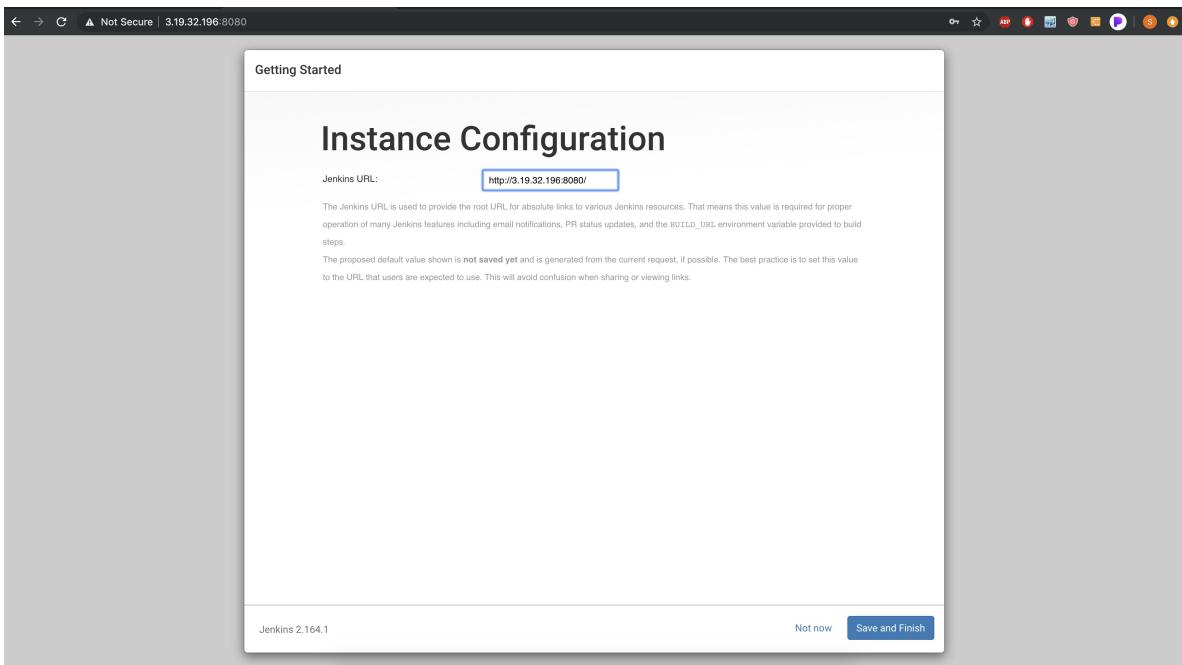
- Wait while the plugins are installed



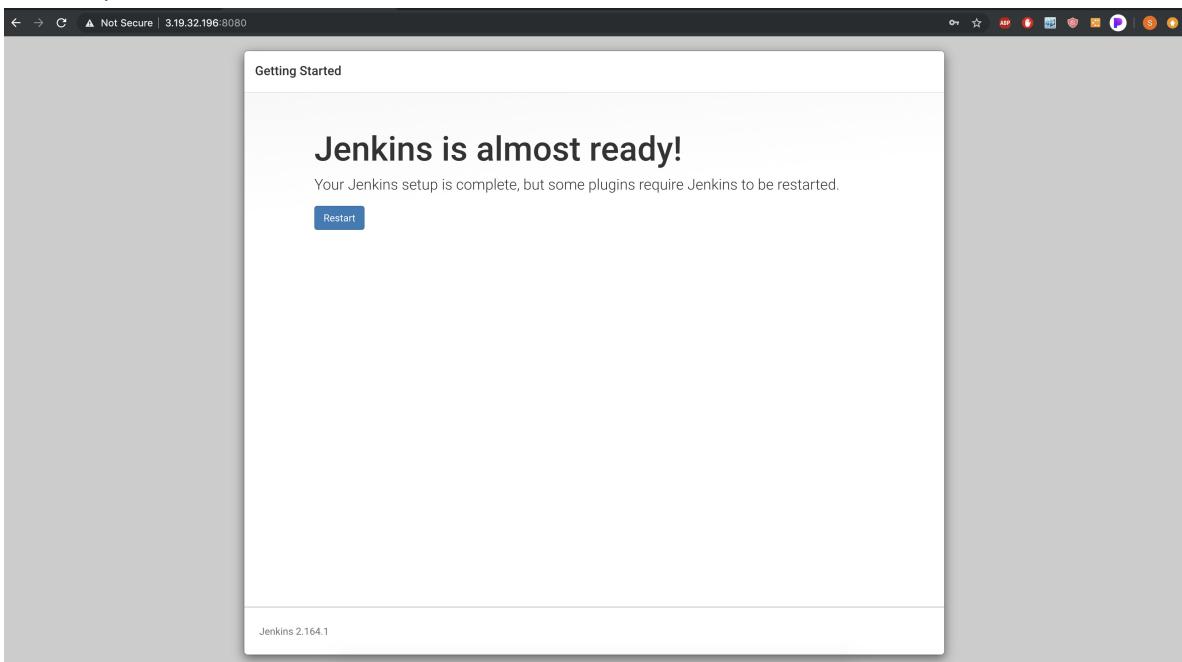
- Fill in the information for the admin user. Make sure to remember your password.
- Select Save and Continue



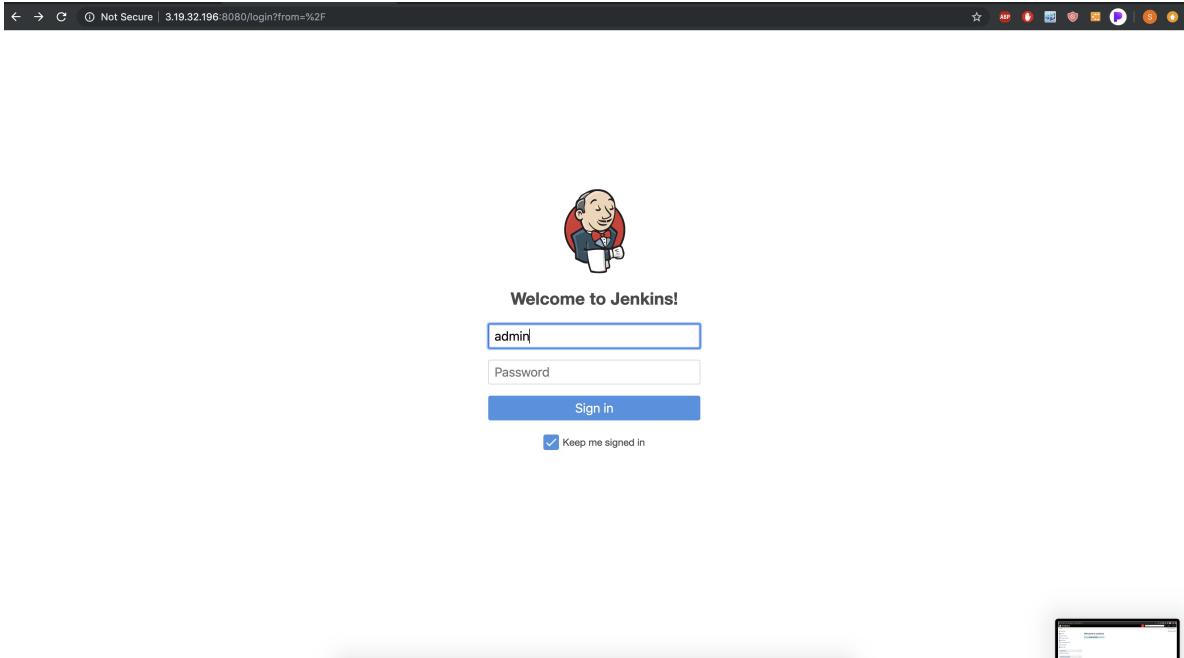
- Select Save and Finish



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



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



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

New version of Jenkins (2.164.2) is available for [download](#) ([changelog](#)).

Warnings have been published for the following currently installed components.

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

Lockable Resources plugin 2.4:
XSS vulnerability

Script Security Plugin 1.55:
Script Security sandbox bypass

Pipeline: Groovy 2.64:
Script Security sandbox bypass

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

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

The screenshot shows the Jenkins Plugin Manager interface. At the top, there's a navigation bar with links for 'Back to Dashboard' and 'Manage Jenkins'. Below that is a search bar and user information ('admin | I log out'). The main area has tabs for 'Updates', 'Available' (which is selected), 'Installed', and 'Advanced'. A 'Filter:' input field is at the top right. The 'Available' tab displays a list of plugins categorized under 'Install'. The categories shown are: 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, Pipeline: Declarative Extension Points API. Each plugin entry includes a checkbox, a brief description, and columns for 'Name', 'Version', and 'Installed'. Buttons at the bottom include 'Download now and install after restart', 'Update information obtained: 46 min ago', and 'Check now'.

- 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"

This screenshot shows the same Jenkins Plugin Manager interface as the previous one, but with a different set of plugins selected in the 'Available' tab. The selected plugins are: SLOCCount, SonarQube Scanner, Sonargraph Integration, Sonargraph, Spelunk, Statistics Gatherer, StepCounter, Summary Display, TAP, Task Scanner, Tattletale, Test Results Analyzer, Test stability history, Testability Explorer, and TestComplete support. The interface is identical to the first screenshot, with the same tabs, filter, and bottom buttons.

- 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 links for 'Back to Dashboard', 'Manage Jenkins', and 'Manage Plugins'. The main content area is titled 'Installing Plugins/Upgrades'. It includes a 'Preparation' section with three bullet points: 'Checking internet connectivity', 'Checking update center connectivity', and 'Success'. Below this is a table of 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), SonarCube Scanner (Pending), Extended Choice Parameter (Pending), Custom Tools (Pending), Selenium HTML report (Pending), and Restarting Jenkins (Pending). At the bottom, there are two buttons: 'Go back to the top page' and 'Restart Jenkins when installation is complete and no jobs are running'.

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

The screenshot shows the Jenkins Manage Jenkins page. On the left, there's a sidebar with links for 'New Item', 'People', 'Build History', 'Manage Jenkins' (which is selected), 'My Views', 'Lockable Resources', 'Credentials', and 'New View'. The main content area is titled 'Manage Jenkins' and displays a message: 'New version of Jenkins (2.164.2) is available for download (changelog)'. Below this, there's a section for 'Warnings have been published for the following currently installed components'. It lists several items with their respective vulnerabilities:

- 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 options with icons: 'Configure System' (gear icon), 'Configure Global Security' (padlock icon), 'Configure Credentials' (key icon), and '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. Under the 'Maven' section, there is a 'Name' field containing 'petclinic'. Below it, the 'Install automatically' checkbox is checked. At the bottom of the page are 'Save' and 'Apply' buttons.

- Click Apply
- Click Save

The screenshot shows the Jenkins Global Tool Configuration page. Under the 'Custom tool' section, there is a 'Name' field containing 'ZAP_2.6.0'. Below it, the 'Install automatically' checkbox is checked. The 'Extract *.zip*.tar.gz' and 'Label' fields are present. The 'Download URL for binary archive' field contains 'https://github.com/zaproxy/zaproxy/releases/download/2.6.0/ZAP_2.6.0_Linux.tar.gz'. The 'Subdirectory of extracted archive' field contains 'ZAP_2.6.0'. At the bottom are 'Add Installer' and 'Delete Custom tool' buttons. The 'Save' and 'Apply' buttons are also visible.

Slack & Hubot integration with Jenkins

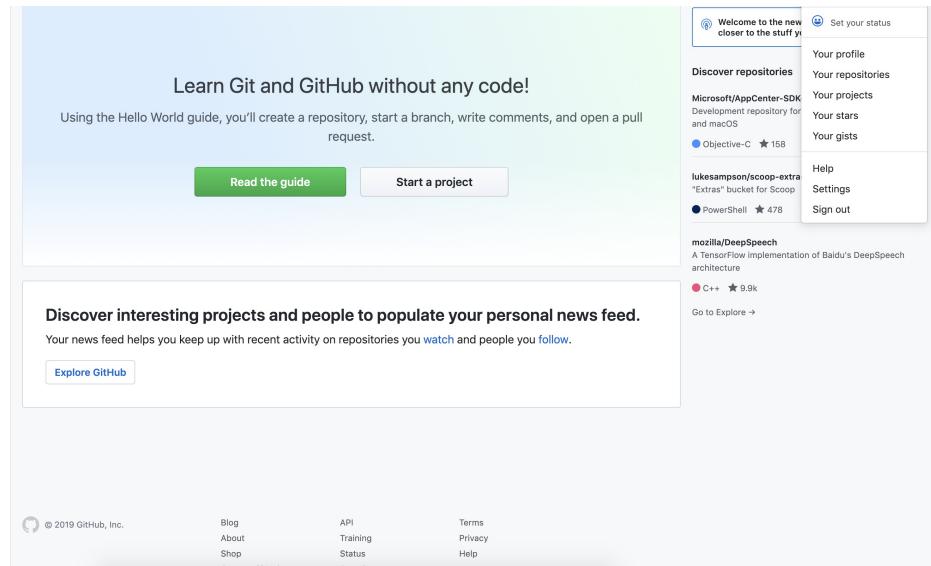
[Return to Table of Contents](#)

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



- Select Developer settings from the left menu list

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

- Select Person Access Tokens
- Click Generate new token

The screenshot shows the GitHub developer settings interface. The top navigation bar includes links for Pull requests, Issues, Marketplace, and Explore. Below the navigation is a search bar and a notifications icon. The main content area is titled "Personal access tokens" and contains a sub-section for "Personal access tokens". It includes a "Generate new token" button and a note explaining that personal access tokens function like OAuth access tokens. The sidebar on the left lists "OAuth Apps", "GitHub Apps", and "Personal access tokens". At the bottom, there's a footer with copyright information and links to 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 "New personal access token" creation form. The top navigation bar and sidebar are identical to the previous screenshot. The main form has a "Token description" field containing "microcosm_class" and a "Select scopes" section. The "Select scopes" section lists various OAuth scopes with checkboxes:

- repo**: Full control of private repositories
- repo:status**: Access commit status
- repo_deployment**: Access deployment status
- public_repo**: Access public repositories
- repo:invite**: Access repository invitations

Below these are sections for "admin:org", "admin:public_key", "admin:repo_hook", "admin:org_hook", and "gist", each with its own set of checkboxes and descriptions.

- 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.

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

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

A sample Spring-based application

429 commits | 6 branches | 0 releases | 17 contributors

This branch is even with SLS-ALL:master.

File	Description	Date
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)



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.

Change your password

1 error prohibited this user from being saved:

- Password is too short (minimum is 8 characters)

New password
.....

Confirm new password
.....

Change your password

Didn't receive a confirmation email? [Request a new one](#)
Already have login and password? [Sign in](#)

Explore Help About GitLab

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



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.

Invalid Login or password.

Sign in [Register](#)

Username or email
root

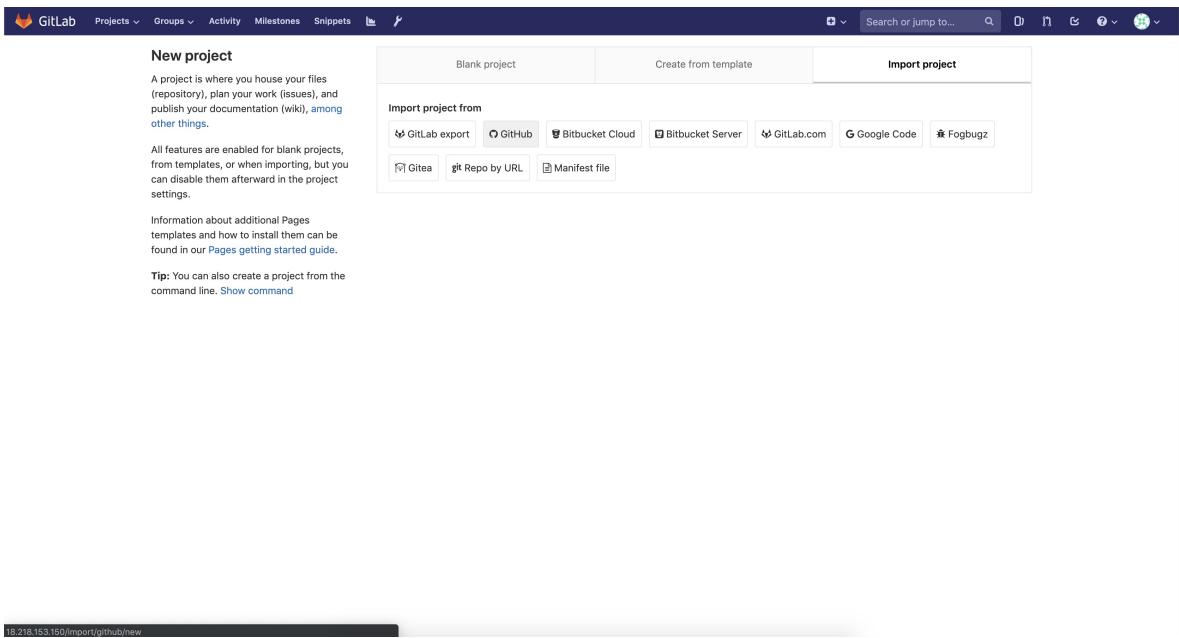
Password
.....

