

Microcosm on AWS

Table of Contents

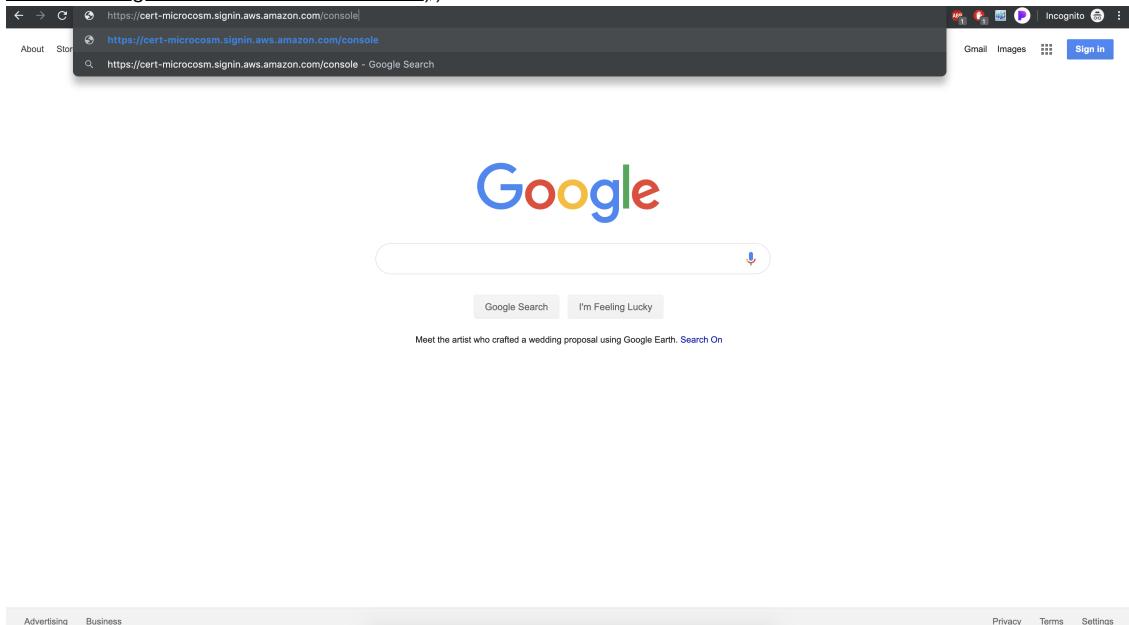
1. [Introduction](#)
2. [Login](#)
3. [Deploy Microcosm using Amazon ECS](#)
 - i. [Clusters](#)
 - ii. [Task Definitions](#)
 - a. [Microcosm Task Definitions](#)
 - iii. [Services](#)
 - iv. [Setting up Jenkins for DevOps Pipeline \(Microcosm\)](#)
 - a. [Blue Ocean Interface](#)
 - b. [Slack & Hubot integration with Jenkins](#)
 - c. [Hubot Setup](#)
 - 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 OwaspZap build step](#)
 - f. [Add Sonatype Nexus build step](#)
 - g. [Slack Integration](#)
 - h. [Nagios Integration](#)
 - i. [Container Security](#)
 - j. [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. [Forward Build Data to AWS Cloud Watch](#)
 - h. [Build and Deploy](#)
 - k. [AWS ECS Based Code Deploy.](#)
 - I. [CloudWatch Dashboard](#)
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](#)
 - iv. [Ansible Deployment](#)
6. [Appendix C](#)
 - i. [Stand Alone Microcosm Template](#)

Introduction

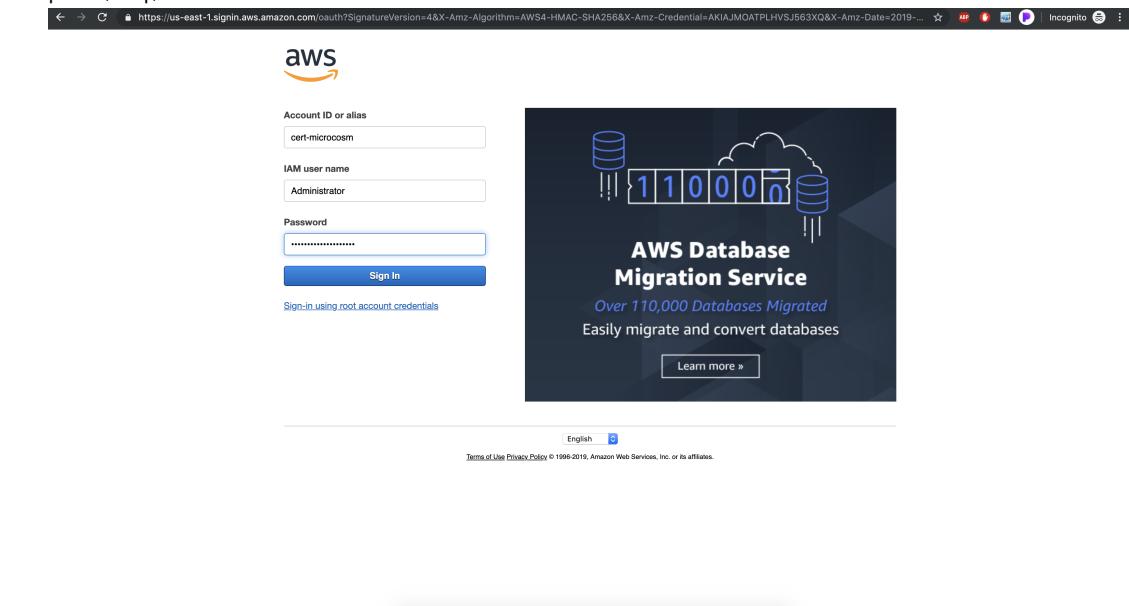
[Return to Table of Contents](#)

Login

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



- Enter the credentials received
 - User: StudentUser_____
 - pwd: \$bzq\$UrFLxw9HfTB-49eRtf!



- Change your password to something you will remember

You must change your password to continue

AWS account 443007076818

IAM user name Administrator

Old password
New password
Retype new password

Confirm password change

[Sign in using root account credentials](#)

English

[Terms of Use](#) [Privacy Policy](#) © 1996-2019, Amazon Web Services, Inc. or its affiliates.

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

AWS Management Console

AWS services

Find Services
You can enter names, keywords or acronyms.

All services

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

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

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

Explore AWS

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

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

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

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

[Return to Table of Contents](#)

Deploy Microcosm using Amazon ECS

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

Clusters

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

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 options like Amazon EKS, Clusters, Task Definitions, Amazon ECR, Repositories, AWS Marketplace, Discover software, and Subscriptions. The main 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 at the bottom lists one cluster: 'microcosm' with 5 services, 6 running tasks, 0 pending tasks, and 0 container instances. The page footer includes links for Feedback, English (US), and various AWS terms.

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

The screenshot shows the 'Create Cluster' wizard at step 1: Select cluster template. It offers three options: 'Networking only', 'EC2 Linux + Networking', and 'EC2 Windows + Networking'. The 'Networking only' option is selected. Each option has a list of resources to be created: 'Cluster', 'VPC (optional)', and 'Subnets' for the first; 'Cluster', 'VPC', 'Subnets', and 'Auto Scaling group with Linux AMI' for the second; and 'Cluster', 'VPC', 'Subnets', and 'Auto Scaling group with Windows 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.

The screenshot shows the 'Create Cluster' wizard at step 2: Configure cluster. It has sections for 'Networking' and 'Tags'. In the Networking section, there's a note about creating a new VPC and a checkbox for 'Create a new VPC for this cluster'. The 'Create VPC' checkbox is unchecked. In the Tags section, there are fields for 'Key' (with 'Add key') and 'Value' (with 'Add value'). At the bottom, there are 'Required', 'Cancel', 'Previous', and 'Create' buttons.

- 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.

Status: ACTIVE

Registered container instances: 0

Pending tasks count	0 Fargate, 0 EC2
Running tasks count	0 Fargate, 0 EC2
Active service count	0 Fargate, 0 EC2
Draining service count	0 Fargate, 0 EC2

[Create](#) [Update](#) [Delete](#) [Actions](#)

Last updated on April 18, 2019 9:51:08 AM (0m ago)

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

i Opt in to the new ARN and resource ID format

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](#).

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

View	list	card
StudentCluster >		
FARGATE		
0 Services	0 Running tasks	0 Pending tasks
EC2		
0 Services	0 Running tasks	0 Pending tasks
No data		No data
CPUUtilization	MemoryUtilization	Container instances
0	0	0
microcosm >		
FARGATE		

Task Definitions

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

Microcosm Task Definitions

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

Task/Service Name	Container Name	Image	Ports	Volumes	Mount Points
jenkins	jenkins	h1kkan/jenkins-docker:lts	8080	jenkins_home	/var/jenkins_home
sonarqube	sonarqube	sonarqube:lts	9000	sonarqube_conf	/opt/sonarqube/conf
				sonarqube_data	/opt/sonarqube/data
				sonarqube_extensions	/opt/sonarqube/extensions
				sonarqube_bundled-plugins	/opt/sonarqube/lib/bundled-plugins
gitlab	gitlab	gitlab/gitlab-ce	443	gitlab-config	/etc/gitlab
			80	gitlab-logs	/var/log/gitlab
				gitlab-data	/var/opt/gitlab
owaspZAP	owaspzap	owasp/zap2docker-stable	8080		
			8090		
hubot	hubot	gillax/hubot-slack-jenkins			
nexus	sonatype_nexus	sonatype/nexus	8081	nexus-data	/sonatype-work
nagios	nagios	jasonrivers/nagios:latest	8080		
			80		
twistlock-fargate	twistlock-fargate	matthewabq/twistlock-fargate:latest	80		
twistlocktest	twistlocktest	twistlocktest/hello-world:latest	80		
cloud-discovery	cloud-discovery	twistlock/cloud-discovery:latest	9083		

[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

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.

[Create new Task Definition](#) [Create new revision](#) [Actions](#)

Status: [ACTIVE](#) [INACTIVE](#)

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

Last updated on April 12, 2019 2:19:48 PM (0m ago) [Edit](#) [Delete](#)

1-6 > Page size 50

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

- Select the Fargate Type and press Next Step

Create new Task Definition

Step 1: Select launch type compatibility
Step 2: Configure task and container definitions

Select launch type compatibility

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

FARGATE

Price based on task size

Requires network mode awsvpc

AWS-managed infrastructure, no Amazon EC2 instances to manage

EC2

Price based on resource usage

Multiple network modes available

Self-managed infrastructure using Amazon EC2 instances

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

- 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 [Edit](#)

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

Network Mode [Edit](#)

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.

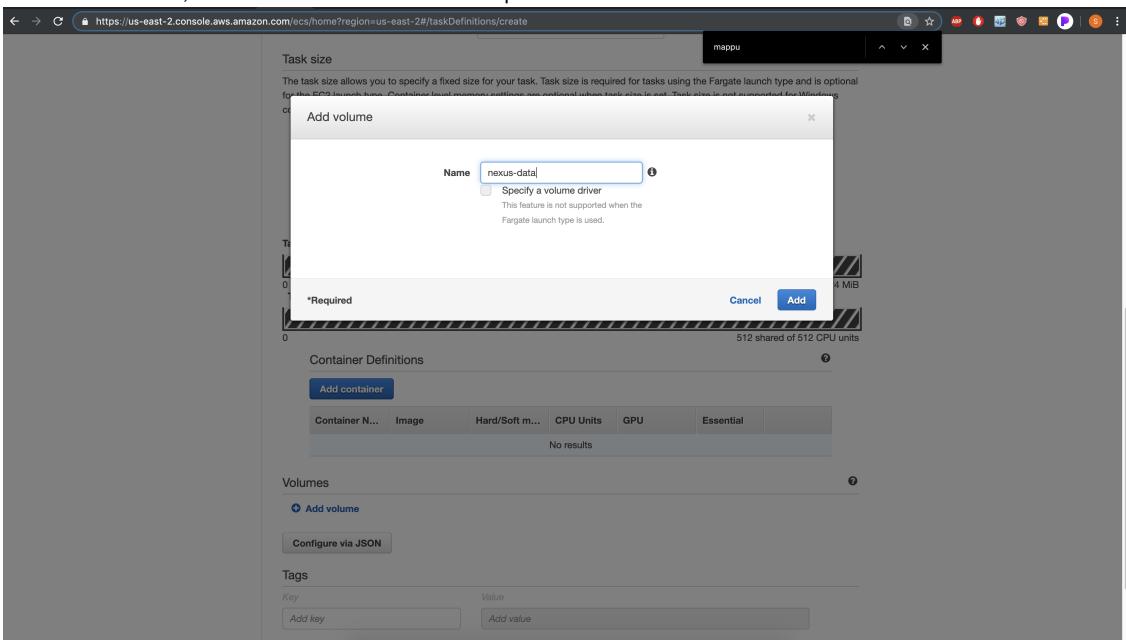
Task execution IAM role

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.

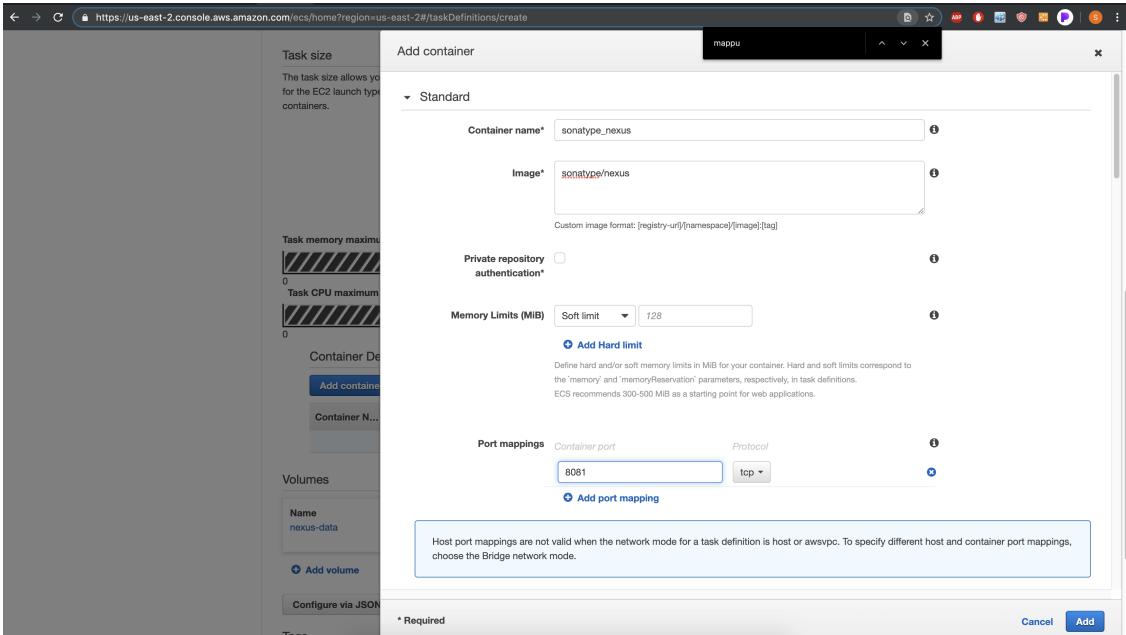
Task execution role [Edit](#)

Task size

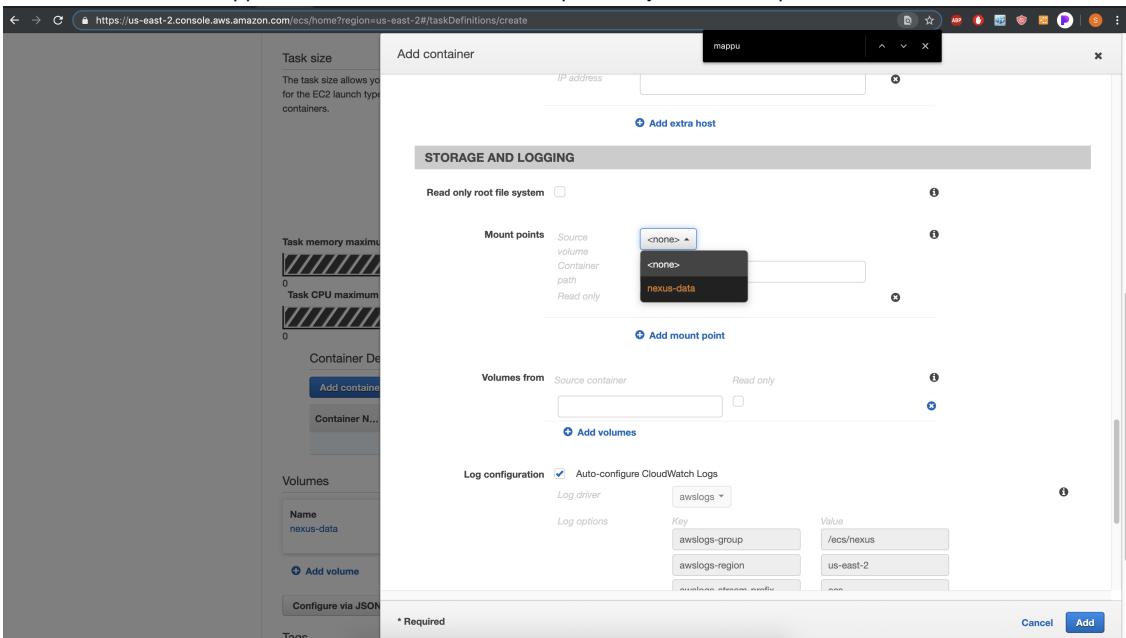
- Select Add Volume, enter the Volume Name and and press add



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



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



- Enter the mount point (s) (Select Add mount point if additional are required).