Remember me [Forgot your password?](#)

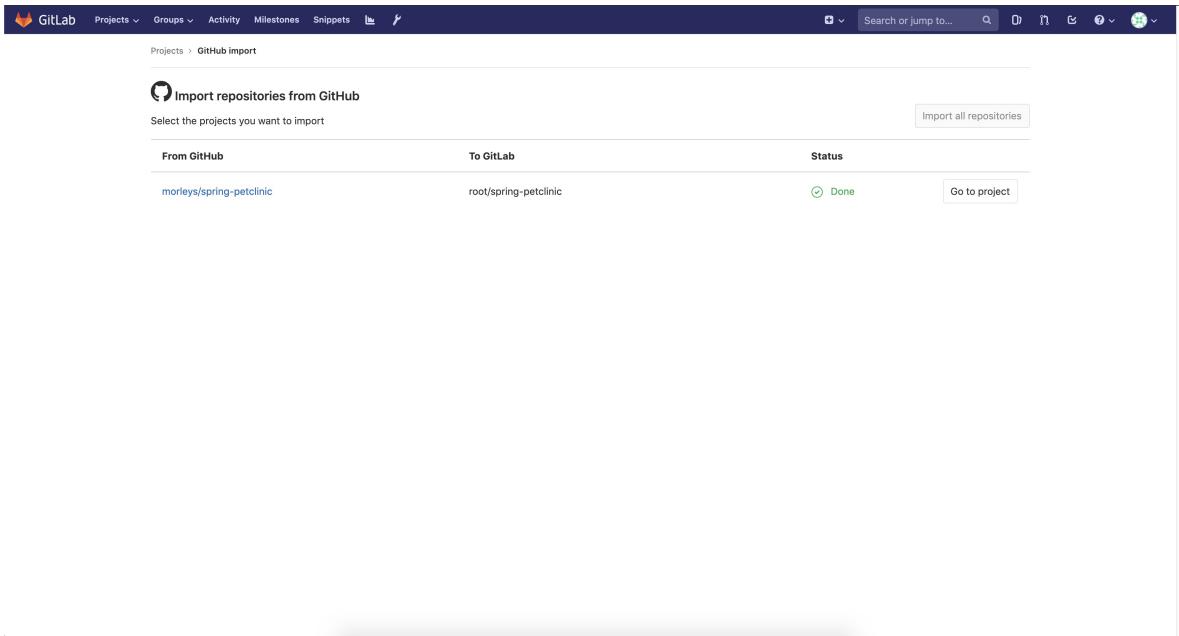
Sign in

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'



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



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

IMPORTANT

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

Create Maven Deployment in Jenkins

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

-  [New Item](#)
-  [People](#)
-  [Build History](#)
-  [Project Relationship](#)
-  [Check File Fingerprint](#)

-  Manage Jenkins
-  My Views
-  Lockable Resources
-  Credentials
-  New View

Build Queue

No builds in the queue.

Build Executor Status

1 Idle

2 Idle

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

Enter an item name

» Required field

 **Freestyle project**

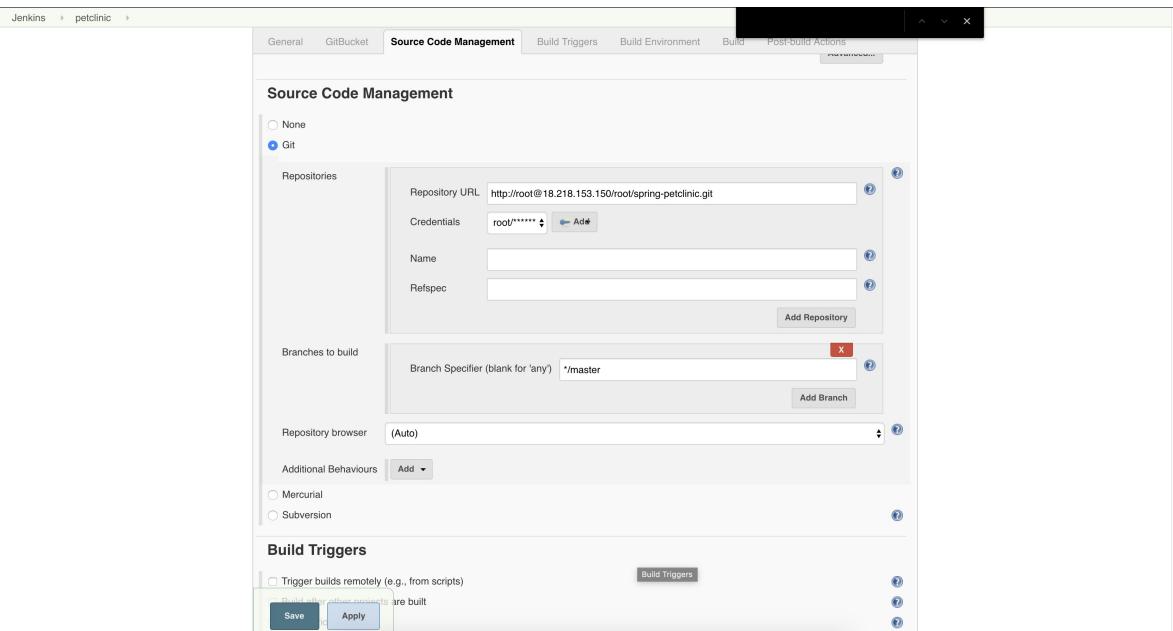
This is the central feature of Jenkins. Jenkins will build your project, combining any SCM with any build system, and this can be even used for something other than software build.

 **Maven project**

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

- Under Source Code Management, select 'git'
- Beside Credentials, click Add -> Jenkins
- Select "Username with password"
- Enter your GitLab credentials (see 'gitlab' VM instructions above) and click Add
- Enter repository URL: `http://[username@gitlab VM private network IP]/spring-petclinic.git`

- NOTE: this is the HTTP URL from the GitLab project page where 'localhost' is replaced by the 'gitlab' VM's private network IP (ex: <http://root@10.1.1.3/root/spring-petclinic.git>)
- Select appropriate credentials
- Leave the default build Root Pom pointing to pom.xml
- Select APPLY, the SAVE



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

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

Continuous Code Quality

[Log in](#) [Read documentation](#)

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

Fix The Leak

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](#)

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

Welcome to SonarQube!

Want to quickly analyze a first project? Follow these 2 easy steps.

[Skip this tutorial](#) [Find it back anytime in the Help section](#)

1 Provide a token

Generate a token

[Generate](#)

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.

2 Run analysis on your project

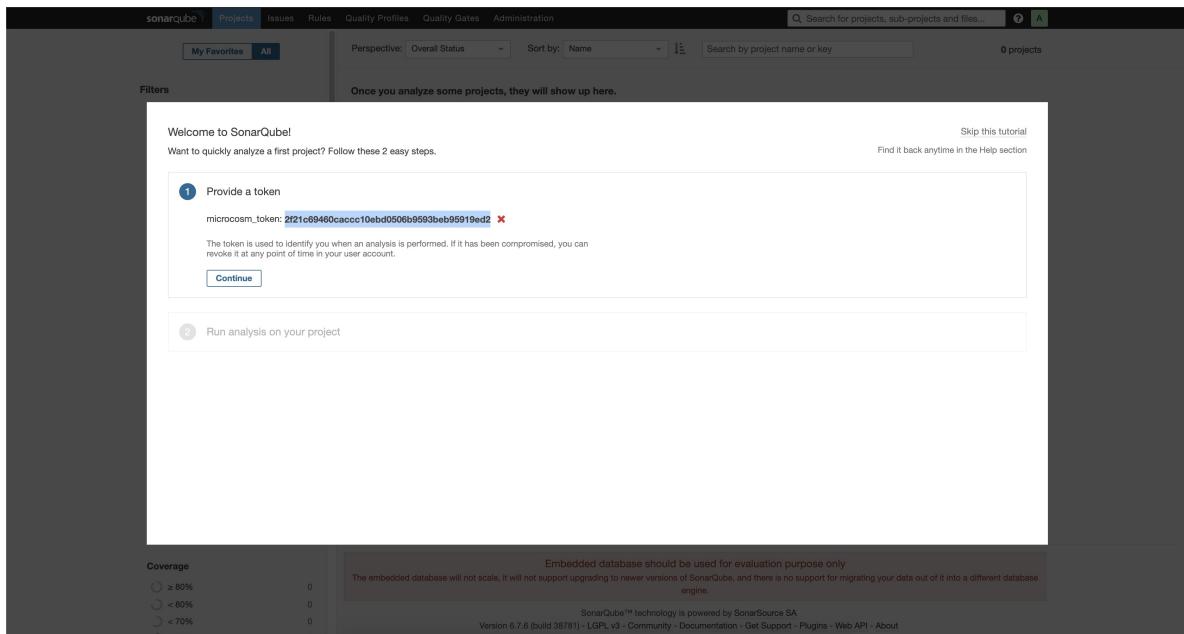
Coverage

<input type="radio"/>	≥ 80%	0
<input type="radio"/>	< 80%	0
<input type="radio"/>	< 70%	0

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.

SonarQube™ technology is powered by SonarSource SA
Version 6.7.6 (build 39781) - 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

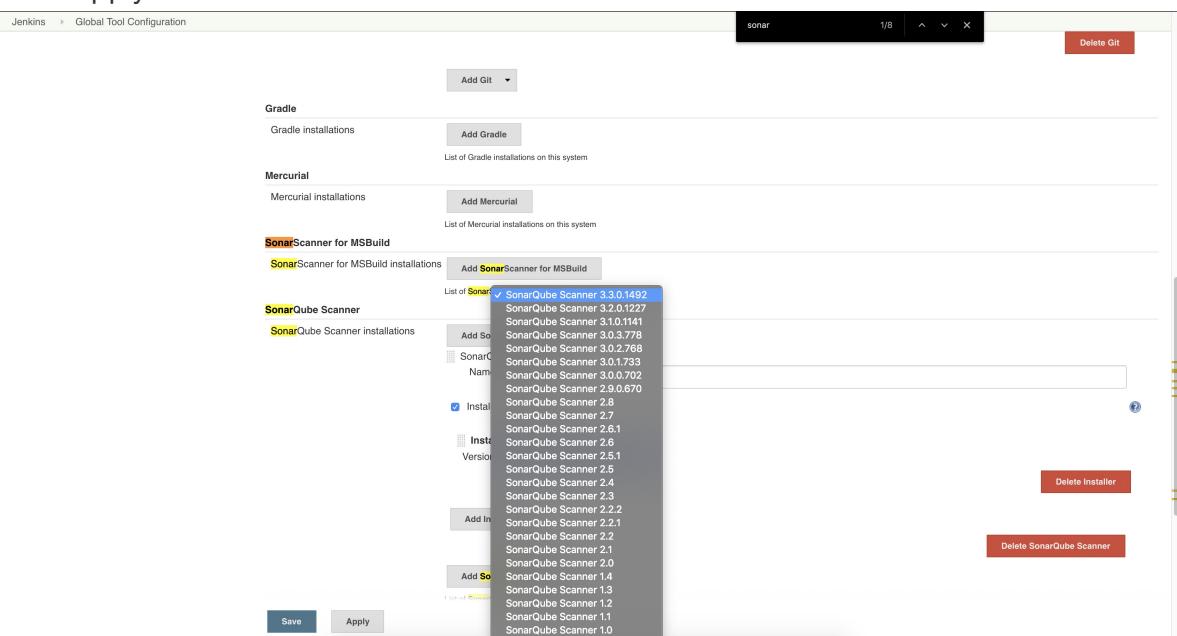


- Return to Jenkins
- From Manage Jenkins -> Under Configure System, scroll down to SonarQube Servers
- Select Add SonarQube
- Enter the name SonarQube
- Paste you server authentication token
- Enter the url inclusing port of your SonarQube server
- Press APPLY, then SAVE

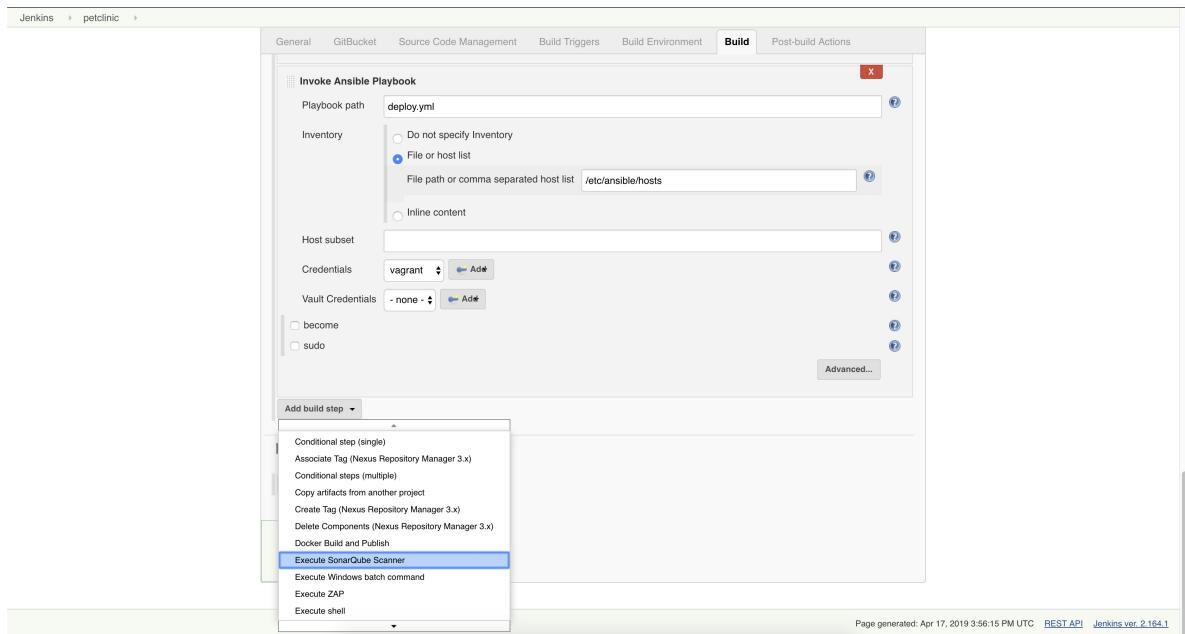
Jenkins > configuration

Build Queue	# of executors	sonar
No builds in the queue.	2	1/6
Build Executor Status	Labels	x
1 Idle	Usage	Use this node as much as possible
2 Idle	Quiet period	5
	SCM checkout retry count	0
	<input type="checkbox"/> Restrict project naming	
Global properties	<input type="checkbox"/> Disable deferred wipeout on this node	
	<input type="checkbox"/> Environment variables	
	<input type="checkbox"/> Prepare jobs environment	
	<input type="checkbox"/> Tool Locations	
SonarQube servers	<input type="checkbox"/> Enable injection of SonarQube server configuration as build environment variables	
Environment variables	If checked, job administrators will be able to inject a SonarQube server configuration as environment variables in the build.	
SonarQube installations	Name	SonarQube
	Server URL	<code>http://18.222.11.148:9000</code>
	Server authentication token	Default is <code>http://localhost:9000</code>
	Version of sonar-maven-plugin	SonarQube authentication token. Mandatory when anonymous access is disabled.
	Additional arguments	If not specified, the goal will be <code>sonar:sonar</code> .
	Additional analysis properties	Additional command line arguments to be passed to the SonarQube scanner. For example, <code>-X</code> .
		Additional analysis properties in the form of key-value pairs. For example, <code>sonar.analysis.mode=issues</code> .
	Save	Apply

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



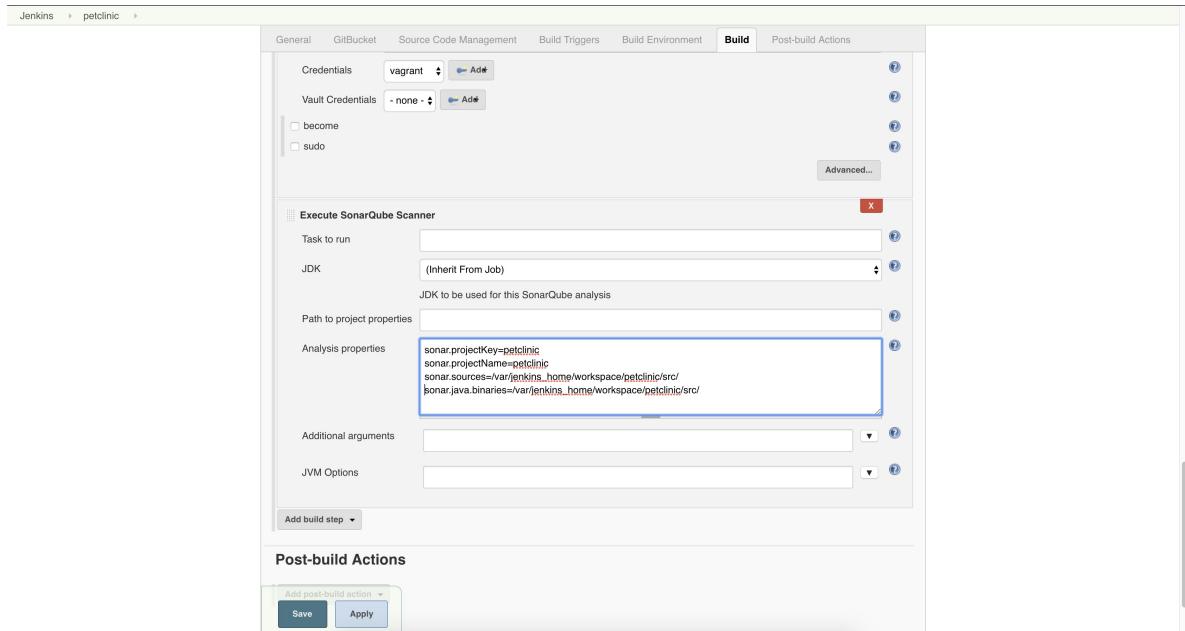
- Return to the PetClinic configuration screen
- Scroll down to the Post Steps Section
- Click "Add build step" and select "Execute SonarQube Scanner"



- 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"

sonarqube Projects Issues Rules Quality Profiles Quality Gates Administration

Perspective: Overall Status Sort by: Name Search by project name or key 1 projects

My Favorites All

Filters

Quality Gate

Passed	1
Warning	0
Failed	0

Reliability (Bug)

A	1
B and worse	0
C and worse	0
D and worse	0
E	0

Security (Vulnerabilities)

A	1
B and worse	0
C and worse	0
D and worse	0
E	0

Maintainability (Code Smells)

A	1
B and worse	0
C and worse	0
D and worse	0
E	0

Coverage

≥ 80%	0
< 80%	0
< 70%	0
< 60%	0

Last analysis: April 17, 2019, 11:23 AM
1 of 1 shown

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.

SonarQube™ technology is powered by SonarSource SA
Version 6.7.6 (build 38781) - LGPLv3 - Community - Documentation - Get Support - Plugins - Web API - About

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

Roles

Dashboard Groups Users **Roles** Policies Identity providers Account settings Credential report

Encryption keys

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 user

Belonging to you

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

Choose the service that will use this role

EC2

Allows EC2 instances to call AWS services on your behalf.

Lambda

Allows Lambda functions to call AWS services on your behalf.

API Gateway

CodeDeploy

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

Select your use case

CodeDeploy

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

CodeDeploy - ECS

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

CodeDeploy for Lambda

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

- Press Next

Create role

1

2

3

4

Attached permissions policies

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

Policy name	Used as	Description
▶ AWSCodeDeployRole	None	Provides CodeDeploy service access to expand...

- Add Tags if desired and press Next

Create role

1 2 3 4

Add tags (optional)

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

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

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

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

[Permissions](#) [Trust relationships](#) [Tags \(1\)](#) [Access Advisor](#) [Revoke sessions](#)

▼ Permissions policies (1 policy applied)

[Attach policies](#)

Policy name ▾
▶  AWSCodeDeployRole

- Search for, and select the following policies. Once all are selected press Attach Policy. (Note that there is a 10 policy limit per role)
 - AmazonEC2FullAccess
 - AutoScalingFullAccess
 - AWSDataLifecycleManagerServiceRole
 - ElasticLoadBalancingFullAccess
 - AmazonS3FullAccess
 - CloudWatchLogsFullAccess
- By way of auditing the use of the role, and to verify that you haven't assigned more access than is necessary, selecting the Access Advisor tab after selecting the desired role, will display what access has actually been used by the role.

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 behalf
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** **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 Windows

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 service 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 service 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, and all of the other software required to run web-based 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? 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
Show less

Select

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

Tomcat Certified by Bitnami

 Free tier eligible	<p>Tomcat Certified by Bitnami</p> <p>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 ...</p> <p>More info</p> <p>View Additional Details in AWS Marketplace</p>	<p>Pricing Details</p> <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th colspan="4">Hourly Fees</th> </tr> <tr> <th>Instance Type</th> <th>Software</th> <th>EC2</th> <th>Total</th> </tr> </thead> <tbody> <tr><td>t2.micro</td><td>\$0.00</td><td>\$0.012</td><td>\$0.024/hr</td></tr> <tr><td>t2.small</td><td>\$0.00</td><td>\$0.023</td><td>\$0.046/hr</td></tr> <tr><td>t2.medium</td><td>\$0.00</td><td>\$0.046</td><td>\$0.092/hr</td></tr> <tr><td>t2.large</td><td>\$0.00</td><td>\$0.093</td><td>\$0.186/hr</td></tr> <tr><td>t2.xlarge</td><td>\$0.00</td><td>\$0.186</td><td>\$0.371/hr</td></tr> <tr><td>t2.2xlarge</td><td>\$0.00</td><td>\$0.371</td><td>\$0.742/hr</td></tr> <tr><td>t3.micro</td><td>\$0.00</td><td>\$0.01</td><td>\$0.02/hr</td></tr> <tr><td>t3.small</td><td>\$0.00</td><td>\$0.021</td><td>\$0.042/hr</td></tr> <tr><td>t3.medium</td><td>\$0.00</td><td>\$0.042</td><td>\$0.083/hr</td></tr> <tr><td>t3.large</td><td>\$0.00</td><td>\$0.083</td><td>\$0.166/hr</td></tr> <tr><td>t3.xlarge</td><td>\$0.00</td><td>\$0.166</td><td>\$0.333/hr</td></tr> <tr><td>t3.2xlarge</td><td>\$0.00</td><td>\$0.333</td><td>\$0.666/hr</td></tr> <tr><td>m4.large</td><td>\$0.00</td><td>\$0.10</td><td>\$0.20/hr</td></tr> <tr><td>m4.xlarge</td><td>\$0.00</td><td>\$0.20</td><td>\$0.40/hr</td></tr> <tr><td>m4.2xlarge</td><td>\$0.00</td><td>\$0.40</td><td>\$0.80/hr</td></tr> <tr><td>m4.4xlarge</td><td>\$0.00</td><td>\$0.80</td><td>\$1.60/hr</td></tr> <tr><td>m4.10xlarge</td><td>\$0.00</td><td>\$2.00</td><td>\$4.00/hr</td></tr> <tr><td>m4.16xlarge</td><td>\$0.00</td><td>\$3.20</td><td>\$6.40/hr</td></tr> <tr><td>c5.large</td><td>\$0.00</td><td>\$0.085</td><td>\$0.17/hr</td></tr> <tr><td>c5.xlarge</td><td>\$0.00</td><td>\$0.17</td><td>\$0.34/hr</td></tr> <tr><td>c5.2xlarge</td><td>\$0.00</td><td>\$0.34</td><td>\$0.68/hr</td></tr> </tbody> </table>	Hourly Fees				Instance Type	Software	EC2	Total	t2.micro	\$0.00	\$0.012	\$0.024/hr	t2.small	\$0.00	\$0.023	\$0.046/hr	t2.medium	\$0.00	\$0.046	\$0.092/hr	t2.large	\$0.00	\$0.093	\$0.186/hr	t2.xlarge	\$0.00	\$0.186	\$0.371/hr	t2.2xlarge	\$0.00	\$0.371	\$0.742/hr	t3.micro	\$0.00	\$0.01	\$0.02/hr	t3.small	\$0.00	\$0.021	\$0.042/hr	t3.medium	\$0.00	\$0.042	\$0.083/hr	t3.large	\$0.00	\$0.083	\$0.166/hr	t3.xlarge	\$0.00	\$0.166	\$0.333/hr	t3.2xlarge	\$0.00	\$0.333	\$0.666/hr	m4.large	\$0.00	\$0.10	\$0.20/hr	m4.xlarge	\$0.00	\$0.20	\$0.40/hr	m4.2xlarge	\$0.00	\$0.40	\$0.80/hr	m4.4xlarge	\$0.00	\$0.80	\$1.60/hr	m4.10xlarge	\$0.00	\$2.00	\$4.00/hr	m4.16xlarge	\$0.00	\$3.20	\$6.40/hr	c5.large	\$0.00	\$0.085	\$0.17/hr	c5.xlarge	\$0.00	\$0.17	\$0.34/hr	c5.2xlarge	\$0.00	\$0.34	\$0.68/hr
Hourly Fees																																																																																														
Instance Type	Software	EC2	Total																																																																																											
t2.micro	\$0.00	\$0.012	\$0.024/hr																																																																																											
t2.small	\$0.00	\$0.023	\$0.046/hr																																																																																											
t2.medium	\$0.00	\$0.046	\$0.092/hr																																																																																											
t2.large	\$0.00	\$0.093	\$0.186/hr																																																																																											
t2.xlarge	\$0.00	\$0.186	\$0.371/hr																																																																																											
t2.2xlarge	\$0.00	\$0.371	\$0.742/hr																																																																																											
t3.micro	\$0.00	\$0.01	\$0.02/hr																																																																																											
t3.small	\$0.00	\$0.021	\$0.042/hr																																																																																											
t3.medium	\$0.00	\$0.042	\$0.083/hr																																																																																											
t3.large	\$0.00	\$0.083	\$0.166/hr																																																																																											
t3.xlarge	\$0.00	\$0.166	\$0.333/hr																																																																																											
t3.2xlarge	\$0.00	\$0.333	\$0.666/hr																																																																																											
m4.large	\$0.00	\$0.10	\$0.20/hr																																																																																											
m4.xlarge	\$0.00	\$0.20	\$0.40/hr																																																																																											
m4.2xlarge	\$0.00	\$0.40	\$0.80/hr																																																																																											
m4.4xlarge	\$0.00	\$0.80	\$1.60/hr																																																																																											
m4.10xlarge	\$0.00	\$2.00	\$4.00/hr																																																																																											
m4.16xlarge	\$0.00	\$3.20	\$6.40/hr																																																																																											
c5.large	\$0.00	\$0.085	\$0.17/hr																																																																																											
c5.xlarge	\$0.00	\$0.17	\$0.34/hr																																																																																											
c5.2xlarge	\$0.00	\$0.34	\$0.68/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 vCPU, memory, and storage resources. 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 Instances: 2
 - Network: Select your Student VPC
 - Subnet: Select either subnet
 - Auto-Assign public IP: Select Enable (or leave the default if it is Enabled)
 - Under Advanced Detail, copy and paste the contents of the

`addCodeDeploytoEC2Image.sh` file in the optional text box. Make sure the As Text radio button is selected. The purpose of this script is to add the libraries and services necessary to the Ubuntu based VM to be able to deploy to the machine in an automated fashion. Note that the script uses the AWS EAST-2 region (Ohio), change this portion of the S3 url for the region you are using. Additionally, this script installs

the services necessary to forward logs to CloudWatch to be used for monitoring and debugging deployments, and program runs. [ref](#).

- Press Next

Step 3: Configure Instance Details

Subnet: subnet-04d355bb02dc06049 | tst1 | us-east-2a | Create new subnet
248 IP Addresses available

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

Placement group: Add instance to placement group

Capacity Reservation: Open | Create new Capacity Reservation

IAM role: EC2_Plus_S3 | Create new IAM role

Shutdown behavior: Stop

Enable termination protection: Protect against accidental termination

Monitoring: Enable CloudWatch detailed monitoring
Additional charges apply.

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

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

T2/T3 Unlimited: Enable
Additional charges may apply

Network interfaces

Device	Network Interface	Subnet	Primary IP	Secondary IP addresses	IPv6 IPs
eth0	New network interface	subnet-04d355bl	Auto-assign	Add IP	Add IP

Advanced Details

User data: As text | As file | Input is already base64 encoded

(Optional)

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

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	Value	Instances	Volumes
Name	Code_Deploy_Instances	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>

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

Step 6: Configure Security Group

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

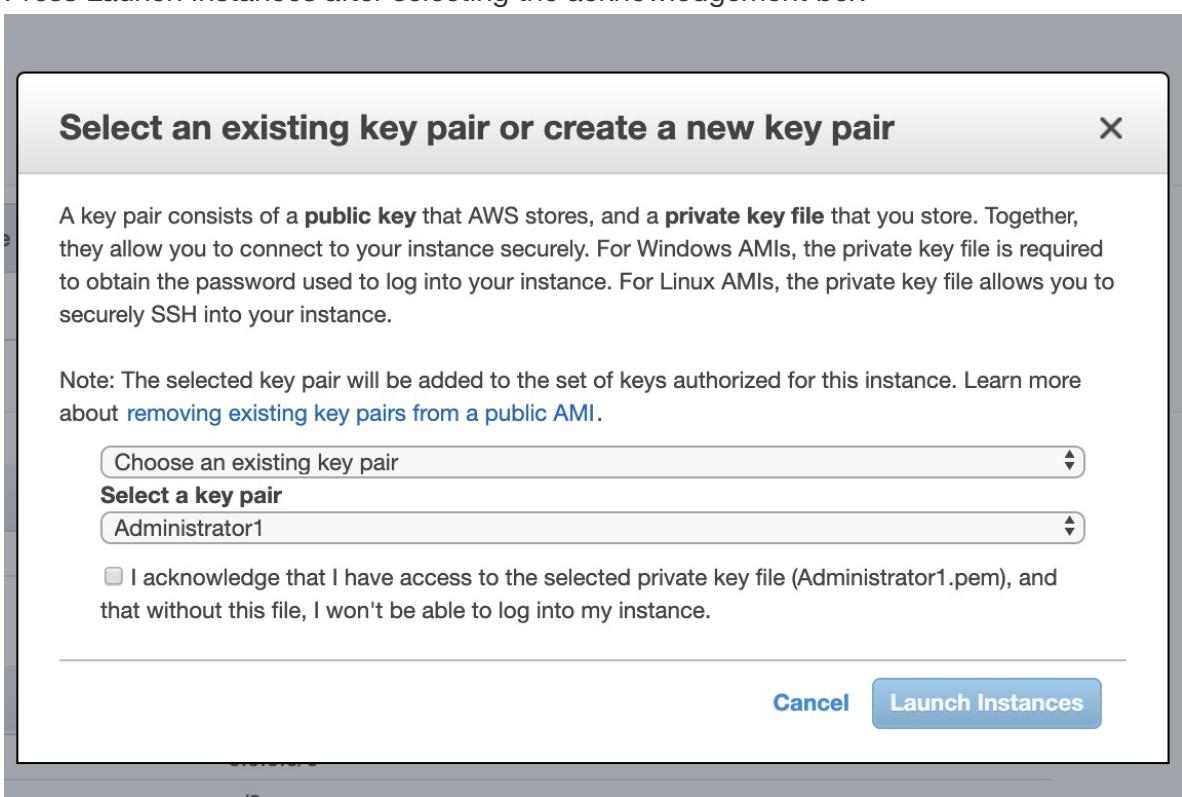
Assign a security group: Create a new security group
 Select an existing security group

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

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

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

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



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

Actions	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	Secure
Launch Instance	Code_Deploy...	i-03a155360dafb37b5	t2.large	us-east-2a	running	Initializing	None	18.218.184.15	-	-	Administrator1	disabled	April 26, 2019 at 9:14:08 AM...	default
Connect	Code_Deploy...	i-0e2a50d45d9bd604	t2.large	us-east-2a	running	Initializing	None	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	ec2betclinic	<small>⚠ Load balancer names must only contain alphanumeric characters or hyphens, and not start with a hyphen.</small>
Scheme	<input checked="" type="radio"/> internet-facing <input type="radio"/> internal	
IP address type	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	vpc-0eb311ea035a4eb9c (10.0.0.0/16) tst
Availability Zones	<input checked="" type="checkbox"/> us-east-2a subnet-04d355bb02dc06049 (tst1) <small>IPv4 address Assigned by AWS</small>
	<input checked="" type="checkbox"/> us-east-2c subnet-0c9b55ecae32d1073 (tst2) <small>IPv4 address Assigned by AWS</small>

Tags

- Press Next for step 2

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

Step 2: Configure Security Settings

⚠ Improve your load balancer's security. Your load balancer is not using any secure listener.
If your traffic to the load balancer needs to be secure, use the HTTPS protocol for your front-end connection. You can go back to the first step to change the protocol.

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

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

Step 3: Configure Security Groups

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

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

Security Group ID	Name	Description
sg-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[New target group](#)**Name**

TGCodedeployec2petclinic

⚠ TargetGroup name cannot contain characters that are not letters, numbers, or underscores.**Target type** Instance IP Lambda function**Protocol**

HTTP

Port

8080

Health checks

Protocol

HTTP

Path

/petclinic

▼ Advanced health check settings

Port traffic port override**Healthy threshold**

5

Unhealthy threshold

2

Timeout

5

seconds

Interval

30

seconds

Success codes

200

- Step 5, Select both instances of your AMI, press Add to Registered on port 8080
- 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.

[Remove](#)

Instance	Name	Port	State	Security groups	Zone
i-03a155360dafb37b5	Code_Deploy_Instances	8080	● running	default	us-east-2a
i-0e2a50cb45d6bd604	Code_Deploy_Instances	8080	● running	default	us-east-2a

Instances

To register additional instances, select one or more running instances, specify a port, and then click Add. The default port is the port specified for the target group. If the instance is already registered on the specified port, you must specify a different port.

[Add to registered](#) on port 8080

Instance	Name	State	Security groups	Zone	Subnet ID	Subnet CIDR
i-03a155360dafb37b5	Code_Deploy_Instances	● running	default	us-east-2a	subnet-04d355bb02dc06049	10.0.0.0/24
i-0e2a50cb45d6bd604	Code_Deploy_Instances	● running	default	us-east-2a	subnet-04d355bb02dc06049	10.0.0.0/24

- Step 6, Review and Press Create

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

Step 6: Review

Please review the load balancer details before continuing

▼ Load balancer

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

▼ Security groups

Security groups sg-0e1a263967d2471f2

▼ Routing

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

▼ Targets

Instances i-03a155360dafb37b5 (Code_Deploy_Instances):8080, i-0e2a50cb45d6bd604 (Code_Deploy_Instances):8080

Create S3 Bucket

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

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

S3 buckets

Search for buckets

+ Create bucket

Edit public access settings

Empty

Delete

Bucket name ▾

Access ⓘ ▾

Region ▾

cf-templates-1a5vqotkxza7-us-east-2

Objects can be public

US East (Ohio)

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)
- Remove public access granted through public ACLs (Recommended)

Manage public bucket policies for this bucket

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

Manage system permissions

Grant Amazon S3 Log Delivery group write access to this bucket

Create Codedeploy Application

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

- From the services menu at the top of the page select CodeDeploy
- Press Create Application

The screenshot shows the AWS CodeDeploy service page. On the left, there's a sidebar with 'Developer Tools' and 'CodeDeploy' selected. Under 'CodeDeploy', there are links for 'Source', 'Build', 'Deploy', 'Getting started', 'Deployments', 'Applications', 'Deployment configurations', 'On-premises instances', and 'Pipeline'. The main content area is titled 'AWS CodeDeploy' and says 'Automate code deployments to maintain application uptime'. It includes a description of AWS CodeDeploy and a 'Create AWS CodeDeploy deployment' button. Below the main content, there's a 'Pricing (US)' section and a 'See CodeDeploy in action' link.

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



Create application

Application configuration

Application name

Enter an application name

100 character limit

Compute platform

Choose a compute platform

[Cancel](#)

[Create application](#)

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

☰ Application

Application
petclinic

Compute type
EC2/On-premises

Deployment group name

Enter a deployment group name
petclinicDepGrp

100 character limit

Service role

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

CodeDeployServiceRole ▾

Deployment type

Choose how to deploy your application

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

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

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

Environment configuration

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

Amazon EC2 Auto Scaling groups

Amazon EC2 instances

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

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

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

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

Tag group 1

Key

Value - optional

<input type="text"/> Name	X	<input type="text"/> Code_Deploy_Instances	X	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](#)

Developer Tools **CodeDeploy**

Source + CodeCommit
Build + CodeBuild
Deploy + CodeDeploy
Getting started
Deployments
Applications **Application** petclinicDepGrp
Deployment configurations
On-premises instances
Pipeline + CodePipeline

Feedback
Return to the old experience

Success
Deployment group created

Developer Tools > CodeDeploy > Applications > petclinic > petclinicDepGrp

petclinicDepGrp

Deployment group details

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		

Environment configuration: Amazon EC2 instances

Key	Value
Name	Code_Deploy_Instances

Edit **Delete** **Create deployment**

Create Secret Key

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

- Navigate to the IAM Concole

Welcome to Identity and Access Management

IAM users sign-in link:

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

IAM Resources

Users: 2

Roles: 17

Groups: 2

Identity Providers: 1

Customer Managed Policies: 2

Security Status

- Select your user

[Add user](#)

[Delete user](#)

<input type="text"/> Find users by username or access key		Groups
<input type="checkbox"/>	User name ▾	
<input type="checkbox"/>	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

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

Path: /

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

Sign-in credentials

- Summary: Console sign-in link: <https://cert-microcosm.siginin.aws.amazon.com/console>
- Console password: Enabled (last signed in Today) | [Manage](#)
- Assigned MFA device: Not assigned | [Manage](#)
- Signing certificates: None [Edit](#)

Access keys

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

[Create access key](#)

Access key ID	Created	Last used
AKIAWOJKHRHJASUYR35N	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

Create access key

Success

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.