- Make sure Auto-configure CloudWatch Logs is selected.
- Press Add.

The screenshot shows the 'Add container' dialog for a task definition. In the 'Log configuration' section, the 'Auto-configure CloudWatch Logs' checkbox is checked. The 'Log driver' is set to 'awslogs'. Log options include 'awslogs-group' set to '/ecs/nexus' and 'awslogs-region' set to 'us-east-2'. Other fields like 'Key' and 'Value' are also visible. A note at the bottom states: 'The valid CPU range for 1GB memory is: 0.25 vCPU - 0.5 vCPU'.

- Add Tags if desired and press create

The screenshot shows the final step of creating a task definition. It includes a summary of resources: Task memory maximum allocation (1024 shared of 1024 MiB) and Task CPU maximum allocation (512 shared of 512 CPU units). The 'Container Definitions' section lists a single container named 'sonatype...', which uses the 'sonatype/nexus' image. The 'Volumes' section shows a volume named 'nexus-data'. The 'Tags' section is empty. At the bottom, there are 'Create' and 'Cancel' buttons.

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

The screenshot shows the 'View task definition' page for the 'nexus' task definition. Under 'Launch Status', it says 'Task definition status - 2 of 2 completed'. The 'Create Task Definition: nexus' section shows the output 'nexus succeeded'. The 'Create CloudWatch Log Group' section shows a success message: 'CloudWatch Log Group created CloudWatch Log Group /ecs/nexus'. At the bottom right are 'Back' and 'View task definition' buttons.

- Review the task definition, if desired

The screenshot shows the AWS ECS Task Definition configuration page for a task named 'nexus:1'. The 'Actions' dropdown menu is open, and the 'Create Service' option is selected. Other visible options in the dropdown include 'Run Task', 'Update Service', and 'Deregister'. The main configuration area shows the task definition name as 'nexus', task role as 'ecsTaskExecutionRole', network mode as 'awsvpc', compatibility as 'EC2, FARGATE', and requires compatibility as 'FARGATE'. It also includes sections for 'Task execution IAM role' and 'Task size'.

[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: do not standup/create the Hubot services until after Configuring the hubot plug-in in Slack. Also make sure to replace the bracket portion in the Hubot environment variables with the Jenkins IP address or URL

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 Definition configuration page for a task named 'nexus:1'. The 'Actions' dropdown menu is open, and the 'Create Service' option is selected. Other visible options in the dropdown include 'Run Task', 'Update Service', and 'Deregister'. The main configuration area shows the task definition name as 'nexus', task role as 'ecsTaskExecutionRole', network mode as 'awsvpc', compatibility as 'EC2, FARGATE', and requires compatibility as 'FARGATE'. It also includes sections for 'Task execution IAM role' and 'Task size'.

- 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

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.

Step 1: Configure service

Step 2: Configure network

Step 3: Set Auto Scaling (optional)

Step 4: Review

Launch type: FARGATE EC2

Task Definition: Family: nexus Revision: 1

Platform version: LATEST

Cluster: jenkinsdefault

Service name: sonatype_nexus

Service type*: REPLICA

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

Services ▾ Resource Groups ▾

Administrator @ cert-microcosm ▾ Ohio ▾ Support ▾

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

Cluster VPC*: vpc-0a8f63ff28e39ced (10.0.0.0/24)

Subnets*:

- subnet-0c12a48c3055e0708 (10.0.0.0/24) | ECS Jenkins default - Public Subnet 1 - us-east-2a assign ipv6 on creation: Disabled
- subnet-0c12a48c3055e0708 (10.0.0.0/24) | ECS Jenkins default - Public Subnet 1 - us-east-2a assign ipv6 on creation: Disabled
- subnet-0a93e6cb145176367 (10.0.1.0/24) | ECS Jenkins default - Public Subnet 2 - us-east-2b assign ipv6 on creation: Disabled

Security groups*:

- ECS Jenkins default

Auto-assign public IP: Enabled

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

Configure network

VPC Id: vpc-0a8f63ffd28e39ced
Subnets: subnet-0c12a48c3055e9708
Create new security group: sonaty-6743
Auto assign IP: ENABLED

Configure service discovery

Namespace: ns-iz26fljy45cjaqgg
Service discovery name: sonatype_nexus
Enable ECS task health propagation: true
DNS record type and TTL: A 60

Set Auto Scaling (optional)

not configured

Create Service

- Once the service is successfully created, select View Service

ECS Service status - 4 of 4 completed

Configure Task Networking

Create security group

- ✓ Create security group
sonaty-6743 succeeded sg-08677ef9fb7f9e049

Set inbound rules

- ✓ Set inbound rules
succeeded sg-08677ef9fb7f9e049

Configure Service Discovery

Create service discovery service

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

Create Service

Create service: sonatype_nexus

- ✓ Service created
Service created. Tasks will start momentarily. View: sonatype_nexus

Additional integrations you can connect to your ECS service

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

The screenshot shows the AWS CloudWatch Metrics interface. On the left, the navigation menu includes 'Clusters' under 'Amazon ECS'. The main content area displays a task named 'Task : 325b875c-9e8f-4686-872a-937c986973d7'. The 'Details' tab is selected, showing information such as Cluster ('jenkinsdefault'), Launch type ('FARGATE'), Platform version ('1.3.0'), Task definition ('nexus:1'), Group ('service:sonatype_nexus'), Task role ('ecsTaskExecutionRole'), Last status ('ACTIVATING'), Desired status ('RUNNING'), and Created at ('2019-04-12 14:25:35 -0600'). Below this is the 'Network' section, which lists Network mode ('awsvpc'), ENI Id ('eni-070590cb8cb942258'), Subnet Id ('subnet-0c12a4bc3055e9708'), Private IP ('10.0.0.86'), Public IP ('13.58.236.137'), and Mac address ('02:38:3b:e6:c3:da'). The 'Containers' section shows a single container named 'sonatype_nexus' with Container Id 'a7474130-6432-44a7-8105-d79153c33a3a', Status 'RUNNING', Image 'sonatype/nexus', CPU Units '0', Hard/Soft memory '.../...', and Essential 'true'. A note at the bottom right indicates the data 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 CloudWatch Metrics interface. On the left, the navigation menu includes 'Clusters' under 'Amazon ECS'. The main content area displays two clusters: 'jenkinsdefault' and 'microcosm'. The 'jenkinsdefault' cluster is shown with 6 Services, 5 Running tasks, and 1 Pending task. The 'EC2' section shows 0 Services, 0 Running tasks, 0 Pending tasks, No data for CPUUtilization, No data for MemoryUtilization, and 0 Container instances. The 'microcosm' cluster is shown with FARGATE settings. Below the clusters, there is a 'Create Cluster' button and a 'Get Started' button.

- 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 CloudWatch Metrics interface. On the left, the navigation menu includes 'Clusters' under 'Amazon ECS'. The main content area displays the 'jenkinsdefault' cluster with a status of 'ACTIVE'. It shows registered container instances (0), pending tasks (0 Fargate, 0 EC2), running tasks (6 Fargate, 0 EC2), active service count (6 Fargate, 0 EC2), and draining service count (0 Fargate, 0 EC2). Below this, the 'Tasks' tab is selected, showing a table of tasks. The table includes columns for Task, Task definition, Container instan..., Last status, Desired status, Started By, Group, Launch type, and Platform version. Tasks listed include '108c5928-1525-4...', '13c3e227-2949-4...', '325b875c-9e8f-4...', '62af7548-c047-4...', '89fd2d92-90fc-4...', and 'caaa3991-accd-4...'. Each task row has a checkbox and a link to its logs. A note at the bottom right indicates the data was last updated on April 12, 2019, at 3:44:52 PM (0m ago).

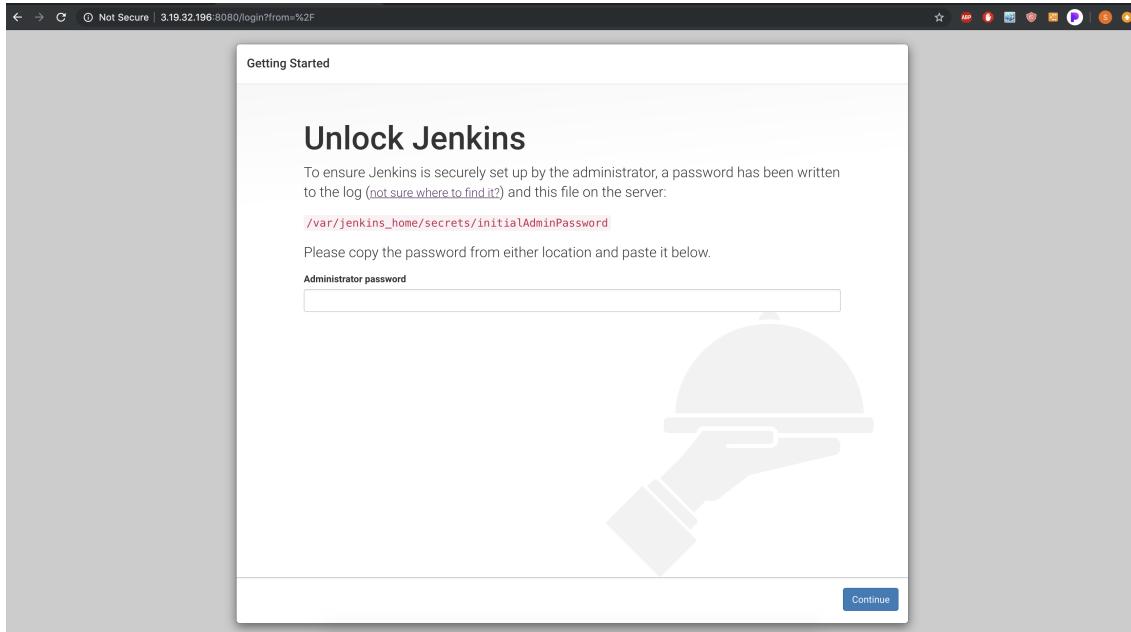
[Return to Table of Contents](#)

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/jenkins
- 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

A screenshot of the AWS CloudWatch Log Groups interface. The left sidebar shows 'Logs' selected. The main area shows 'Streams for /ecs/jenkins'. A search bar at the top says 'Search Log Group'. A table lists log streams with columns for 'Log Stream' and 'Last Event Time'. The table is sorted by 'Last Event Time' in descending order. The most recent log stream listed is 'ecs/jenkins/92c59a81-6a6a-45e0-bc7b-3932928fdfa9' with a timestamp of '2019-04-16 13:27 UTC-6'. Other streams listed include 'dd0e962c-acaa-43e5-b514-70cd84adfd06', '234135b-5a0d-40a5-a8bc-2c96a5977d03', and many others with timestamps ranging from April 16, 2019, to April 15, 2019.

- 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

CloudWatch Dashboards CloudWatch > Log Groups > /ecs/jenkins > ecs/jenkins/92c59a81-6a6a-45e0-bc7b-3932928fdfa9... Expand all Row Text Filter events all 2019-04-15 (19:27:55) Time (UTC +00:00) Message

2019-04-16

19:28:31 Apr 16, 2019 7:28:31 PM org.springframework.context.support.AbstractApplicationContext prepareRefresh

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

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 Apr 16, 2019 7:28:31 PM org.springframework.beans.factory.support.DefaultListableBeanFactory@10118f44: defining beans [authenticationManager]; root of factory hierarchy

19:28:32 Apr 16, 2019 7:28:32 PM org.springframework.context.support.AbstractApplicationContext prepareRefresh

INFO Refreshing org.springframework.web.context.support.StaticWebApplicationContext@23803935; 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

INFO Bean factory for application context [org.springframework.web.context.support.StaticWebApplicationContext@23803935]: 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 Apr 16, 2019 7:28:33 PM jenkins.install.SetupWizard init

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:

47d0d8ea244d55b030ad2cd5ad78db

47d0d8ea244d55b030ad2cd5ad78db

19:28:33 This may also be found at: /var/jenkins_home/secrets/initialAdminPassword

19:28:33 ****

19:28:33 ****

19:28:33 ****

19:28:40 --> setting agent port for jnlp

19:28:41 --> setting agent port for jnlp... done

19:28:48 Apr 16, 2019 7:28:48 PM hudson.model.UpdateSite updateData

19:28:48 INFO: Obtained the latest update center data file for UpdateSource default

- Click Install suggested plugins

The screenshot shows the Jenkins 'Getting Started' page. The title bar says 'Getting Started'. The main heading is 'Customize Jenkins' with the subtext 'Plugins extend Jenkins with additional features to support many different needs.' Below this are two options: 'Install suggested plugins' (selected, highlighted in blue) and 'Select plugins to install'. A large background image of a hand holding a tablet is visible.

- Wait while the plugins are installed

- Fill in the information for the admin user. Make sure to remember your password

- Select Save and Continue

Getting Started

Create First Admin User

Username:

Password:

Confirm password:

Full name:

E-mail address:

Jenkins 2.164.1 [Continue as admin](#) [Save and Continue](#)

- Select Save and Finish

Getting Started

Instance Configuration

Jenkins URL:

The Jenkins URL is used to provide the root URL for absolute links to various Jenkins resources. That means this value is required for proper operation of many Jenkins features including email notifications, PR status updates, and the BUILD_URL environment variable provided to build steps.

The proposed default value shown is **not saved yet** and is generated from the current request, if possible. The best practice is to set this value to the URL that users are expected to use. This will avoid confusion when sharing or viewing links.

Jenkins 2.164.1 [Not now](#) [Save and Finish](#)

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

Getting Started

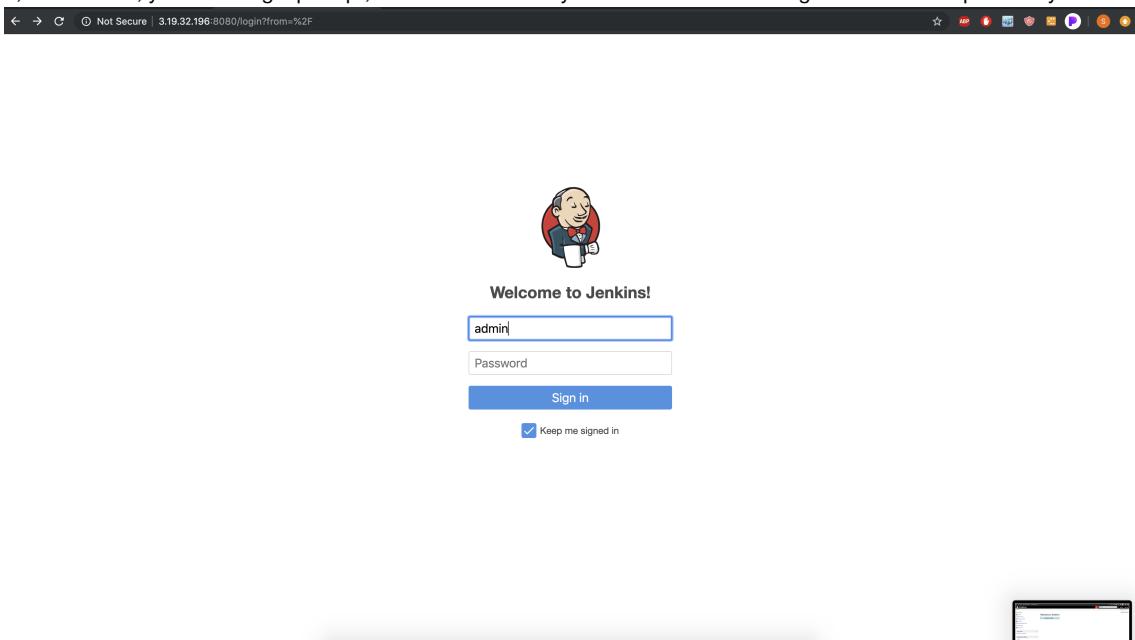
Jenkins is almost ready!