[Download .csv file](#)

Access key ID	Secret access key
AKIAWOJKHRHJASUYR35N	7ar8nUSG1C2gstL9x3H6qBf46ta202Man6/wRXF Hide

[Close](#)

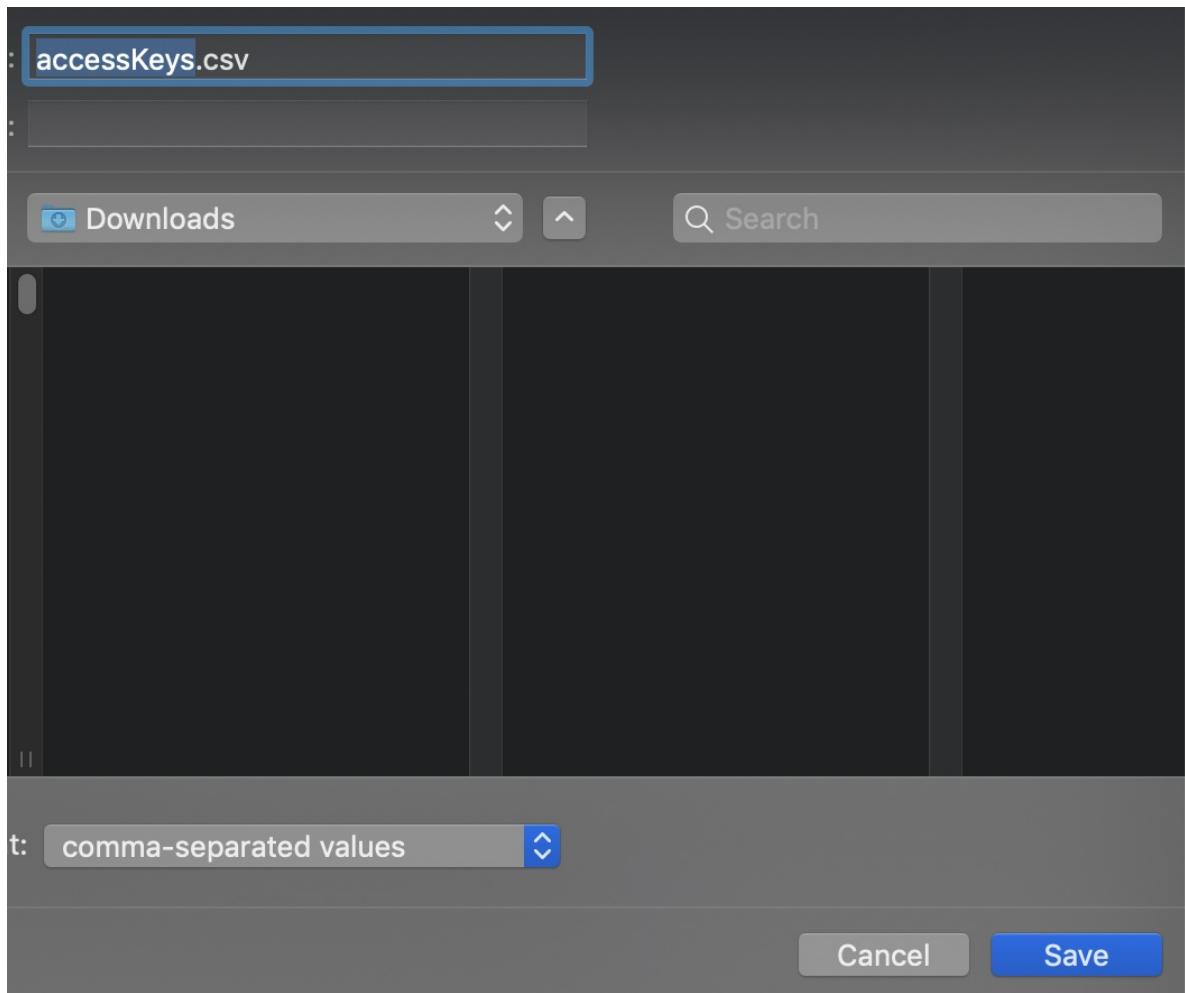
Access keys

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 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	<code>**/*.yml, **/*.html, **/*.yaml, **/scripts/*.* , **/target/*.war</code>
Exclude Files	<code>./src,/target</code>
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.	
Save	Apply
AWS Secret Key	

Build and Deploy

- From your Project Page in Jenkins select Build

 Back to Dashboard

 **Status**

 Changes

 Workspace

 Build Now

 Delete Maven project

 Configure

 Modules

 Favorite

 SonarQube

 Open Blue Ocean

 Rename

 Embeddable Build Status



Build History

trend 

find 

 #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

The screenshot shows the Amazon S3 console interface. At the top, the path is shown as Amazon S3 > petclinicdeploy > target. Below this, the word "Overview" is centered. A search bar contains the placeholder "Type a prefix and press Enter to search. Press ESC". Below the search bar are three buttons: "Upload" (with an upward arrow icon), "+ Create folder" (with a plus sign icon), and "Download" (with a downward arrow icon). Two files are listed in the main content area: "#43-3400450200434229806.zip" and "petclinic.war". Each file entry includes a checkbox and a small preview icon.

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

The screenshot shows the AWS CodeDeploy console under the "Deployments" section. The deployment ID is d-9GMQAH25Y. The "Deployment status" section shows "Installing application on your instances" and "0 of 2 instances updated" with a progress bar labeled "In progress". Below this are sections for "Deployment details" (Application: petclinic, Deployment ID: d-9GMQAH25Y, Deployment configuration: custom1, Deployment group: petclinicDepGrp) and "Revision details" (Revision location: s3://petclinicdeploy/target/#54-8254645117549694950.zip?, eTag: beff67ebed49a1f6645a1beefb4d5fb, Revision created: 1 minute ago, Description: Application revision registered via Jenkins). The final section is "Deployment lifecycle events", which lists two entries: "l-03a155360dafb37b5" and "l-0c2a50cb45d6bd604", both showing "In progress" status with "BeforeBlockTraffic" events at "Apr 26, 2019 11:11 AM".

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

Screenshot of the AWS CodeDeploy console showing the deployment of the petclinic application.

Deployment Summary:

Application	petclinic	Deployment ID	d-9GMQAH25Y	Status	In progress
Deployment configuration	custom1	Deployment group	petclinicDepGrp	Initiated by	user

Revision details:

Revision location	s3://petclinicdeploy/target/#54-8254645117549694950.zip eTag:beff67edee49a1f6645a1beefb4d5fb	Revision created	2 minutes ago	Description	Application revision registered via Jenkins
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	-	-	-

Deployment Details:

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

Revision details:

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	(S) Succeeded

Deployment details		
Application petclinic	Deployment ID d-61VZJA35Y	Status (S) 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	(S) Succeeded	AfterAllowTraffic	View events	Apr 26, 2019 3:51 PM	Apr 26, 2019 3:52 PM
i-0fcfb0d38fdf28923	1 minute 0 seconds	(S) Succeeded	AfterAllowTraffic	View events	Apr 26, 2019 3:51 PM	Apr 26, 2019 3:52 PM



Welcome



[Return to Table of Contents](#)

AWS ECS Based Code Deploy

Additional Task Definitions

Task/Service	Container			
--------------	-----------	--	--	--

Name	Name	Image	Ports	Volumes	Mount Po
petclinic	tomcat8	tomcat:8-jre8	8080	/webapps	[TODO path to a folder /opt/tomcat/wek
			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 defniniton
- 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

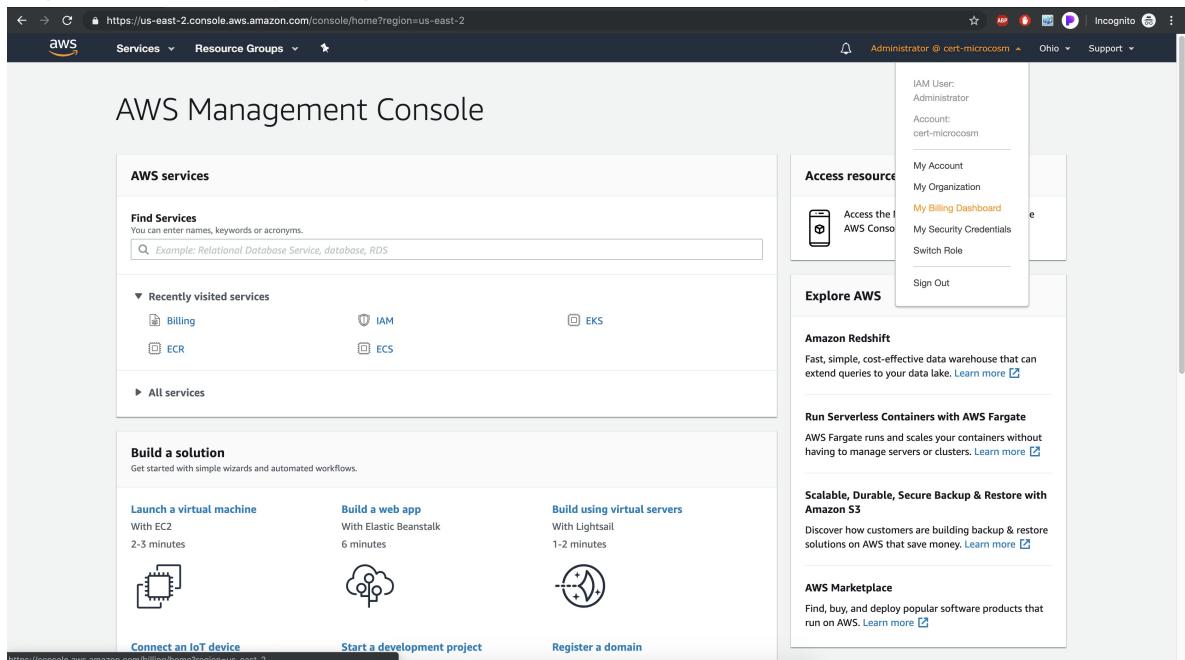
object key if the path within the bucket eg target/petclinic.war

[Return to Table of Contents](#)

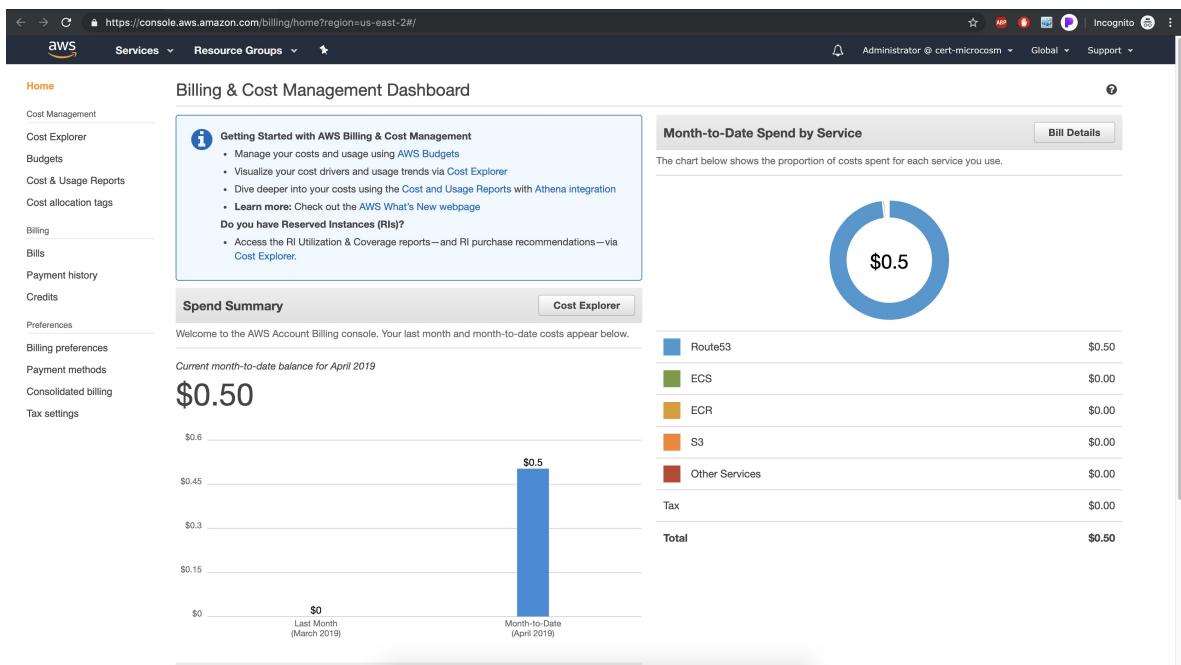
APPENDIX A

Billing

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



- 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	
<input type="button" value="Add"/> <input type="button" value="Cancel"/>	

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

Sonatype Nexus													
Nexus Repository Manager Servers													
<table border="1"> <thead> <tr> <th colspan="2">Nexus Repository Manager 2.x Server</th> </tr> </thead> <tbody> <tr> <td>Display Name</td> <td>Nexus</td> </tr> <tr> <td>Server ID</td> <td>hello1234567890bobmyfriend</td> </tr> <tr> <td>Server URL</td> <td>http://18.221.105.75:8081/nexus</td> </tr> <tr> <td>Credentials</td> <td>admin/***** <input type="button" value="Add"/></td> </tr> <tr> <td colspan="2">Nexus Repository Manager 2.x connection succeeded (2 hosted release Maven 2 repositories) <input type="button" value="Test connection"/></td> </tr> </tbody> </table>		Nexus Repository Manager 2.x Server		Display Name	Nexus	Server ID	hello1234567890bobmyfriend	Server URL	http://18.221.105.75:8081/nexus	Credentials	admin/***** <input type="button" value="Add"/>	Nexus Repository Manager 2.x connection succeeded (2 hosted release Maven 2 repositories) <input type="button" value="Test connection"/>	
Nexus Repository Manager 2.x Server													
Display Name	Nexus												
Server ID	hello1234567890bobmyfriend												
Server URL	http://18.221.105.75:8081/nexus												
Credentials	admin/***** <input type="button" value="Add"/>												
Nexus Repository Manager 2.x connection succeeded (2 hosted release Maven 2 repositories) <input type="button" value="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

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

Artifacts

Maven Artifact

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

Add Package ▾

Add post-build step ▾

Save Apply

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

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

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

Nexus Repository Manager OSS

Welcome

Sonatype™

Artifact Search

Advanced Search

Views/Repositories

Repositories

Help

Nexus Repository Manager OSS

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.

Get Started

- Configuration Set things up properly
- Documentation Visit our help site
- Community Ask and answer questions

Repository Formats

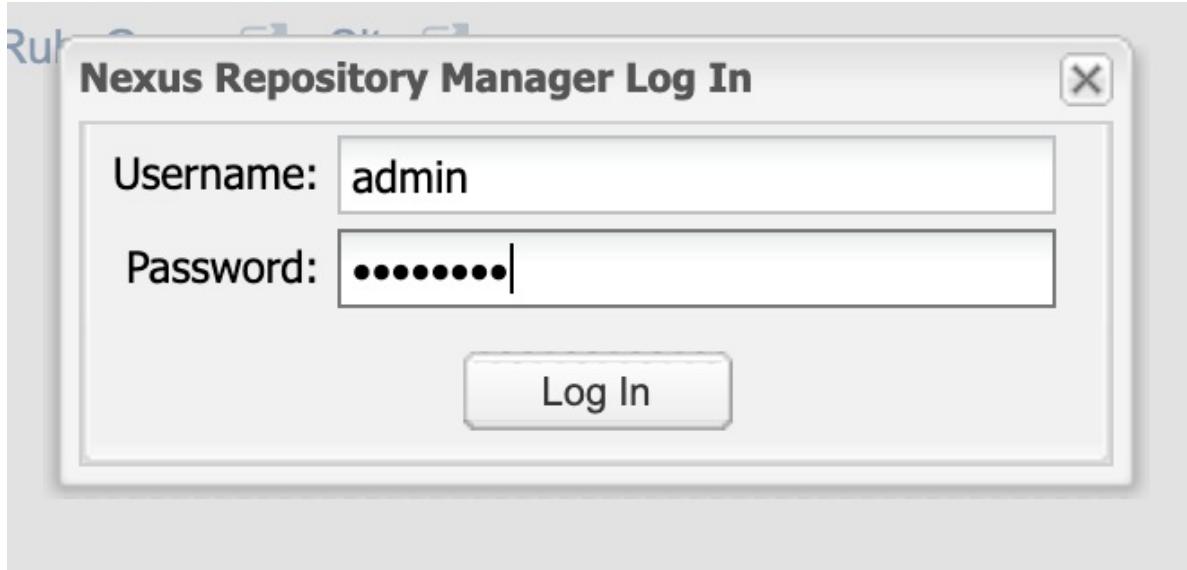
Maven · .NET/NuGet · Node/npm · OSGI · P2 · RPM/YUM · RubyGems · Site

Go Language Feedback

Are you using the Go language or thinking about it? Help us understand your needs for managing your Go modules and dependencies. Fill out the extended survey [here](#) and you can sign up to enter a prize drawing for a \$100 or a \$50 Amazon gift card.

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

A screenshot of the Sonatype Nexus interface. The left sidebar shows navigation options like "Artifactory", "Advanced Search", "Views/Repositories", "Security", "Administration", and "Help". The main area has tabs for "Welcome", "Repositories", and "Artifactory". Under "Repositories", there's a table showing various repositories: "Public Repositories" (group), "3rd party" (hosted), "Apache Snapshots" (proxy), "Central" (proxy), "Central M1 shadow" (virtual), "Releases" (hosted, highlighted in grey), and "Snapshots" (hosted). The "Releases" row shows details: Type "hosted", IQ Policy Violations "ANALYZE", Format "maven2", Policy "Release", Status "In Service", and Path "http://18.221.105.75:8081/nexus/content/repositories/releases". Below this, a "Releases" tab is active, showing a file tree with "Releases" (containing "petclinic-main" and "archetype-catalog.xml").