Your Jenkins setup is complete, but some plugins require Jenkins to be restarted.

[Restart](#)

Jenkins 2.164.1

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



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

The screenshot shows the Jenkins 2.164.2 dashboard. At the top, there's a banner indicating a new version (2.164.2) is available for download. Below this, a warning message states that multiple security vulnerabilities have been published for currently installed components. A link to the plugin manager and a button to configure warnings are provided. The left sidebar contains links for New Item, People, Build History, Manage Jenkins, My Views, Lockable Resources, Credentials, and New View. The main content area features a "Welcome to Jenkins" header and a message to create new jobs. It lists several security vulnerabilities across different Jenkins components: Jenkins core, Lockable Resources, Script Security, Pipeline, EnvInject, and Build Queue.

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

Warnings have been published for the following currently installed components:

- [Go to plugin manager](#)
- [Configure which of these warnings are shown](#)

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 1.55: SandBox Sandbox bypass](#)

Pipeline Groovy 2.64:
[Script Security sandbox bypass](#)

EnvInject Plugin 2.1.6:
[Exposure of sensitive build variables stored by Envinject 1.90 and earlier](#)

Build Queue
No builds in the queue.

Build Executor Status
1 Idle
2 Idle

[Manage Jenkins](#)

- 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. The title bar indicates the URL is 3.19.32.196:8080/pluginManager/. The main content area has a header "Updates" with tabs for Available, Installed, and Advanced. A search bar is at the top right. Below is a table of available plugins:

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

At the bottom are buttons for "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: “OWASP ZAP Jenkins”
- search & select: “Git Plugin” (May be pre-installed)
- search & select: “Pipeline Maven Integration”
- search & select: “Ansible Plugin” (May be pre-installed)
- search & select: “Custom Tools”
- search & select: “Summary Display”
- search & select: “Selenium HTML Report”
- search & select: “HTML Publisher”
- search & select: “Slack Notification”
- search & select: “SonarQube Scanner”
- search & select: “Nexus Platform”
- search & select: “AWS CodeDeploy Plugin for Jenkins”
- search & select: “Blue Ocean”. - Feel free to add all of the Blue Ocean plugins for additional functionality
- search & select: “AWS CloudWatch Logs Publisher”
- click “install without restart” at bottom of page
- check box next to “Restart Jenkins when installation is complete and no jobs are running.”
- at top-left menu, click “back to Dashboard”

The screenshot shows the Jenkins Plugin Manager interface with the "Available" tab selected. The main content area has a header "summary di" with a search bar. Below is a table of available plugins:

Install	Name	Version
<input type="checkbox"/>	For a given Parameterized Project, this plugin shows the builds sorted by the parameters used to execute the builds.	1.5
<input type="checkbox"/>	SLOCCount	1.24
<input type="checkbox"/>	SonarQube Scanner	2.8.1
<input checked="" type="checkbox"/>	SonarQube Integration	2.2.1
<input type="checkbox"/>	SonarGraph	1.6.4
<input type="checkbox"/>	Spelunker	1.7.1
<input type="checkbox"/>	Statistics Gatherer	2.0.3
<input type="checkbox"/>	StepCounter	2.0.0
<input checked="" type="checkbox"/>	Summary Display	1.15
<input type="checkbox"/>	TAP	2.2.2
<input type="checkbox"/>	Task Scanner	4.53
<input type="checkbox"/>	Tattletale	0.3
<input type="checkbox"/>	Test Results Analyzer	0.3.5
<input type="checkbox"/>	Test stability history	2.3
<input type="checkbox"/>	Testability Explorer	0.4
<input type="checkbox"/>	TestComplete support	2.1

At the bottom are buttons for "Install without restart", "Download now and install after restart", "Update information obtained: 14 min ago", and "Check now".

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

Back to Dashboard
 Manage Jenkins
 Manage Plugins

Installing Plugins/Upgrades

Preparation

- Checking internet connectivity
- Checking update center connectivity
- Success

Nexus Platform

Nexus Platform

- Slack Notification
- Config File Provider
- Pipeline Maven Integration
- HTML Publisher
- Official OWASP ZAP
- Summary Display
- SonarQube Scanner
- Extended Choice Parameter
- Custom Tools
- Selenium HTML report
- Restarting Jenkins

Go back to the top page

(you can start using the installed plugins right away)

Restart Jenkins when installation is complete and no jobs are running

Page generated: Apr 16, 2019 7:51:39 PM UTC REST API Jenkins ver. 2.164.1

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

← → C O Not Secure | 3.19.32.196:8080/manage Jenkins search admin log out ENABLE AUTO REFRESH

Manage Jenkins

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

Warnings have been published for the following currently installed components.

Go to plugin manager Configure which of these warnings are shown

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 2.4: 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

Configure System
Configure global settings and paths.

Configure Global Security
Secure Jenkins; define who is allowed to access/use the system.

Configure Credentials
Configure the credential providers and types.

Global Tool Configuration
Configure tools, their locations and automatic installers.

- 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

Maven installations

Add Maven

Name: petclinic

Install automatically

Install from Apache

Version: 3.6.1

Delete Installer

Save Apply

- Create Custom Tool for Owasp Zap Plugin
 - Scroll up to the Cutom Tool Section
 - click “Custom Tool Installations”
 - click “Add Custom tool”
 - enter “ZAP_2.6.0” in the “name” field
 - click the “Install automatically” checkbox
 - enter “https://github.com/zaproxy/zaproxy/releases/download/2.6.0/ZAP_2.6.0_Linux.tar.gz” (https://github.com/zaproxy/zaproxy/releases/download/2.6.0/ZAP_2.6.0_Linux.tar.gz) in the “Download URL for binary archive” field
 - enter “ZAP_2.6.0” in the “Subdirectory of extracted archive” field

- Click Apply

- Click Save

Trigger downstream upon result

Success Unstable Failure Not Built Aborted

Options

Add Publisher Options

Custom tool

Custom tool installations

Add Custom tool

Custom tool

Name: ZAP_2.6.0

Custom Tool Configuration...

Install automatically

Extract *.zip*.*tar.gz

Label

Download URL for binary archive: https://github.com/zaproxy/zaproxy/releases/download/2.6.0/ZAP_2.6.0_Linux.tar.gz

Subdirectory of extracted archive: ZAP_2.6.0

Add Installer

Delete Custom tool

Add Custom tool

JDK

JDK installations

Add JDK

Save Apply

Blue Ocean Interface

The Blue Ocean interface is an updated look and feel for Jenkins. Feel free to explore it and using it features.

- Open Blue Ocean from the left side-bar

- Alternately, you can navigate directly to the Blue Ocean interface at [http://\[gitlab ip\]:8080/blue](http://[gitlab ip]:8080/blue)

The screenshot shows the Jenkins interface with a dark header featuring the Jenkins logo and the word "Jenkins". Below the header is a light gray navigation bar with the word "Jenkins" and a right-pointing arrow. The main content area contains a list of links with icons:

- New Item** (file icon)
- People** (people icon)
- Build History** (document icon)
- Project Relationship** (magnifying glass icon)
- Check File Fingerprint** (key icon)
- Manage Jenkins** (gear icon)
- My Views** (eye icon)
- Open Blue Ocean** (blue circle icon)

- Explore the Blue Ocean interface

The screenshot shows the Jenkins Blue Ocean interface. At the top is a blue header with the Jenkins logo, a "Pipelines" button (which is highlighted), "Executors", "Administration", a user icon, and a "Logout" button. Below the header is a search bar with the placeholder "search pipelines...". To the right is a "New Pipeline" button. The main area is a table with columns: NAME, HEALTH, BRANCHES, and PR. One row is visible for the pipeline "petclinic", which has a yellow star icon under HEALTH.

NAME	HEALTH	BRANCHES	PR
petclinic		-	-

Slack & Hubot integration with Jenkins

Create free Slack workspace and add Hubot/Jenkins Slack apps

- Navigate to "<https://slack.com/get-started>" (<https://slack.com/get-started>) and follow the instructions to create a new Slack workspace if you don't have one already

The screenshot shows the Slack workspace creation interface. At the top is a dark header with the Slack logo and a search bar. Below the header is a large central box with the heading "Start with a workspace". It contains two main buttons: "Find your Slack workspace" (with a magnifying glass icon) and "Create a new workspace" (with a plus sign icon). Below these buttons is an illustration of a person sitting at a desk with a computer monitor displaying the Slack interface.

- Enter your email address

https://slack.com/get-started#create

slack

Create a new workspace

To make a workspace from scratch, please confirm your email address.

hame@example.com

Confirm

Please fill in your email.

slack

- Pick a name for your team

https://slack.com/create#teamname

slack

What's the name of your company or team?

SampleCompany

Next

By continuing, you're agreeing to our Customer Terms of Service, Privacy Policy, and Cookie Policy.

slack

- Enter a meaningful name for your project, “build” or “petclinic” may be appropriate for this tutorial.

https://slack.com/create#channelname

slack

What's a project your team is working on?

SomethingInteresting

Next

TheSinc

slack

- Select “Skip for now”

The screenshot shows the Slack workspace creation interface. On the left, there's a section titled "Who else is working on this project?" with three input fields containing "name@example.com". Below these is a link to "Add another" and a button to "Add Teammates". There's also a link to "Get an invite link to share" and a "skip for now" option. On the right, a sidebar shows the "TheSinc" channel and a preview of the "#sometinginteresting" channel.

- Select See Your Channel in Slack

The screenshot shows the Slack workspace creation interface. It features a green button labeled "See Your Channel in Slack". To its left, text reads "Tada! Meet your team's first channel: #sometinginteresting". Below this, a smaller text says "A channel brings together every part of your project – the people, conversations, ideas, updates, and files – so your team can move forward and get more done." On the right, a sidebar shows the "TheSinc" channel and a preview of the "#sometinginteresting" channel.

- Select Finish Signing Up

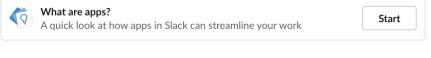
The screenshot shows the Slack workspace creation interface. At the top, it displays the URL ".slack.com/messages/CHXJDQAY/" and the channel name "#sometinginteresting". Below this, there are icons for stars, messages, and a plus sign, followed by "Add a topic". A central image shows a person pointing at a screen. Below the image, text reads "Here it is! The first channel in your team's Slack workspace". A note below states: "You can send messages here right away – but to get the most out of Slack, the next step is to save your account and invite a few people to join the conversation." A green "Finish Signing Up" button is at the bottom.

Today

- Once a workspace has been created, navigate to the “browse apps” page within your workspace (Plus button next to the word Apps in the left sidebar)

[Manage apps...](#)X
esc

Browse Apps

[View App Directory](#) Search by name or category (e.g. productivity, sales)

Add apps to your workspace

 Giphy	An online library of animated GIFs	Install
 Google Drive	Get notifications about Google Drive files within Slack.	Install
 Trello	Collaborate on Trello projects without leaving Slack.	Install
 Dropbox	Cloud file storage and syncing	Install
 Simple Poll	Create native and simple polls in Slack.	Install
 Google Calendar for Team Events	A shared calendar for your team.	Install
 Twitter	Bring tweets into Slack.	Install
 Google+ Hangouts		Install

- Search for “Hubot” and select Install

[Manage apps...](#)X
esc

Browse Apps

[View App Directory](#) hubot

From the App Directory

 Hubot	GitHub's scriptable chat bot.	Install
-----------------------------------------------------------------------------------------	-------------------------------	-------------------------

- Select Install

slack | [App Directory](#) Search App Directory [Browse](#) [Manage](#) [Build](#) 

[Browse Apps](#)

Hubot

[App Info](#) [Settings](#)

Hubot is an open source chat robot for your company that's easy to program using simple scripts written in CoffeeScript and runs on Node.js.
This integration will allow your Hubot instance to connect and interact with your Slack workspace.

[Install](#) [App Homepage](#)

[App help](#)

[Terms](#)

Categories: [Developer Tools](#) [Bots](#) [Essential Apps](#)

[Report this app to Slack for inappropriate content or behavior.](#)

[USING SLACK](#) [SLACK ▾](#) [LEGAL](#) [HANDY LINKS](#)

Product Jobs Privacy Download desktop app

- Choose a username for your Hubot instance and select Add Hubot Integration

The screenshot shows the Slack App Directory with the URL <https://thesinc.slack.com/apps/new/A07XDU93-hubot>. The page title is "Hubot" and the sub-page title is "New configuration". The main content area contains the following information:

- Hubot**: GitHub's scriptable chat bot.
- Hubot is an open source chat robot for your company that's easy to program using simple scripts written in CoffeeScript and runs on Node.js.
- This integration will allow your Hubot instance to connect and interact with your Slack workspace.

Below this, there is a form field labeled "Username" with the placeholder "Start by choosing a username for your hubot" and a text input field. A large green button labeled "Add Hubot Integration" is centered below the input field. At the bottom of the form, a small note states: "By creating a hubot integration, you agree to the [Slack API Terms of Service](#)".

At the bottom of the page, there are navigation links for "USING SLACK", "SLACK", "LEGAL", and "HANDY LINKS".

- Copy and save the “HUBOT_SLACK_TOKEN” (that begins with xoxb) from the Setup Instructions. This will be need to be included with initiating the Hubot bot from the command line to connect your local Hubot bot to the Slack workspace
- **IMPORTANT** return to the ECS task definitions, and create a revision of the hubot definition. Paste the token in the environment variables (replace the bracket portion “[REPLACE_WITH_SLACK_TOKEN]”)

The screenshot shows the "Setup Instructions" section of the Hubot integration setup page. It includes the following text:

Hubot is an open source chat robot for your company that's easy to program using simple scripts written in CoffeeScript and runs on Node.js.
This integration will allow your Hubot instance to connect and interact with your Slack workspace.

Below this, the "Integration Settings" section is shown. It has a sub-section for "API Token" with the text: "The Hubot adapter needs an API token." and a text input field containing the value "xoxb-613745337111-603893166177-44B28UUt26u8FpYGfjL99Do6". A "Regenerate" link is located to the right of the input field.

IMPORTANT

- Repeat this process to install the slack Jenkins CI plugin
- After clicking Install on the Jenkins plugin page, choose or create a new channel for Jenkins notifications

The screenshot shows the "Jenkins CI" integration setup page. The main content area contains the following information:

Jenkins CI: An open source continuous integration server.

Jenkins CI is a customizable continuous integration server with over 600 plugins, allowing you to configure it to meet your needs.

This integration will post build notifications to a channel in Slack.

Below this, the "Post to Channel" section is shown. It has a text input field containing "# sometinginteresting" and a dropdown menu icon. A link "or create a new channel" is located to the right of the input field. A large green button labeled "Add Jenkins CI integration" is at the bottom of the section.

- Follow the on screen Setup Instructions
- For further guidance see [ref \(<https://www.youtube.com/watch?v=TWwvxn2-J7E>\)](https://www.youtube.com/watch?v=TWwvxn2-J7E)

- When finished press Save Settings on the Slack Jenkins page

Browse Apps > Jenkins CI > Edit configuration

Jenkins CI
Added by morleys2 on April 19th, 2019

Disable • Remove

Jenkins CI is a customizable continuous integration server with over 600 plugins, allowing you to configure it to meet your needs.

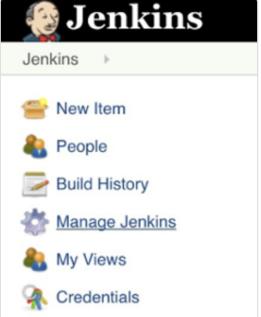
This integration will post build notifications to a channel in Slack.

Setup Instructions

Here are the steps necessary to add the Jenkins CI integration.

Note: These instructions are for v2.2. To install an older version, go down to [Previous Setup Instructions](#).

Step 1 In your Jenkins dashboard, click on Manage Jenkins from the left navigation.