Repository	Type	IQ Policy Violations	Health Check	Format	Policy	Repository Status	Repository Path
Public Repositories	group	ANALYZE	maven2	Release	In Service		http://18.221.105.75:8081/nexus/content/groups/public
3rd party	hosted	ANALYZE	maven2	Snapshot	In Service		http://18.221.105.75:8081/nexus/content/repositories/thirdparty
Apache Snapshots	proxy	ANALYZE	maven2	Snapshot	In Service		http://18.221.105.75:8081/nexus/content/repositories/apache-snapshots
Central	proxy	ANALYZE	maven2	Release	In Service		http://18.221.105.75:8081/nexus/content/repositories/central
Central M1 shadow	virtual	ANALYZE	maven1	Release	In Service		http://18.221.105.75:8081/nexus/content/shadow/central-m1
Releases	hosted	ANALYZE	maven2	Release	In Service		http://18.221.105.75:8081/nexus/content/repositories/releases
Snapshots	hosted	ANALYZE	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 IAM Home page. On the left, there's a navigation sidebar with links like Dashboard, Groups, Users, Roles, Policies, Identity providers, Account settings, Credential report, and Encryption keys. The main area displays the 'Welcome to Identity and Access Management' section, which includes a 'Sign-in link' for IAM users, IAM Resources (Users: 1, Groups: 1, Roles: 9, Customer Managed Policies: 0, Identity Providers: 1), and a 'Security Status' section with four items: 'Activate MFA on your root account' (checked), 'Create individual IAM users' (checked), 'Use groups to assign permissions' (checked), and 'Apply an IAM password policy' (checked). A progress bar indicates '4 out of 4 complete.' To the right, there's a 'Feature Spotlight' section with a video thumbnail titled 'Introduction to AWS IAM' and an 'Additional Information' section with links to IAM best practices, documentation, and other resources.

- Select Users on the Left

- Click Add User

The screenshot shows the AWS IAM Users page. The left sidebar has a 'Users' link selected. The main area shows a table with one result for 'smorley'. 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.

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](#)

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

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

- 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

Add user

Set permissions

Add user to an existing group or create a new one. Using groups is a best-practice way to manage user's permissions by job functions. [Learn more](#)

Add user to group

[Create group](#) [Refresh](#)

Group	Attached policies
admin	AdministratorAccess

Showing 1 result

[Set permissions boundary](#)

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

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

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

- 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

Group name: Administrators

Create policy Refresh

Filter policies administrator Showing 10 results

	Policy name	Type	Used as	Description
<input checked="" type="checkbox"/>	AdministratorAccess	Job function	Permissions policy (2)	Provides full access to AWS services and resources.
<input type="checkbox"/>	AmazonAPIGatewayAdministrator	AWS managed	None	Provides full access to create/edit/delete APIs in Amazon API Gateway via the AWS Manag...
<input type="checkbox"/>	AWSAppSyncAdministrator	AWS managed	None	Provides administrative access to the AppSync service, though not enough to access via th...
<input type="checkbox"/>	AWSCloud9Administrator	AWS managed	None	Provides administrator access to AWS Cloud9.
<input type="checkbox"/>	AWSSSODirectoryAdministrator	AWS managed	None	Administrator access for SSO Directory
<input type="checkbox"/>	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...

Set permissions boundary Cancel Previous Next: Tags

- Select the created group

- Press Next:Tags

Add user to group

Create group Refresh

Q Search Showing 2 results

Group	Attached policies
Administrators	AdministratorAccess
admin	AdministratorAccess

Set permissions boundary Cancel Previous Next: Tags

- Add tags if desired

- Click Review

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

Review your choices. After you create the user, you can view and download the autogenerated password and access key.

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

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

- 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't copy this key now, you'll need to generate a new key or create a new user later.
- Press close

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

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

[Download .csv](#)

User	Access key ID	Secret access key	Email login instructions
Administrator	AKIAWOKHRHJCDYDAKNE	ZNHgDgrWBqnBR+K2AxGn csul7m2SIHOPJk4yqt5 Hide	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.

User ARN: arn:aws:iam::44300707681:user/Administrator

Path: /

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

Security credentials

Summary	Console sign-in link: https://cert-microcosm.sigin.aws.amazon.com/console
Console password	Enabled (last signed in Today) Manage
Assigned MFA device	Not assigned Manage
Signing certificates	None Edit

Access keys

Create access key

Access key ID	Created	Last used	Status
AKIAWOKHRHJCDYDAKNE	2019-04-10 10:04 MDT	2019-04-15 16:13 MDT with servicediscovery in us-east-2	Active Make inactive Delete

[Return to Table of Contents](#)

Create Key Pair

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

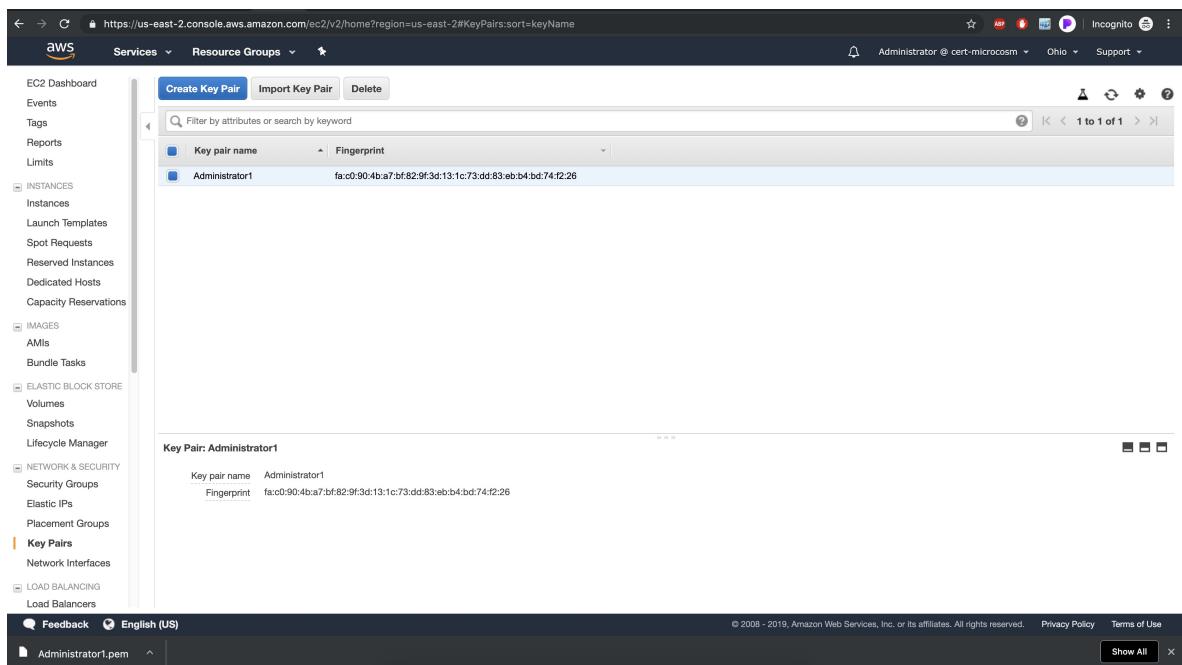
- From the EC2 Service Dashboard, select Key Pairs under Network & security on the left
- Select Create Key Pair

The screenshot shows the AWS EC2 dashboard with the 'Key Pairs' section selected. The main area displays a message: 'You do not have any Key Pairs in this region.' Below it is a button labeled 'Create Key Pair'. On the left sidebar, under 'Key Pairs', there is a sub-section for 'Network Interfaces'.

- 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 'Create Key Pair' dialog box overlaid on the EC2 dashboard. The 'Key pair name:' field contains the value 'Administrator1'. At the bottom right of the dialog are 'Cancel' and 'Create' buttons.

- View key pair details



[Return to Table of Contents](#)

Creating Roles

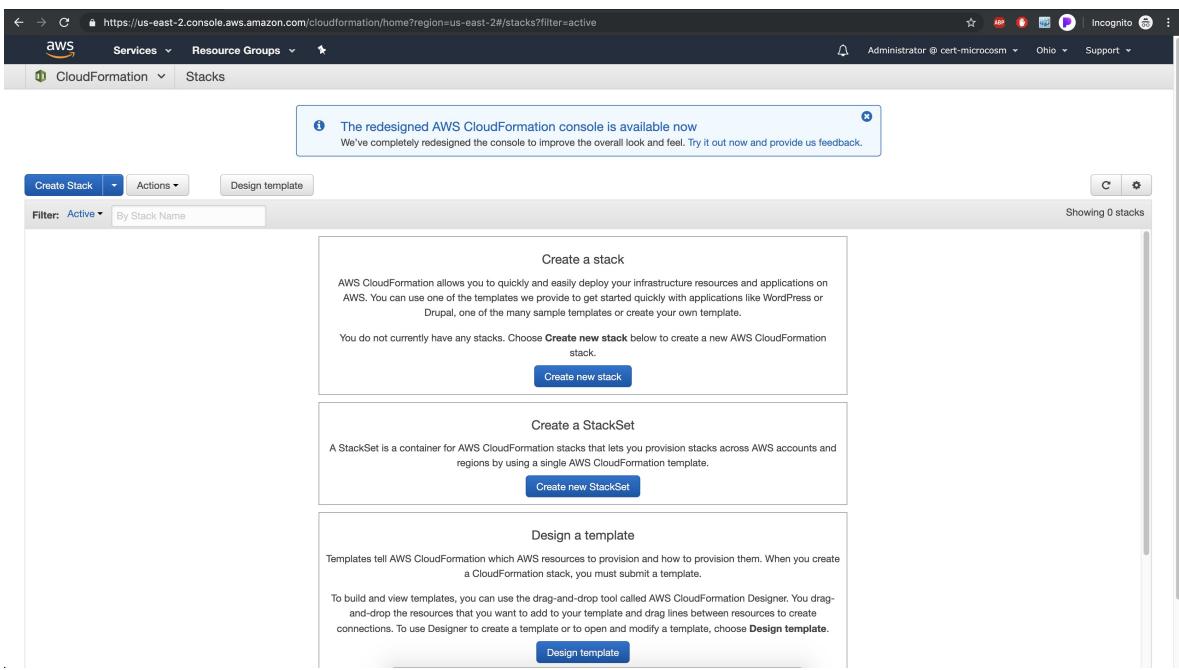
See the instructions in the [Create IAM Role for EC2 Based CodeDeploy](#) section.

[Return to Table of Contents](#)

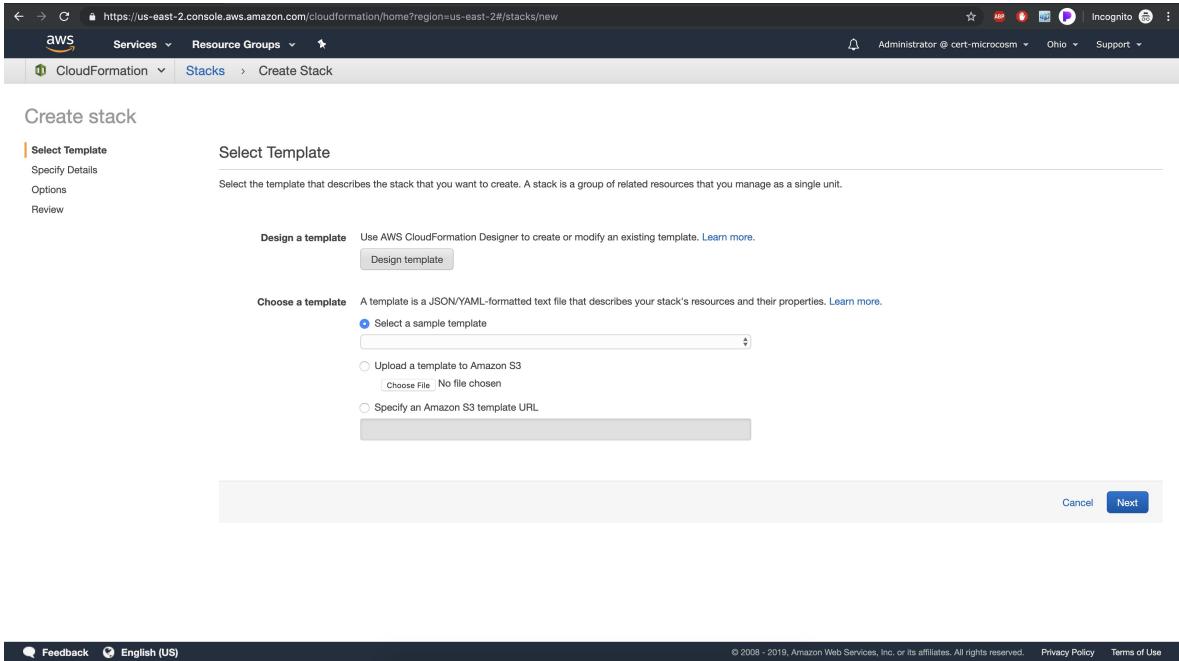
Create VPCs and user Accounts using CloudFormation Templates

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

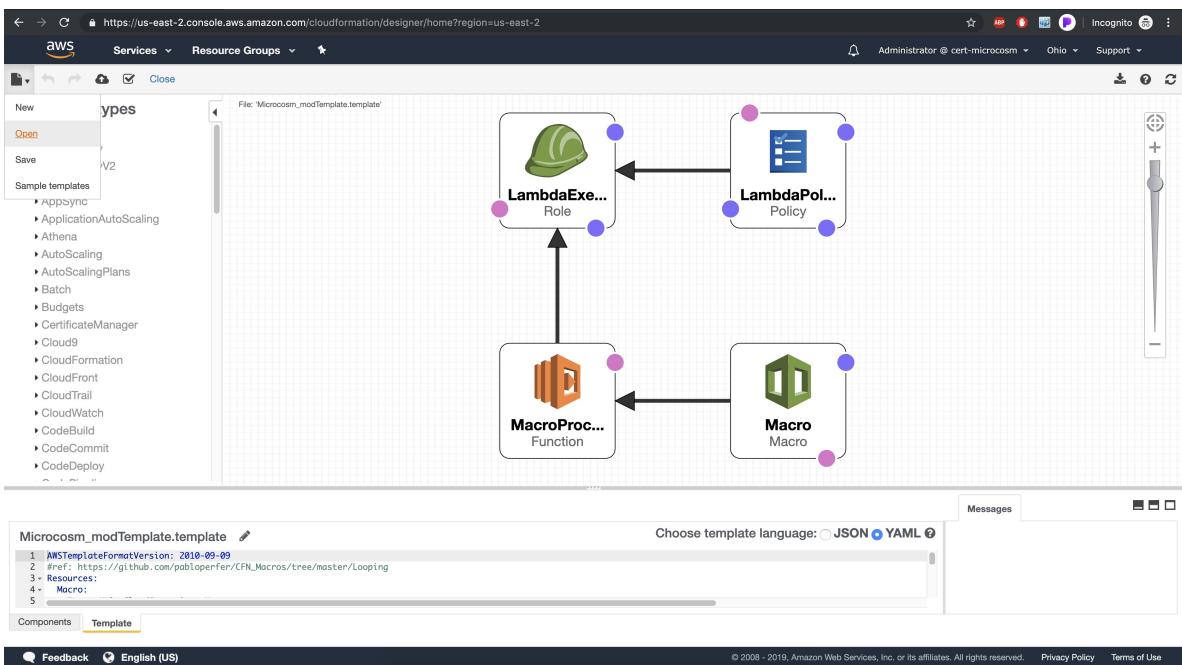
- Using the Services dropdown at the top on most AWS pages, navigate to CloudFormation. and select Create Stack



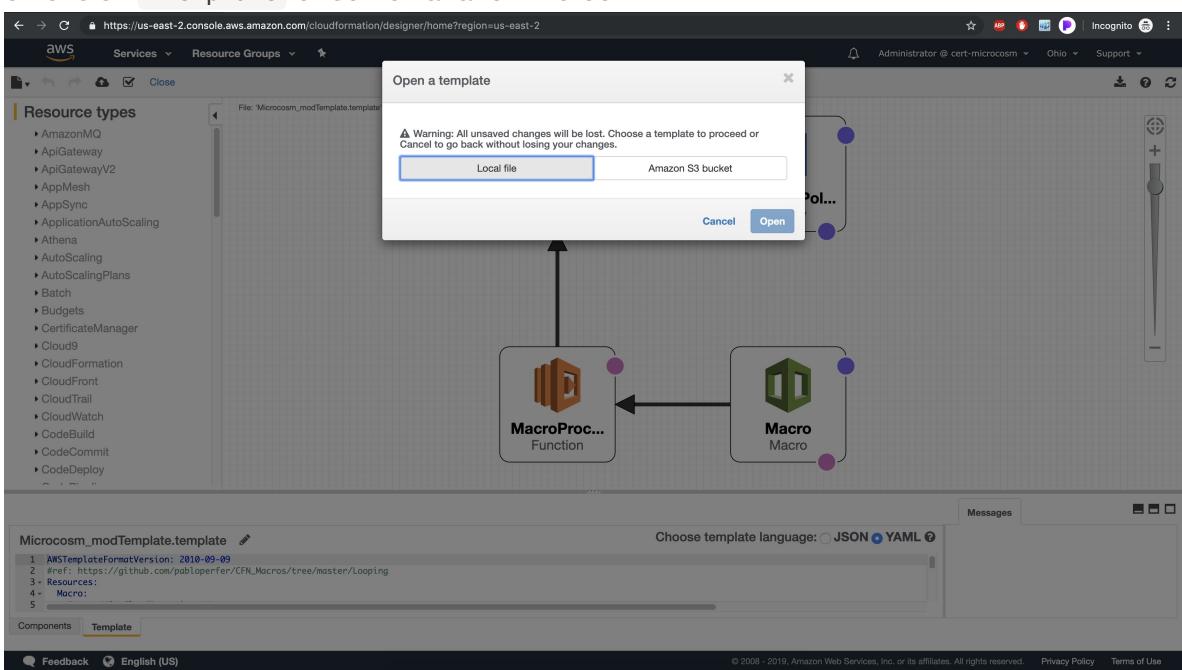
- To create or view a template, on the Select Template page, click Design Template