Step 2 Click on Manage Plugins and search for the Slack Notification Plugin in the Available tab. Make

NOTE If the Integration Token field is missing in the Global Slack Notifier Settings section, follow the following instructions

- Enter the URL from Slack in the Slack Compatible app URL (Optional) field
- In the Integration Token Credential ID section, select Add -> Jenkins

Global Slack Notifier Settings

Slack compatible app URL (optional)	<input type="text" value="https://[REDACTED].slack.com/services/hooks/jenkins-ci/"/>
Team Subdomain	<input type="text"/>
Integration Token Credential ID	<input type="button" value="- none -"/> <input type="button" value="Add"/>
Is Bot User?	<input type="checkbox"/>
Channel or Slack ID	<input type="text"/>

Client error : the token with the provided ID could not be found and no token was specified

- Under Kind, select Secret Text, paste the Integration Token in the Secret field, Enter an ID and Description and press Add

Jenkins Credentials Provider: Jenkins

Add Credentials

Domain	Global credentials (unrestricted)
Kind	Secret text
Scope	Global (Jenkins, nodes, items, all child items, etc)
Secret
ID	SlackIntegrationToken
Description	SlackIntegrationToken

- In the Integration Token Credential ID section, select the ID you entered from the drop down next to the Add button.
- Enter the channel to send messages to (eg #sometinginteresting)

- Press Test Connection. If you see Success, congratulations, if not, make sure you followed the steps correctly.

Global Slack Notifier Settings

Slack compatible app URL (optional)	<input type="text" value="https://████████.slack.com/services/hooks/jenkins-ci"/>	
Team Subdomain	<input type="text"/>	
Integration Token Credential ID	<input type="text" value="SlackIntegrationToken"/>	
Is Bot User?	<input type="checkbox"/>	
Channel or Slack ID	<input type="text" value="#somethinginteresting"/>	

- Double check the connection by going to Slack and viewing the relevant channel. (The example below shows both the connection messages and a successful build.)

The screenshot shows a Slack workspace interface. On the left, there's a sidebar with sections for Channels (# everyone, # random, # somethinginteresting), Direct Messages (Slackbot, morleys2 (you)), and Apps (hubot_morley). In the main area, under the # somethinginteresting channel, a message from 'jenkins' is listed with the timestamp '9:08 AM'. The message content is: 'added an integration to this channel: jenkins'. Below this, several messages from 'morleys2' are shown at different times (9:09 AM, 9:10 AM, 9:11 AM) indicating the removal and addition of the same integration. At the bottom of the list, a message from 'Jenkins' at '9:42 AM' says: 'Slack/Jenkins plugin: you're all set on http://3.18.104.142:8080/'. Below this, two more messages from 'petclinic2' are shown: '#7 Started by user admin (Open)' and '#7 Success after 33 sec (Open)'.

Hubot

In the context of the DevOps pipeline, Hubot can be used to start builds within Jenkins from Slack. The purpose of including it here is to provide setup/integration instructions, but it is not used in this tutorial for the purpose of initiating builds. See documentation at <https://hubot.github.com/> (<https://hubot.github.com/>) for further information on using and integrating Hubot in your pipeline for increased automation.

[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> (<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

Repositories
Your most active repositories will appear here.
[Create a repository or explore repositories.](#)

Welcome to the new closer to the stuff you care about

Discover repositories

- Microsoft/AppCenter-SDK Development repository for and macOS
- Objective-C ★ 158
- lukesampson/scoop-extras "Extras" bucket for Scoop
- PowerShell ★ 478

Help **Settings** **Sign out**

mozilla/DeepSpeech
A TensorFlow implementation of Baidu's DeepSpeech architecture

C++ ★ 9.9k

Go to Explore →

Learn Git and GitHub without any code!

Using the Hello World guide, you'll create a repository, start a branch, write comments, and open a pull request.

[Read the guide](#) [Start a project](#)

Discover interesting projects and people to populate your personal news feed.

Your news feed helps you keep up with recent activity on repositories you [watch](#) and people you follow.

[Explore GitHub](#)

© 2019 GitHub, Inc.

Blog API Terms

About Training Privacy

Shop Status Help

Contact GitHub

- Select Developer settings from the left menu list

Personal settings

- Profile** (selected)
- Account
- Emails
- Notifications
- Billing
- SSH and GPG keys
- Security
- Sessions
- Blocked users
- Repositories
- Organizations
- Saved replies
- Applications
- Developer settings**

Public profile

Name

Public email

Select a verified email to display ↗

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

Bio

Tell us a little bit about yourself

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

URL

Company

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

Location

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

[Update profile](#)

- Select Person Access Tokens

- Click Generate new token

The screenshot shows the GitHub developer settings interface. The top navigation bar includes links for Search or jump to..., Pull requests, Issues, Marketplace, and Explore. Below this, the Settings / Developer settings page is shown. A sidebar on the left lists OAuth Apps, GitHub Apps, and Personal access tokens, with Personal access tokens currently selected. The main content area is titled "Personal access tokens" and contains instructions: "Need an API token for scripts or testing? Generate a personal access token for quick access to the GitHub API." It explains that personal access tokens function like ordinary OAuth access tokens but can be used instead of a password for Git over HTTPS or for basic authentication. A "Generate new token" button is located in the top right corner.

- 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". A "What's this token for?" link is present. The "Select scopes" section contains a table of OAuth scopes with checkboxes. The "repo" scope is checked, while others like "public_repo" and "gist" are also available. Other scopes listed include admin:org, admin:public_key, admin:repo_hook, admin:org_hook, and admin:org_hook.

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

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

The screenshot shows the GitHub repository page for "SLS-ALL / spring-petclinic". The top navigation bar includes links for Search or jump to..., Pull requests, Issues, Marketplace, and Explore. The repository details are shown, including a star count of 3, a fork count of 6,056, and a watch count of 2. The repository description is "A sample Spring-based application". The main content area shows the repository structure with branches (master), releases (0), and contributors (17). A table of recent commits is displayed, with the latest commit being "Merge pull request #2 from kingsman142/master" by "kontostathisk" on Dec 12, 2017. The commits list includes various changes to files like cookbook, environments, src, .gitignore, .springBeans, .temp, Vagrantfile, deploy.yml, pom.xml, pom_provision_demo.xml, provision.sh, readme.md, solo.json, and solo.rb.

Commit	Author	Date
Merge pull request #2 from kingsman142/master	kontostathisk	Latest commit 38a3a41 on Dec 12, 2017
removed 'bash' resource from petclinic cookbook - added in a template...		4 years ago
removed 'bash' resource from petclinic cookbook - added in a template...		4 years ago
Set for demo - Jenkins-		2 years ago
added .DS_Store to .gitignore		2 years ago
using latest versions of hibernate, spring-data, joda...		5 years ago
Added empty file.		4 years ago
Commented out chef provisioning. Also, added ubuntu/trusty64 box for ...		4 years ago
added firewall rule for 8080/tcp to deploy.yml		2 years ago
Missed a section when commenting out...		4 years ago
minimized shell scripting demo. updated readme		4 years ago
Parallelized server setup, added option to modify security groups, an...		a year ago
Updated readme to indicate where to download resources to usbstick.		4 years ago
removed 'bash' resource from petclinic cookbook - added in a template...		4 years ago
minimized shell scripting demo. updated readme		4 years ago

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

A sample Spring-based application

Manage topics

429 commits 6 branches 0 releases 17 contributors

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

This branch is even with SLS-ALL:master.

kontostathisk Merge pull request SLS-ALL#2 from kingsman142/master

Latest commit 38a3a41 on Dec 12, 2017

File	Description	Age
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](#)

- 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.

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’

The screenshot shows the 'New project' page on GitLab. At the top, there are three tabs: 'Blank project', 'Create from template', and 'Import project'. The 'Import project' tab is selected. Below it, under 'Import project from', the 'GitHub' tab is selected. Other options include 'GitLab export', 'Bitbucket Cloud', 'Bitbucket Server', 'GitLab.com', 'Google Code', 'Fogbugz', 'Gitea', 'Repo by URL', and 'Manifest file'. A note at the bottom says: 'Tip: You can also create a project from the command line. Show command'.

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

The screenshot shows the 'GitHub import' page on GitLab. It displays a table titled 'Import repositories from GitHub' with one row. The row shows 'From GitHub' as 'morleys/spring-petclinic', 'To GitLab' as 'root/spring-petclinic', and 'Status' as 'Done'. There is a button 'Go to project' next to the status. A note at the top right says 'Import all repositories'.