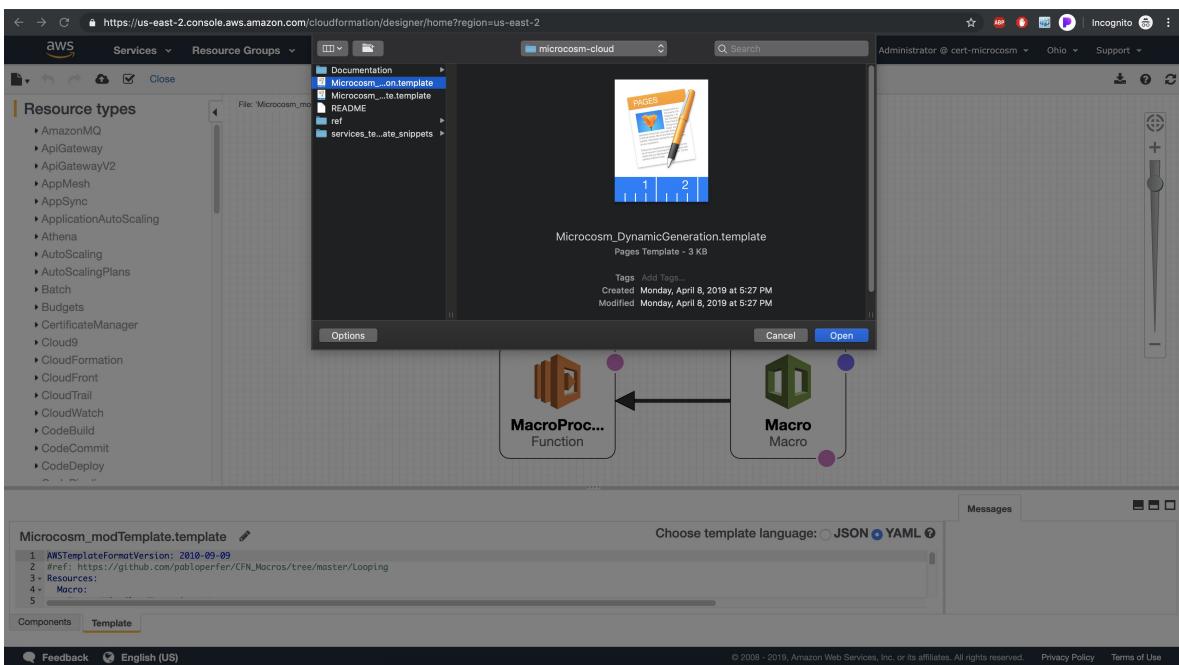
- To load an existing template, select open from the file icon in the left menu. (Note, the template pictured graphically here represents the Macro stack we'll be deploying later)



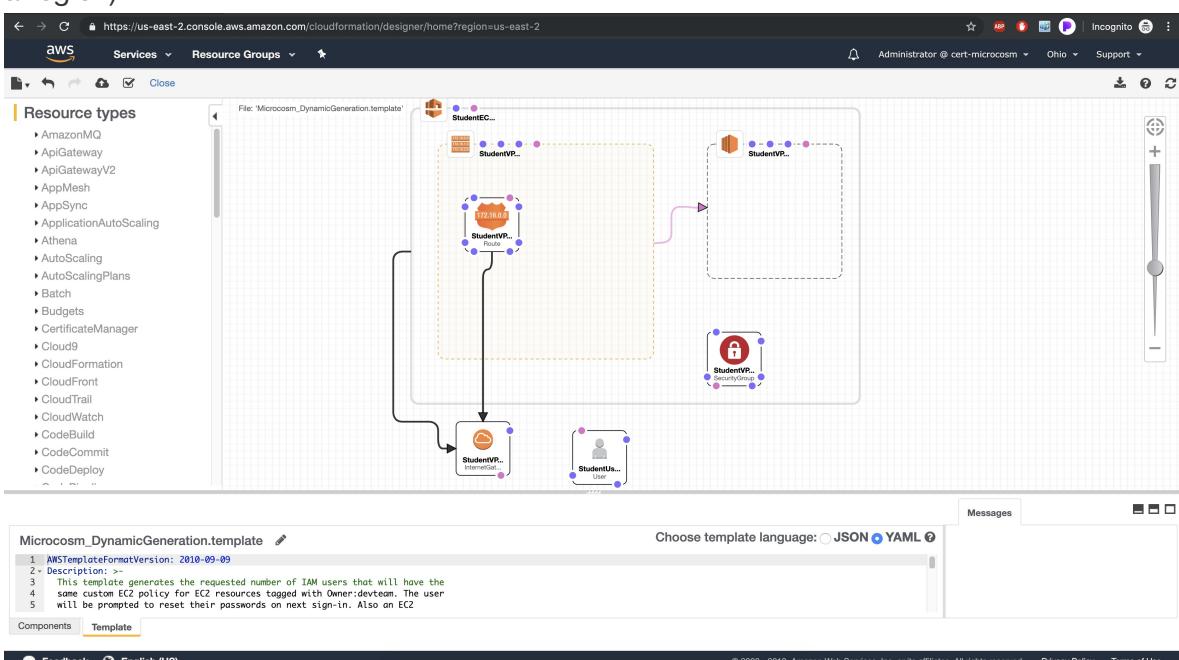
- Select the source of your template. It is good practice to name your templates with the extension `.template` or some variation thereof



- Navigate to and select your template and press Open



- With the template loaded, you are free to inspect, augment, change, or update its various elements. The template depicted here instantiates a VPC with the accompanying internet gateway, routes, security group, a user account, etc. This is our Dynamic template - as in when the template is loaded, the Macro template uses a lambda function, written in python, to duplicate the elements within the template as many times as specified. Note, that AWS templates have a hard limit of 200 elements per template and AWS have a default limit of 5 VPCs and internet gateways (see above for requesting additional VPCs in a region).



- 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'. On the left, a sidebar lists 'Select Template' (which is highlighted in orange), 'Specify Details', 'Options', and 'Review'. The main area is titled 'Select Template' with the sub-instruction: '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' (with a link to 'Use AWS CloudFormation Designer') and 'Choose a template' (with a link to 'A template is a JSON/YAML-formatted text file that describes your stack's resources and their properties'). Under 'Choose a template', three options are listed: 'Select a sample template' (radio button), 'Upload a template to Amazon S3' (radio button selected), and 'Specify an Amazon S3 template URL'. A file input field shows 'Microcosm_mo...te.template'. At the bottom right are 'Cancel' and 'Next' buttons.

- IMPORTANT name the stack being created from this template `Macro`. The dynamic template being loaded next requires this naming for proper deployment and variable reference.
- Press Next

The screenshot shows the 'Create stack' wizard in the AWS CloudFormation console. The current step is 'Specify Details'. On the left, a sidebar lists 'Select Template' (highlighted in orange), 'Specify Details' (selected), 'Options', and 'Review'. The main area is titled 'Specify Details' with the sub-instruction: 'Specify a stack name and parameter values. You can use or change the default parameter values, which are defined in the AWS CloudFormation template.' Below this is a 'Stack name' input field containing 'Macro'. At the bottom right are 'Cancel', 'Previous', and 'Next' buttons.

- Press Next

Permissions

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

IAM Role Choose a role (optional)
Enter role arn

Rollback Triggers

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

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

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

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

- Select the acknowledgement and press Create

Details

Stack name: Macro

Options

Tags
No tags provided

Rollback Triggers
No monitoring time provided
No rollback triggers provided

Advanced

Notification	
Termination Protection	Disabled
Timeout	none
Rollback on failure	Yes

Capabilities

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

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

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

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

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

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

Events

Date	Status	Type	Logical ID	Status Reason
2019-04-11	CREATE_COMPLETE	AWS::CloudFormation::Stack	Macro	
	CREATE_COMPLETE	AWS::IAM::Policy	LambdaPolicy	
	CREATE_COMPLETE	AWS::CloudFormation::Macro	Macro	
	CREATE_IN_PROGRESS	AWS::CloudFormation::Macro	Macro	Resource creation Initiated
	CREATE_IN_PROGRESS	AWS::CloudFormation::Macro	Macro	
	CREATE_COMPLETE	AWS::Lambda::Function	MacroProcessor	
	CREATE_IN_PROGRESS	AWS::Lambda::Function	MacroProcessor	Resource creation Initiated
	CREATE_IN_PROGRESS	AWS::IAM::Policy	LambdaPolicy	Resource creation Initiated
	CREATE_IN_PROGRESS	AWS::Lambda::Function	MacroProcessor	
	CREATE_IN_PROGRESS	AWS::IAM::Policy	LambdaPolicy	
	CREATE_COMPLETE	AWS::IAM::Role	LambdaExecutionRole	

- Repeat the Stack creation process for the Microcosm Dynamic Template

Create stack

Select Template

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

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

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

Select a sample template

Upload a template to Amazon S3

Specify an Amazon S3 template URL

[Choose File](#) Microcosm_Dy...on.template

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

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

<https://us-east-2.console.aws.amazon.com/cloudformation/home?region=us-east-2#/stacks/new>

AWS Services Resource Groups Create Stack

CloudFormation Stacks Create Stack

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: VPCCreation

Parameters

InstanceType: t2.micro WebServer EC2 instance type

NumberOfUsers: 0 Enter the number of users to create for the Dev Team

SSHLocation: 0.0.0.0/0 Lockdown SSH access to the bastion host (default can be accessed from anywhere)

Cancel Previous Next

- Press Next

<https://us-east-2.console.aws.amazon.com/cloudformation/home?region=us-east-2#/stacks/new>

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: Choose a role (optional) Enter role arn

▼ Rollback Triggers

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

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

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

► Advanced

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

Cancel Previous Next

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

Advanced

Notification	Disabled
Termination Protection	Disabled
Timeout	none
Rollback on failure	Yes

Capabilities

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

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

Transforms

Check the following transforms: ["443007076818::Macro"]
You must use a change set to create this stack because it includes one or more transforms. The change set shows the resources that transforms add to your stack's template. Choose Create Change Set, check the resources that the transforms add, and then choose Execute. [Learn more.](#)

[Create Change Set](#)

Cancel Previous Execute

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

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

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.](#)

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

Action	Logical ID	Physical ID	Resource type	Replacement
Add	StudentEC2VPC		AWS::EC2::VPC	
Add	StudentUser		AWS::IAM::User	
Add	StudentVPCGatewayAttachment		AWS::EC2::VPCGatewayAttachment	
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

Cancel Previous Execute

Feedback English (US) © 2008 - 2019, Amazon Web Services, Inc. or its affiliates. All rights reserved. Privacy Policy 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 for 'MacroProcessor'. The table columns are 'Function name', 'Description', 'Runtime', 'Code size', and 'Last modified'. The 'MacroProcessor' row has 'IAM Users Macro processor function' in the Description column, 'Python 3.6' in the Runtime column, '889 bytes' in the Code size column, and '4 minutes ago' in the Last modified column. At the top right of the main area, there are 'Actions' and 'Create function' buttons.

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

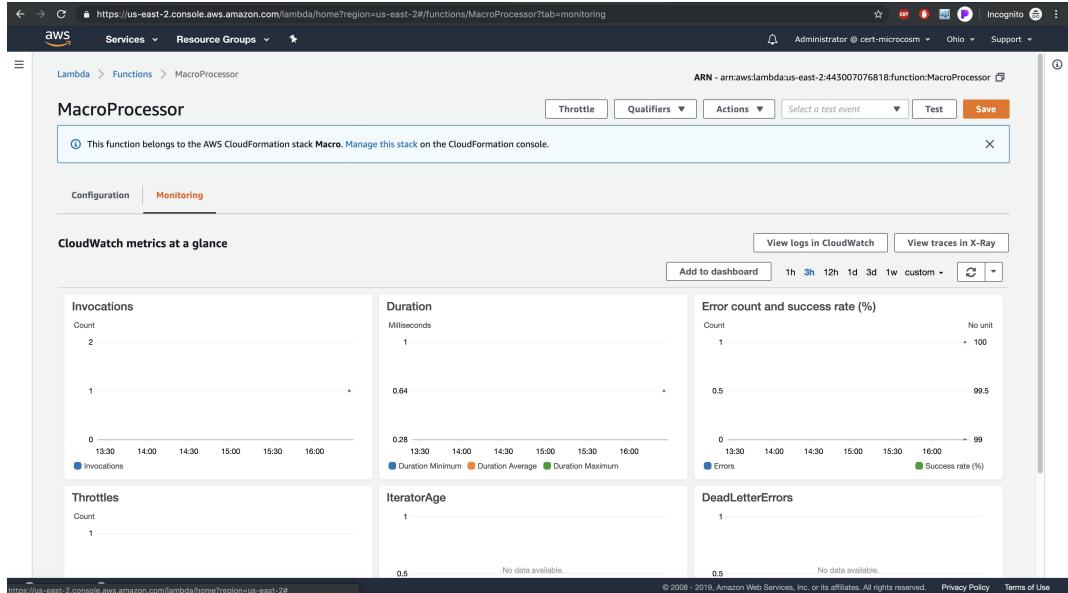
The screenshot shows the AWS Lambda Function code editor for the 'MacroProcessor' function. The top navigation bar includes tabs for 'Throttle', 'Qualifiers', 'Actions', 'Select a test event', 'Test', and 'Save'. The 'Handler' dropdown is set to 'index.lambda_handler'. The code entry type is 'Edit code inline' and the runtime is 'Python 3.6'. The code editor window displays the 'index.py' file content:

```

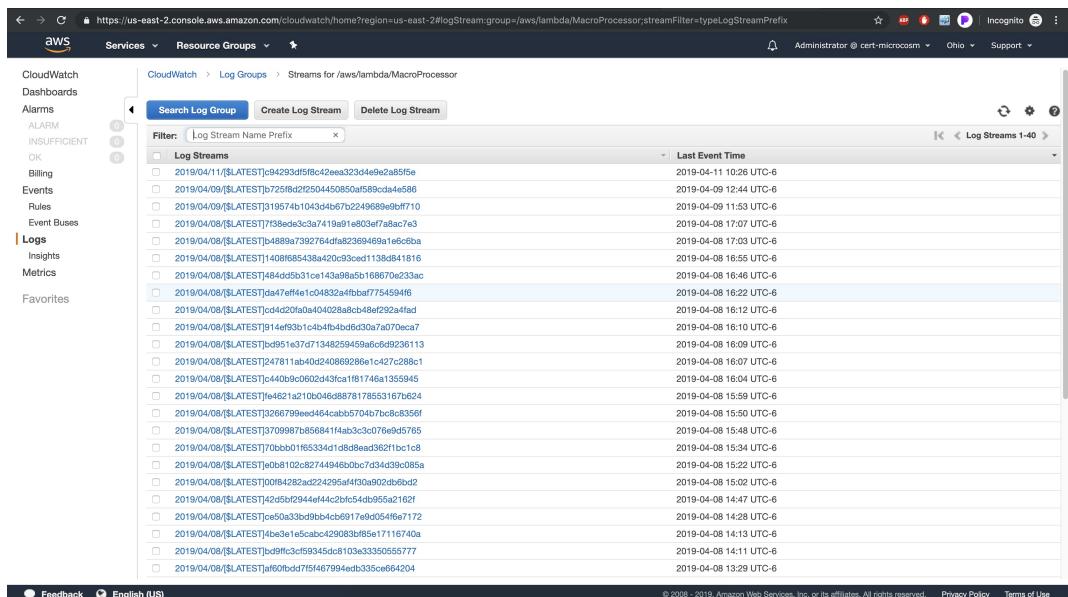
1 import json
2 import copy
3
4 def lambda_handler(event, context):
5     Number = event["templateParameterValues"]["NumberOfUsers"]
6     #I declare a new dictionary to have the resources object
7     TemplateDetailsVar = {}
8     TemplateDetailsVar["FinalFragment"] = FinalFragment["Resources"]
9     for i in range(int(Number)):
10         num=str(i)
11         TemplateDetailsVar["StudentUser"+num] = copy.deepcopy(TemplateDetailsVar["StudentUser"])
12         TemplateDetailsVar["StudentVPC"+num] = copy.deepcopy(TemplateDetailsVar["StudentVPC"])
13         for i in range(int(Number)):
14             num=str(i)
15             #I modify the IAM User resource logical Id in order to add it later to "Resources" object the number of times requested
16             num=str(i)
17             TemplateDetailsVar["StudentVPCSUBNET"+num] = copy.deepcopy(TemplateDetailsVar["StudentVPCSUBNET"])
18             TemplateDetailsVar["StudentVPC"+num]["Properties"]["VpcId"]["Ref"] = "Vpc"+num
19             TemplateDetailsVar["StudentVPC"+num]["Properties"]["VpcId"]["Ref"] = "Vpc"+num
20             TemplateDetailsVar["StudentVPCSUBNET"+num] = copy.deepcopy(TemplateDetailsVar["StudentVPCSUBNET"])
21             TemplateDetailsVar["StudentVPCSecurityGroup"+num] = copy.deepcopy(TemplateDetailsVar["StudentVPCSUBNET"+num])
22             TemplateDetailsVar["StudentVPCSecurityGroup"+num]["Properties"]["VpcId"]["Ref"] = "Vpc"+num
23             TemplateDetailsVar["StudentVPCSecurityGroup"+num]["Properties"]["VpcId"]["Ref"] = "Vpc"+num
24             TemplateDetailsVar["StudentVPCSecurityGroup"+num] = copy.deepcopy(TemplateDetailsVar["StudentVPCSecurityGroup"])
25             TemplateDetailsVar["StudentVPCSecurityGroup"+num]["Properties"]["VpcId"]["Ref"] = "Vpc"+num
26             TemplateDetailsVar["StudentVPCRoutingTable"+num] = copy.deepcopy(TemplateDetailsVar["StudentVPCRoutingTable"])
27             TemplateDetailsVar["StudentVPCRoutingTable"+num]["Properties"]["VpcId"]["Ref"] = "Vpc"+num
28             for dep in TemplateDetailsVar["StudentVPCRoutingTable"+num]["DependsOn"]:
29                 dependencies.append("Vpc"+dep)
30             TemplateDetailsVar["StudentVPCRoutingTable"+num]["Properties"]["VpcId"]["Ref"] = "Vpc"+num
31             TemplateDetailsVar["StudentVPCRoutingTable"+num]["DependsOn"] = dependencies
32
    
```

The status bar at the bottom right shows '14:33 Python Spaces: 2'.

- 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



- View the logs generated by the lambda function

- 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

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-WMCVVKHLQ8WCF	None	None	Today	None	Not enabled

- 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, there's a sidebar with various VPC-related options like Subnets, Route Tables, and Security Groups. The main area displays a table with two rows of VPC information:

Name	VPC ID	State	IPv4 CIDR	IPv6 CIDR	DHCP options set	Main Route table	Main Network ACL	Ter
vpc-01ca865d17b887b66	vpc-01ca865d17b887b66	available	10.0.0.0/24	-	dopt-5f946034	rtb-0a1862380212061b	acl-0a3d1f85b31db190	def
vpc-9dd0c3f5	vpc-9dd0c3f5	available	172.31.0....	-	dopt-5f946034	rtb-d764bcbc	acl-597f8132	def

[Return to Table of Contents](#)

APPENDIX B

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

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

```
microcosm-cloud — vi ~./.bash_profile — 122×60
export PATH=$PATH:/Users/smorley/Library/Python/3.7/bin
```

- Install the AWS CLI

```

microcosm-cloud — bash — 122x60
mac-topazpommme:microcosm-cloud smorley$ echo $SHELL
/bin/bash
mac-topazpommme:microcosm-cloud smorley$ vi ~/.bash_profile
mac-topazpommme:microcosm-cloud smorley$ source ~/.bash_profile
mac-topazpommme: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-topazpommme: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-topazpommme:microcosm-cloud smorley$ aws --version
aws-cli/1.16.140 Python/3.7.2 Darwin/18.5.0 botocore/1.12.130
mac-topazpommme:microcosm-cloud smorley$
```

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

```

[mac-topazpommme:devops-microcosm smorley$ aws ecr get-login --region us-east-2
Unable to locate credentials. You can configure credentials by running "aws configure".
[mac-topazpommme:devops-microcosm smorley$ aws configure
AWS Access Key ID [None]: AKIAWOJJKHRHJCDDYAKNE
AWS Secret Access Key [None]: ZNHgDgxWBqnBR+K2AxGncsul7m2SIH0PJk4yqit5
Default region name [None]: us-east-2
Default output format [None]: json
```

- Get Temporary AWS ECR Login credentials to use with Docker

```

[mac-topazpommme:devops-microcosm smorley$ aws ecr get-login --region us-east-2 --no-include-email
docker login -- AWS -eyJwXlsb2fkijoiKzhsTwCL2krRjFol3p1VzfcjBlaJNvbWhPUgMw50K01xNVJLUUpXR1pHTGFDTQTMzQ2VBY3hYZZ9XT0
SS1YzYkp1dGdvZ0kvaWxzZAY21EanJTOxCaVzJemdXWNNScEx4Ymg3NDvbw11QzTfHQS8wwWdmRnZ1N1Bb1fQa0pwS1Nza0hPzkr5N0vUswtVVVBjWxOrZ
1EzWDNBcldcydnF5bjY0WDHaU014b3F2bkdoV09hKzM4bz1PNwdBvWtMwmVsMxhewpBQncqyNFbxVWdlcldKeG5SbGt2enVrSxhmQjFhZ20zeK5sMkZhRTFydh
ibEW3n12ZGvNzNGeTu4c3Fraig1SG1MdFl6Y0oztLrmMkjkck1c1j1UkR67G1xZ2ZMRf14eG1PMVVIdjaWaUov2hEa1vtoFJMcdMSGhabzJ1MnpjtkRvc
3gybVNROw5EM2dyvZ1Ntl1ywUfhZcm1dkkkvNFJERERxYnA5exZmcxHVsazV2OUF6MGzUclld3nJUSS9IZhJCzzFxevVnc3hnSm0RXY5Tuata0WZzb0pvRF1xSlh
6N2ZndWpIRXM0Y3AyM2dUSxEhd0JzrN09FVkl1NvgvV3NIMEFMaHbhYwI2UxmzbujVa1Rvd1pMd1M0Uk5zL01WmThxb0hRrk4vUU1iSw9qaWxNRGxnMFHQkqkV
GxaYTNy0FvrcKZMNE5BU01mdFNS0l0Nnc3a1VkeMz1ZwvDZZWZiekwot3VtndRPWTY3RDRdz3JmTRuRWZ3bKj1duo5sUnpveFyK0pJtms0aEVDuNf
JQ2x3SFU1S0p5REZOYXlyRFEzeQ05cU5GUeJxR314buRxerW00MkorUy8vsE1DTTc3QWerl0UzNFBnaft1N2E5a3kEyZPVYXQzUGNxueFEUKJaaFYzyzQyd
TRKQ1VQUWtDZUuYudV4WEFLuNjNOUz4RFJwT315by9iUxVHNgiWeQxZ0htUVhnRx30Uh1wnN5aEVXdpn3bnVhaC9tajVUUU1svjNRL1YzbW9LMGz0Vw9Hwm9
FwmIxRhrUjBHSU9SUZtZoZGMwGxrZS9zNu9URfl1szJvsKck3bE1yVzdCM09jXmddoNG1oUhPvNxxNpYUhjK01nfOpb252Z1vrtMTN40GxMRVrwefHln011q1g5N
zdeErVtjdnbksvZS9pd1JhvVvhHS010Usxbw1Lb1VSL1R6V1ZzeK1MVHJ30Tk5MXBGaU5CK2pmeVVQu0o4ZVmNkQwhaHc2cmw5bkhkbw1BYUzhNHFuUW5bUQ
xbmcxUWVwa01iuUh6aUrZzm5tNFZGOWNmavZSY21DWGJXM0FMWwpaxZBkJFJWM0FkCw9VUe1kd3NtD2NyChnsZHBCU0x5Yk1UvdrmzxanRvt2yzWgpObzVke
GBzU3RzbmxOsKvpMu9yUgkxTvh2BFUlcisMkd4an1CMWY3K29DmzR13orVkuVudFnMjtR22ZFRpTVkxMckJ3PT0iLCjkYXRa2V5IoiQVFFQkFiakI3L21
nd01nNE50d2F1cnhTSV14NeHmbnh1R2MvNdh1RHd2d0RwT11XWmdBQUFINHdmQv1KS29asWh2Y05BWnhb0c4d2JRSUJBREJvQmdrcWhraUc5dzBCQndFd0hnW
UpZSVpJQVdVREJBRXVNQKVFXRERYKViQzNwY0htR3c2bk5BSUJSUE3eUuveW1I1Mkp00X15MnMzeTRuD3JiN9nRGv0d1ArNCswOViwcim1kM0dPL11va196TU9
BNEpiNvprUHM4QWFqwnRNm09vL11xY1FLNmJsdzob1LCJ2ZXJzaW9uIjoimIsInR5cGUio1jeQvrbx0tFWSIsImV4cGlyXRpb24i0jE1NTQ5NjQ40DF9 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 eyJwYXlsb2FkIjoiKzsTwTcL2krRjFoL3plVzFVcjBlajNvbWhPUgPmW
50K01xNVJLUUpXR1pHTGFDQTMzQ2VBY3hY29XT0RSS1YzYkpIdGdvZ0kvawXhZZ4Y21EanJTQXFcaVzJemdXvNsCEx4Ymg3NDbvBw1JQTHQs8wWdmRnZ
N1dBb1FQo0pwS1Nza0hPzkr5N0VuSwtVYyBjWxOrZ1EzWDNb1cydnF5bjY0WDhaU014b3f2bkdoV09hKzM4bz1PNwdbVwttMwmVsMXhweWpBQncyNFBXVd1cl
dkG5SbGt2enPvrsXhmQjFHZ20zek55mkzHRTFydhnbew3n12ZQqvNzGeTu4c3FrAig1SG1MdF1GYy0z1RmMkJckd1cjlUUkr6TGlxZ2ZMRf1eG1PMVVI
djAwaUovb2hEa1t0fMcDm3GhabzJ1MnpjTkRvc3gvbVNRwp5EM2d2vZ1ntlUyUfhZcm3pdkvNFJERERxYnA5ezXmcHVsazV2OUF6MGZucl3NnJUSS9IZH
JCzzFxevVnC3hnSm00RXY5Tuta0WZzb0pvRF1Xs1h6N2ZndWpIRXM0Y3AyM2dUSEhodUjzN09FVkl1NVvgvV3NIMEFMaHBhYWI2UMzbUJVa1Rvd1pMd1M0Uk5z
L01WMThXb0hRRk4uuU1isW9qWxNRGxnMFICGkozVGxaTYNYOFVrckzZNMe5BU01mdFNSL0Ncc2hTSUd3aV1kemZIZWcyyDZVVZiekwot3VtNDRPWTY3RDRzb3
JaMTRuRWZ3bkJ1duo55UNPVEFyK0pJ7ms0aEVDUfFJQ2x3SFU1S0p5REZOYXlyRFEEzeQ05Cu5GUeJxR314bURxeFYwd0@0MkorUy8vSE1DTTc3QWEzL0UzNFBN
aEt1N2E5a3k5YzPVYXQzUGNxelUEFuKjaAFyZyQdydTRkQ1VQUytDZUuyUDV4WEFLuNjNOUz4RFJwT315by9IUVHNG9iWEQz0h0tUVhnRx30U1WnN5aEVxdn
p3bnVhaC9ta1vU015VjNRL1yb2w9LMGZ0W9Hwm9FwMrHrjUjBHSU9u9rZGmzWgxZ921uJsvkk3b1yvzdCM09jMxd0Ng1oUhPvnXNyUjh
K01oNfp0b25221vrMTN40GxMRVWeHFLN0T101g5NzddeErVtjdnbksvZS9Dd1JHVVHHS0NI0uSw1b1vSL1R5V1Zze1MVHU3OTk5MXBGAU5CK2pmeVQu0
o4ZXVmNkQwaHc2cmw5bkhkblBYUZhNHFuUW55bUQxbmcxUWVwa01iuUh6aURZm5tNFZGOWNmazVSZy21DWGJXM0FMWwpaxZBjUFJW0FkcW9VUe1kd3NtD2Ny
oZHZBCUxGyK11uvdRzxanTzYzGpObzVkeG8zU3RzbmxSkpUyUgkxTvh2bfUfciszMkd4an1CMWY3K29DmzRil3orVkuwvUdfNmJTR2ZFRopTvK
xMckJ3PT0iLCJyXRh2V5IjoiQVFQkF1ak13L21nd01nHE5Qd2f1cnhTSV14Nehbmn1R2MvNDhRhd2dRwt11xWmdBQUFINHdmQV1KS29aSwH2Y05BuWNH
b0c4d2JRSUJBRE3vQmdrcWhraUc5dzBCQndFd0hnWUpzSVpJQvDvREJBRXVNQkVFRexRYkViQzNwY0hR3c2bk5BSUJFSUe3eUuveWI1Mkp0Ox15MnMzeTRUd3
J1m9nRGV0d1ArNCsw0V1wm1kM0dPL11va196TU9BNep1NvpurUHM40WFqwnRNM09v1L1ix1FLNmJsdz0iLCJ2ZJzaW9uIjoiMiIsInR5cGuio1jeQvrbx0tf
WS1sImV4cGlyYXRp24i0je1NTQ5NjQ40DF9 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
b57f36821ce5        sonarqube:lts      "/bin/run.sh"           7 days ago          Exited (0) 42 minutes ago
73a4061cd9f0        gillax/hubot-slack-jenkins   "/bin/hubot -a slack"    7 days ago          Exited (0) 37 minutes ago
ba06c7de2e42        gitlab/gitlab-ce       "/assets/wrapper"        7 days ago          Exited (0) 37 minutes ago
3193cf8470b         sonatype/nexus        "/bin/sh -c 'java ...'"  7 days ago          Exited (143) 37 minutes ago
013bb877e566        h1kkan/jenkins-docker:lts   "/sbin/tini -- /usr/..."  7 days ago          Exited (143) 37 minutes ago
3fd4643668ec        oawasp/zap2docker-stable   "zap-webswing.sh"       7 days ago          Exited (0) 44 minutes ago
[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

```

mac-topazpomme:devops-microcosm smorley$ docker 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  4.77GB
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:l
ts

```

- Verify tag is correct by relisting images

```

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

```

- Create the desired ECR repository (1 repository per image type, in this example only

sonarqube images) and push the image

```
mac-topazpomme:devops-microcosm smorley$ aws ecr create-repository --repository-name sonar
qube
{
    "repository": {
        "repositoryArn": "arn:aws:ecr:us-east-2:443007076818:repository/sonarqube",
        "registryId": "443007076818",
        "repositoryName": "sonarqube",
        "repositoryUri": "443007076818.dkr.ecr.us-east-2.amazonaws.com/sonarqube",
        "createdAt": 1554929452.0
    }
}
[mac-topazpomme:devops-microcosm smorley$ docker tag sonarqube:lts 443007076818.dkr.ecr.us-]
east-2.amazonaws.com/sonarqube:lts
[mac-topazpomme:devops-microcosm smorley$ docker push 443007076818.dkr.ecr.us-east-2.amazon]
aws.com/sonarqube:lts
The push refers to repository [443007076818.dkr.ecr.us-east-2.amazonaws.com/sonarqube]
68e5087b6ffc: Pushed
237afe62393f: Pushed
079758d1bbb7: Pushed
e11f2ab9e2f4: Pushed
f7d12d471667: Pushed
f350d0146bb3: Pushed
e38df31d449c: Pushed
af5ae4841776: Pushed
b17cc31e431b: Pushed
12cb127eee44: Pushed
604829a174eb: Pushed
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 h1kkan/jenkins-docker
{
  "repository": {
    "repositoryArn": "arn:aws:ecr:us-east-2:443007076818:repository/h1kkan/jenkins-docker",
    "registryId": "443007076818",
    "repositoryName": "h1kkan/jenkins-docker",
    "repositoryUri": "443007076818.dkr.ecr.us-east-2.amazonaws.com/h1kkan/jenkins-docker",
    "createdAt": 1554931781.0
  }
}
[mac-topazpomme:devops-microcosm smorley$ docker tag h1kkan/jenkins-docker:lts 443007076818.dkr.ecr.us-east-2.amazonaws.com/h1kkan/jenkins-docker:lts
[mac-topazpomme:devops-microcosm smorley$ docker push 443007076818.dkr.ecr.us-east-2.amazonaws.com/h1kkan/jenkins-docker:lts
The push refers to repository [443007076818.dkr.ecr.us-east-2.amazonaws.com/h1kkan/jenkins-docker]
4717473633af: Pushed
01e56e46ef42: Pushed
0e21db749a1b: Pushed
1abb109c6333: Pushed
ea9254dfb470: Pushed
781f3534b8f0: Pushed
69ed8c046be0: Pushed
1cf9d4808e4e: Pushed
2b138287e104: Pushed
e674c0445366: Pushed
86cea145f8b3: Pushed
653304fe4cf3: Pushed
04b4fa9a597a: Pushed
0d790265fcc4: Pushed
d97f26ef1623: Pushed
e87475e5811f: Pushed
51e6d0a07219: Pushed
ba57bc494b22: Pushed
3443d6cf0f1f: Pushed
f3a38968d075: Pushed
a327787b3c73: Pushed
5bb0785f2eee: Pushed
lts: digest: sha256:a2a91ba9249a31a33ab9877bab42c7dea49955b30740c7c7beb6ffa124bc77ca size: 4925
```

- View repositories and their details in the AWS ECR Console

The screenshot shows the AWS ECR console homepage. The main heading is "Amazon Elastic Container Registry" with the subtext "Easily store, manage, and deploy container images". Below this is a brief description: "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 large diagram titled "How it works" illustrates the process: "Write code" leads to "Amazon ECR" (represented by a cube icon), which then connects to "Deploy, store, and control access to images" (represented by a camera icon). This leads to "Version history and image lifecycle policies" (represented by a gear icon), which then connects to "Run containers" (represented by a server icon). Finally, "Run containers" connects to "Amazon ECS, Amazon EKS, AWS Lambda, or on-premises" (represented by a cloud icon). To the right of the diagram, there are sections for "Pricing (US)" (pay only for storage and data transferred to the internet), "Getting started", and links to "What is Amazon ECR?", "Getting started with ECR", and "Set up a CI/CD pipeline with ECR". On the left sidebar, under "Amazon Container Services", there are links for "Amazon ECS Clusters", "Task definitions", "Amazon EKS Clusters", and "Amazon ECR Repositories". The top navigation bar shows the URL "https://us-east-2.console.aws.amazon.com/ecr/get-started?region=us-east-2", the user "Administrator @ cert-microcosm", and the region "Ohio".

The screenshot shows two screenshots of the AWS ECR console side-by-side.

Left Screenshot (List of Repositories):

- URL: <https://us-east-2.console.aws.amazon.com/ecr/repositories?region=us-east-2>
- Header: AWS Services Resource Groups
- Left sidebar: Amazon Container Services (Amazon ECS, Clusters, Task definitions), Amazon EKS (Clusters), Amazon ECR (Repositories)
- Main content: **Repositories (6)**
 - gillax/hubot-slack-jenkins
 - gitlab/gitlab-ce
 - h1kkan/jenkins-docker
 - owasp/zap2docker-stable
 - sonarqube
 - sonatype/nexus

Right Screenshot (Detailed View of Repository Images):

- URL: <https://us-east-2.console.aws.amazon.com/ecr/repositories/gillax/hubot-slack-jenkins?region=us-east-2>
- Header: AWS Services Resource Groups
- Left sidebar: Amazon Container Services (Amazon ECS, Clusters, Task definitions), Amazon EKS (Clusters), Amazon ECR (Repositories, Images, Permissions, Lifecycle Policy, Tags)
- Main content: **gillax/hubot-slack-jenkins**
 - Images (1)**

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

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

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

```
graph LR; A[Write code in a Docker image] --> B[Amazon ECR]; B --> C[Compress, encrypt, and control access to images]; C --> D[Version, tag, and manage image lifecycles]; D --> E[Push containers]; E --> F[Pull images and run containers anywhere]; F --> G[Amazon ECS, Amazon EKS, AWS Cloud, On-premises]
```

Benefits

Fully managed	Secure
Amazon ECR	Amazon ECR

Launchpad

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

[Get Started](#)

ECR > Repositories > Create repository

Create repository

Repository configuration

Repository name
443007076818.dkr.ecr.us-east-2.amazonaws.com/ []
A namespace can be included with your repository name (e.g. namespace/repo-name).

Cancel [Create repository](#)

ECR > Repositories > Create repository

Create repository

Repository configuration

Repository name
443007076818.dkr.ecr.us-east-2.amazonaws.com/ [] microcosm-cloud
A namespace can be included with your repository name (e.g. namespace/repo-name).

Cancel [Create repository](#)

Screenshot of the AWS ECR console showing a successfully created repository named "microcosm-cloud".

The screenshot shows the AWS ECR service page with a green header bar indicating "Successfully created repository". The main area displays a table of repositories, with one entry for "microcosm-cloud". The table includes columns for Repository name, URI, and Created at.

Repository name	URI	Created at
microcosm-cloud	443007076818.dkr.ecr.us-east-2.amazonaws.com/microcosm-cloud	04/10/19, 11:49:49 AM

[Return to Table of Contents](#)

ECS Get Started Wizard

Using Get Started Wizard

Follow the screen shots for reference

Screenshot of the AWS ECS Get Started Wizard landing page.

The page features a video player showing a cartoon character interacting with a large screen displaying a pie chart and a grid of icons. Below the video, a paragraph of text describes the benefits of using Amazon ECS, followed by a "Get started" button and a link to "Learn more about Amazon ECS".

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](#)

The page also includes three callout sections with icons and text:

- Run containers at scale** (Icon: Three stacked containers with a plus sign)
- Flexible container placement** (Icon: A stack of containers with a gear icon)
- Integrated and extensible** (Icon: A stack of containers with a circular arrow icon)

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

AWS Services Resource Groups

Clusters

An Amazon ECS cluster is a regional grouping of one or more container instances on which you can run task requests. Each account receives a default cluster the first time you use the Amazon ECS service. Clusters may contain more than one Amazon EC2 instance type.

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 ARNs of tasks, container instances, and services are longer because they now contain the cluster name.

[Configure ECS ARN setting](#)

[Create Cluster](#) [Get Started](#)

View list card

No clusters found

[Get Started](#)

0 - 0 of 0

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

AWS Services Resource Groups

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	nginx
image : httpd:2.4	image : nginx:latest
memory : 0.5GB (512)	memory : 0.5GB (512)
cpu : 0.25 vCPU (256)	cpu : 0.25 vCPU (256)

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

Task definition

[Edit](#)

Screenshot of the AWS ECR (Amazon Container Registry) console showing the repository 'h1kkan/jenkins-docker'. The repository contains one image named 'lts'.

Image tag	Image URI	Pushed at	Digest	Size (MB)
lts	443007076818.dkr.ecr.us-east-2.amazonaws.com/h1kkan/jenkins-docker:lts	04/10/19, 3:53:58 PM	sha256:a2a91be92...	644.06

Screenshot of the AWS ECS (Amazon Elastic Container Service) console showing the 'Edit container' configuration page. The container is named 'jenkins' and uses the image '443007076818.dkr.ecr.us-east-2.amazonaws.com/h1kkan/jenkins-docker:lts'.

Container definition

Choose an image for your container based on the task definition you selected.

Task definition

- sample-app
 - image : httpd:2.4
 - memory : 0.6GB (512)
 - cpu : 0.25 vCPU (256)
- tomcat-webserver
 - image : tomcat
 - memory : 2GB (2048)
 - cpu : 1 vCPU (1024)

Port mappings

Container port	Protocol
8080	tcp

Host port mappings are not valid when the network mode for a task definition is host or awsvpc. To specify different host and container port mappings, choose the Bridge network mode.

* Required

Cancel Update

Getting Started with Amazon Elastic Container Service

Step 1: Container and Task

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)

* Required

Task definition

STORAGE AND LOGGING

Read only root file system

Mount points

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

Add mount point

Volumes from

Log configuration Auto-configure CloudWatch Logs

Log driver: awslogs

Log options

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

Add key **Add value**

Cancel **Update**

sample-app

nginx

tomcat-webserver

jenkins

Configure

Task definition

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

Task definition name: first-run-task-definition

Network mode: awsvpc

Task execution role: Create new

Compatibilities: FARGATE

Task memory: 0.5GB (512)

Task CPU: 0.25 vCPU (256)

*Required

Cancel **Next**

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

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

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

Diagram of ECS objects and how they relate

Define your service

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

Service name: Jenkins-service

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)

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

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

Diagram of ECS objects and how they relate

Configure your cluster

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

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

Cluster name: microcosm

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

VPC ID: Automatically create new

Subnets: Automatically create new

*Required Cancel Previous Next

Feedback English (US)

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

Review
Review the configuration you've set up before creating your task definition, service, and cluster.

Task definition

Task definition name **first-run-task-definition** [Edit](#)

Network mode **awsvpc**

Task execution role **Create new**

Container name **jenkins**

Image **443007076818.dkr.ecr.us-east-2.amazonaws.com/h1khan/jenkins-docker:its**

Memory **512**

Port **8080**

Protocol **HTTP**

Service

Service name **jenkins-service** [Edit](#)

Number of desired tasks **1**

Cluster

Cluster name **microcosm** [Edit](#)

VPC ID **Automatically create new**

Subnets **Automatically create new**

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

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

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

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

[Back](#) [View service](#) Enabled after service creation completes successfully

Additional features that you can add to your service after creation

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

Preparing service : 2 of 9 complete

ECS resource creation	Status
Cluster microcosm	complete ✓
Task definition first-run-task-definition:3	complete ✓
Service	pending ⏱
Additional AWS service integrations	
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

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

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

Feedback English (US)

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

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

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

Service : jenkins-service

Cluster	microcosm	Desired count	1
Status	ACTIVE	Pending count	1
Task definition	first-run-task-definition:3	Running count	0
Service type	REPLICA		
Launch type	FARGATE		
Platform version	LATEST(1.3.0)		
Service role	AWSServiceRoleForECS		

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

Load Balancing

Load Balancer Name	Container Name	Container Port
No load balancers		

Network Access

Allowed VPC	vpc-0530acf15c751393
Allowed subnets	subnet-0521ea54a64435dd4 , subnet-0e50c4d05af5052a7
Security groups*	sg-06f07d8d65ef6b2fd
Auto-assign public IP	ENABLED

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#clusters/microcosm/services/jenkins-service/deployments

Administrator @ cert-microcosm Ohio Support

Clusters > microcosm > Service: jenkins-service

Service : jenkins-service

Cluster: microcosm Status: ACTIVE Desired count: 1 Pending count: 0 Running count: 1

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

Details Tasks Events Auto Scaling Deployments Metrics Tags Logs

Task Placement

Strategy: No strategies Constraint: No constraints

Service Deployment Options

Minimum healthy percent: 100 Maximum percent: 200

create pipeline | view pipelines

Last updated on April 11, 2019 8:48:34 AM (0m ago)

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

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

Administrator @ cert-microcosm Ohio Support

Clusters > microcosm > Service: jenkins-service

Service : jenkins-service

Cluster: microcosm Status: ACTIVE Desired count: 1 Pending count: 0 Running count: 1

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

Details Tasks Events Auto Scaling Deployments Metrics Tags Logs

Task status: Running | Stopped

Last updated on April 11, 2019 8:48:45 AM (0m ago)

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

Amazon ECS

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

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

Run more like this Stop

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-ad8f-fcab3ac20078	RUNNING	443007076818.dkr...	0	--/-	true

Amazon ECS

Clusters

Task Definitions

Amazon EKS

Clusters

Amazon ECR

Repositories

AWS Marketplace

Discover software

Subscriptions

Task Definitions

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.

Learn more

Create new Task Definition Create new revision Actions

Last updated on April 11, 2019 8:49:12 AM (0m ago)

Status: ACTIVE INACTIVE

Filter in this page Page size 50

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

The screenshot shows the AWS ECS Clusters page. On the left, there's a sidebar with links like Amazon ECS, Clusters, 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 shows a summary for the 'microcosm' cluster: 1 Service, 1 Running tasks, 0 Pending tasks, and EC2 metrics (0 Services, 0 Running tasks, 0 Pending tasks, No data for CPUUtilization and MemoryUtilization, and 0 Container instances). At the bottom, there are 'Create Cluster' and 'Get Started' buttons.

Everything Created By Get Started Wizard

Follow the screen shots for reference

The screenshot shows the AWS CloudFormation Stacks page. The top navigation bar includes 'Services', 'Resource Groups', and 'CloudFormation'. The main content area displays a table of stacks. A message box at the top says 'The redesigned AWS CloudFormation console is available now' and encourages feedback. The table shows one stack: 'EC2ContainerService-microcosm' (Status: CREATE_COMPLETE, Drift Status: NOT_CHECKED), created on 2019-04-11. Below the table, tabs for Overview, Outputs, Resources, Events, Template, Parameters, Tags, Stack Policy, Change Sets, and Rollback Triggers are visible. A message 'Select a stack' is displayed at the bottom.

CloudFormation > Stacks > Stack Detail

EC2ContainerService-microcosm

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

Status reason:
 Termination protection: Disabled
 Drift status: NOT_CHECKED [View details](#)

Last drift check time:
 IAM role:
 Description: AWS CloudFormation template to create a new ECS Fargate First Run stack

Outputs

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

Resources

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

Logical ID	Physical ID	Type	Drift Status	Status	Status Reason
AttachGateway	EC2Co-Attach-1AVZP5KOSFL1P	AWS::EC2::VPCCGatewayAttachment	NOT_CHECKED	CREATE_COMPLETE	
EcsSecurityGroup	sg-06f07d8d65ef6b2fd	AWS::EC2::SecurityGroup	NOT_CHECKED	CREATE_COMPLETE	
InternetGateway	igw-0db5c3b0358004693	AWS::EC2::InternetGateway	NOT_CHECKED	CREATE_COMPLETE	
InternetGateway	igw-0db5c3b0358004693	AWS::EC2::InternetGateway	NOT_CHECKED	CREATE_COMPLETE	
PublicRouteVialgw	EC2Co-Publi-1QU2JZOBMR836	AWS::EC2::Route	NOT_CHECKED	CREATE_COMPLETE	
PublicSubnet1RouteTable...	rtbassoc-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

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

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

08:45:29 UTC-0600 CREATE_IN_PROGRESS AWS::CloudFormation::Stack EC2ContainerService-microcos User Initiated

Template

Parameters

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

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

Template

```

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/

```

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

AWS Services Resource Groups

Clusters : microcosm

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
- Draining service count: 0 Fargate, 0 EC2

Services	Tasks	ECS Instances	Metrics	Scheduled Tasks	Tags			
Create	Update	Delete	Actions	Last updated on April 11, 2019 8:55:29 AM (0m ago)				
<input type="text"/> Filter in this page						Launch type: ALL Service type: ALL		
						1-1 >		
<input type="checkbox"/>	Service Name	Status	Service type	Task Definition	Desired tasks	Running tasks	Launch type	Platform version
<input type="checkbox"/>	jenkins-service	ACTIVE	REPLICA	first-run-task-de...	1	1	FARGATE	LATEST(1.3.0)

Feedback English (US)

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

AWS Services Resource Groups

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

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

Service Health

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

Account Attributes

Resource ID length management

[Report an Issue](#)

Additional Information

VPC Documentation All VPC Resources Forums Report an Issue

Site-to-Site VPN Connections

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

[Create VPN Connection](#)

Feedback English (US)

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

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

Feedback English (US)

https://us-east-2.console.aws.amazon.com/vpc/home?region=us-east-2#vpcs:sort=VpcId

Administrator @ cert-microcosm Ohio Support

Create VPC Actions

Filter by tags and attributes or search by keyword

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-07738971cf93937c	def
vpc-9dd0c3f5		available	172.31.0....	-	dopt-5f946034	rtb-d764bcfc	acl-597f8132	def

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

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

Feedback English (US)

https://us-east-2.console.aws.amazon.com/vpc/home?region=us-east-2#subnets:sort=SubnetId

Administrator @ cert-microcosm Ohio Support

Create subnet Actions

Filter by tags and attributes or search by keyword

Name	Subnet ID	State	VPC	IPv4 CIDR	Available IPv4	IPv6 CIDR	Availability Zone	Availability Zone ID	Rout
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-0ecab74	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

© 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/vpc/home?region=us-east-2#RouteTables:sort=routeTableId

AWS Services Resource Groups

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

Create route table Actions

Filter by tags and attributes or search by keyword

1 to 3 of 3

Name	Route Table ID	Explicitly Associated with	Main	VPC ID	Owner
rtb-089b8ba33139a46fa	-	-	Yes	vpc-0530acf15c751393 ...	443007076818
ECS microc...	rtb-028d4e934944d0fc	2 subnets	No	vpc-0530acf15c751393 ...	443007076818
rtb-d764bbc	-	-	Yes	vpc-9dd0c3f5	443007076818

https://us-east-2.console.aws.amazon.com/vpc/home?region=us-east-2#gws:sort=internetGatewayId

AWS Services Resource Groups

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

Create internet gateway Actions

Filter by tags and attributes or search by keyword

1 to 2 of 2

Name	ID	State	VPC	Owner
ECS microc...	igw-0db5c3b0358...	attached	vpc-0530acf15c7...	443007076818
	igw-d403beb	attached	vpc-9dd0c3f5	443007076818

Select an internet gateway above

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/vpc/home?region=us-east-2#SecurityGroups:sort=groupId

AWS Services Resource Groups

Create security group Actions

Name	Group ID	Group Name	VPC ID	Type	Description	Owner
ECS microc...	sg-06f07d8d65ef1...	EC2ContainerSer...	vpc-0530acf15c7...	EC2-VPC	ECS Allowed Ports	443007076818
	sg-0f84b4941885...	default	vpc-0530acf15c7...	EC2-VPC	default VPC securi...	443007076818
	sg-1e087170	default	vpc-9dd0c3f5	EC2-VPC	default VPC securi...	443007076818

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/ec2/v2/home?region=us-east-2#Home:

AWS Services Resource Groups

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): ● Availability zone is 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

Supported Attributes

- Supported Platforms: VPC
- Default VPC: [vpc-9dd0c3f5](#)
- Resource ID length management
- Console experiments

Additional Information

- Getting Started Guide
- Documentation
- All EC2 Resources
- Forums
- Pricing
- Contact Us

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. On the left, there's a sidebar with links for Amazon ECS, Clusters, Task Definitions (which is selected and highlighted in orange), Amazon EKS, Clusters, Amazon ECR, Repositories, AWS Marketplace, Discover software, and Subscriptions. The main content area has a title 'Task Definitions' with a subtitle explaining what task definitions are. It includes a 'Create new Task Definition' button, a 'Create new revision' button, and an 'Actions' dropdown. A status filter shows '(ACTIVE) INACTIVE'. Below this is a table with one row, showing a task definition named 'first-run-task-definition' with a status of 'ACTIVE'. The table has columns for 'Task Definition' and 'Latest revision status'. At the bottom of the page, there are links for Feedback, English (US), and footer links for Privacy Policy and Terms of Use.

Deleting an ECS Service and Associated Tasks

- Navigate to the ECS Console and select your cluster

Clusters

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

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



Opt in to the new ARN and resource ID format

Amazon ECS has introduced a new format for ARNs and resource IDs. The ARN format is now required for all new clusters and services.

[Configure ECS ARN setting](#)

[Create Cluster](#)

[Get Started](#)

View

list

card

[microcosm >](#)

FARGATE

10

Services

4

Running tasks

EC2

0

Services

0

Running tasks

- Scroll down and select your service

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

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

Clusters > microcosm > Service: cloud-discovery	Update	Delete
Service : cloud-discovery		
Cluster	microcosm	Desired count 1
Status	ACTIVE	Pending count 0
Task definition	cloud-discovery:1	Running count 1

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

Number of tasks 

Minimum healthy percent 

Maximum percent 

[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 ,

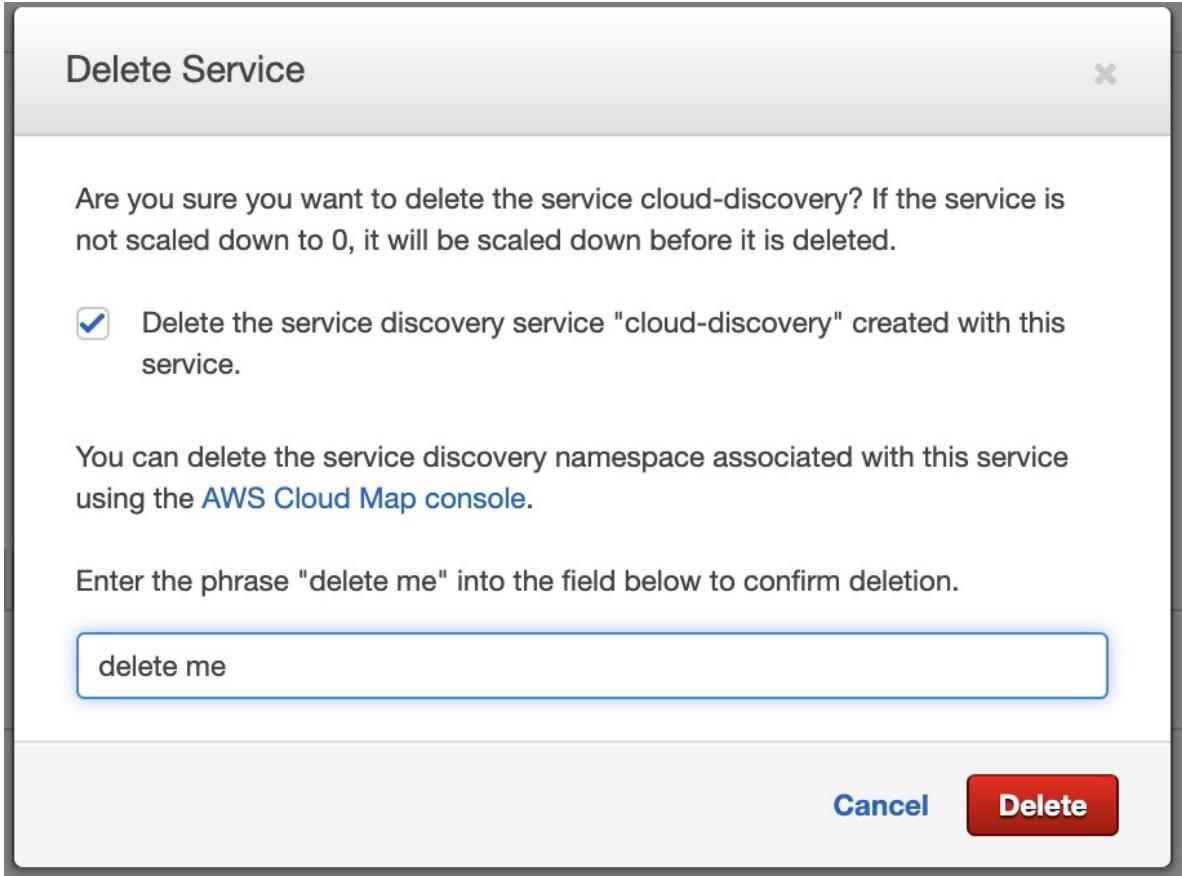
Filter in this page						
Deployment Id	Status	Desired count	Pending count	Running count	Created time	Updated time
ecs-svc/9223370480216347022	PRIMARY	0	0	0	2019-04-30 09:33:48 -0600	2019-05-01 09:15:06 -0600

- Press Delete in the top right corner

[Update](#)

[Delete](#)

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



Notes:

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

[Return to Table of Contents](#)

APPENDIX C

Deploy Stand Alone Microcosm Template

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

[Return to Table of Contents](#)

Deploy Stand Alone Petclinic Deployment Template

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

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

Screenshot of the AWS Lambda console showing the Actions menu for an instance named "tomcat_codedeployTST".

The Actions menu is open, showing the following options:

- Connect
- Get Windows Password
- Create Template From Instance
- Launch More Like This
- Instance State
- Instance Settings (highlighted)
- Image
- Networking
- CloudWatch Monitoring

The Instance Settings option is expanded, showing the following sub-options:

- Add/Edit Tags
- Attach to Auto Scaling Group
- Attach/Replace IAM Role (highlighted)
- 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

Below the Actions menu, the instance details are displayed:

Attribute	Value
AMI ID	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 duration	1 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](#)