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

The screenshot shows the 'spring-petclinic' project details page on GitLab. The sidebar has sections like 'Project', 'Repository', 'Issues', 'Merge Requests', 'CI / CD', 'Operations', 'Wiki', 'Snippets', and 'Settings'. The main area shows project statistics: '429 Commits', '6 Branches', '0 Tags', and '0 Bytes Files'. It also shows a sample application: 'A sample Spring-based application'. On the right, there's a 'Clone' dropdown with options 'Clone with SSH' (SSH address: git@4bb00da20027:root/spring) and 'Clone with HTTP' (HTTP address: http://4bb00da20027:root/spr). A 'Copy URL to clipboard' button is shown next to the HTTP address. Below this, a merge pull request is listed: 'Merge pull request #2 from kingsman142/master ... kontostathis authored 1 year ago'. The commit list shows several commits from 'kontostathis' and 'Unverified' status.

IMPORTANT

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

Create Maven Deployment in Jenkins

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

The screenshot shows the Jenkins dashboard. At the top is a dark header bar with the Jenkins logo and the word "Jenkins". Below it is a light-colored sidebar with the following items:

- New Item
- People
- Build History
- Project Relationship
- Check File Fingerprint
- Manage Jenkins
- My Views
- Lockable Resources
- Credentials
- New View

Below the sidebar is a section titled "Build Queue" which contains the message "No builds in the queue." To the right of this message is a small minus sign icon.

Further down is a section titled "Build Executor Status" which lists "1 Idle" and "2 Idle". To the right of this list is another small minus sign icon.

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

Enter an item name

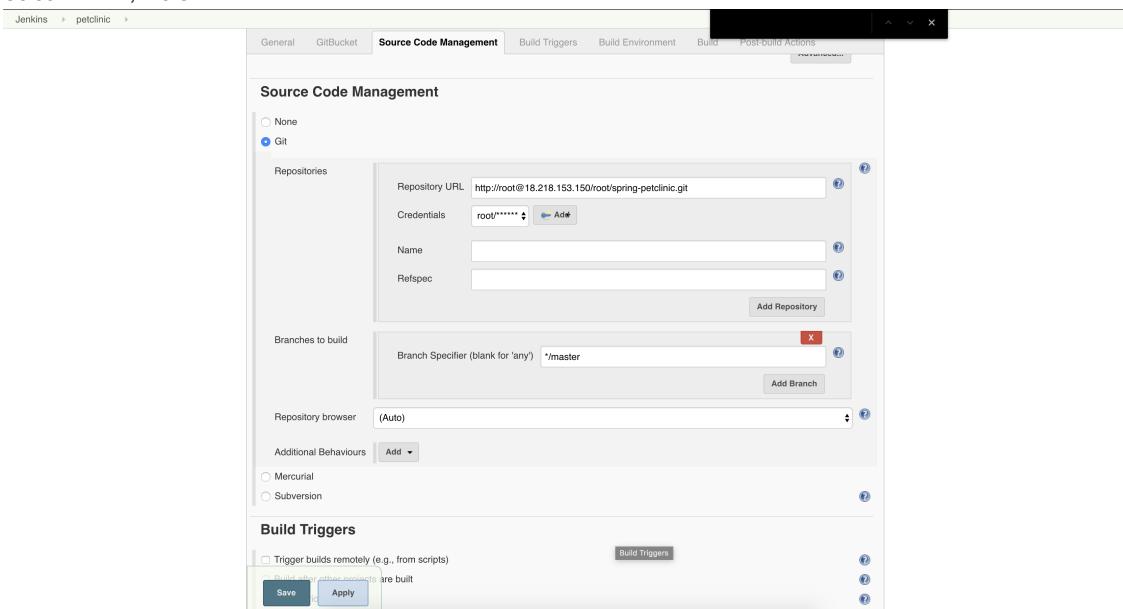
`petclinic`

» Required field

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

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

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



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

Maven project petclinic2

SonarQube

Recent Changes

SonarQube Quality Gate

Permalinks

- Last build (#2), 40 sec ago
- Last stable build (#2), 40 sec ago
- Last successful build (#2), 40 sec ago
- Last completed build (#2), 40 sec ago

[RSS for all](#) [RSS for failures](#)

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

The screenshot shows the SonarQube homepage with the following details:

- Continuous Code Quality**
- Projects Analyzed:** 0 Bugs, 0 Vulnerabilities, 0 Code Smells
- Multi-Language:** Java, C/C++, C#, COBOL, ABAP, HTML, RPG, JavaScript, TypeScript, Objective C, XML, VB.NET, PL/SQL, T-SQL, Flex, Python, Groovy, PHP, Swift, Visual Basic, PL/I
- Quality Model:**
 - Bugs:** track code that is demonstrably wrong or highly likely to yield unexpected behavior.
 - Vulnerabilities:** are raised on code that is potentially vulnerable to exploitation by hackers.
 - Code Smells:** will confuse maintainers or give them pause. They are measured primarily in terms of the time they will take to fix.
- Write Clean Code:** By fixing new issues as they appear in code, you create and maintain a clean code base. Even on legacy projects, focusing on keeping new code clean will eventually yield a code base you can be proud of. [Read More](#)
- Fix The Leak:** The leak test paradigm and the default Quality Gats are based on the leak period - the recent period against which you're tracking issues. For some previous_version makes the most sense, for others the last 30 days is a good option. [Read More](#)

- Enter a name for your token generation

The screenshot shows the 'Provide a token' step of the tutorial:

- 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.

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

The screenshot shows the 'Provide a token' step of the tutorial after generating the token:

- 1 Provide a token

microcosm_token: [X](#)

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. [Continue](#)

- Return to Jenkins

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

The screenshot shows the Jenkins configuration interface under the 'Global Properties' section. In the 'SonarQube servers' section, there is a 'SonarQube installations' table. A new row is being added, with the 'Name' field set to 'SonarQube' and the 'Server URL' field set to 'http://18.222.11.148:9000'. Other fields like 'Server authentication token' and 'Version of sonar-maven-plugin' are also visible.

Name	Server URL
SonarQube	http://18.222.11.148:9000

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

The screenshot shows the Jenkins configuration interface under the 'Global Tools' section. In the 'SonarScanner for MSBuild' section, a new 'SonarScanner for MSBuild installations' entry is being added. A dropdown menu shows various versions of the SonarQube Scanner, with 'SonarQube Scanner 3.3.0.1492' selected. Other options include 'SonarQube Scanner 3.2.0.1227', 'SonarQube Scanner 3.1.0.1141', 'SonarQube Scanner 3.0.3.778', 'SonarQube Scanner 3.0.2.768', 'SonarQube Scanner 3.0.1.733', 'SonarQube Scanner 3.0.0.702', 'SonarQube Scanner 2.9.0.670', 'SonarQube Scanner 2.8.0.637', 'SonarQube Scanner 2.7', 'SonarQube Scanner 2.6', 'SonarQube Scanner 2.5.1', 'SonarQube Scanner 2.5', 'SonarQube Scanner 2.4', 'SonarQube Scanner 2.3', 'SonarQube Scanner 2.2.2', 'SonarQube Scanner 2.2.1', 'SonarQube Scanner 2.2', 'SonarQube Scanner 2.1', 'SonarQube Scanner 2.0', 'SonarQube Scanner 1.4', 'SonarQube Scanner 1.3', 'SonarQube Scanner 1.2', 'SonarQube Scanner 1.1', and 'SonarQube Scanner 1.0'.

- Return to the PetClinic configuration screen
- Scroll down to the Post Steps Section

- Click “Add build step” and select “Execute SonarQube Scanner”

The screenshot shows the Jenkins petclinic project configuration under the 'Build' tab. A dropdown menu titled 'Add build step' is open, and the 'Execute SonarQube Scanner' option is highlighted with a blue selection bar.

- 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

The screenshot shows the Jenkins petclinic project configuration under the 'Build' tab. The 'Analysis properties' field contains the following configuration:

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

- 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](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

Status	Count
Passed	1
Warning	0
Failed	0

Reliability (Bug)

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

Security (Vulnerabilities)

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

Maintainability (Code Smells)

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

Coverage

Grade	Count
> 80%	0
< 80%	0
< 70%	0

petclinic Passed

Last analysis: April 17, 2019, 11:23 AM

0 Bugs 0 Vulnerabilities 0 Code Smells Coverage Duplications

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.

SonarCube™ technology is powered by SonarSource SA
Version 6.7.6 (build 38781) - LGPL v3 - Community Documentation - Get Support - Plugins - Web API - About

Add OwaspZap build step

- The ZAP web interface is available at: [http://\[IP from ECS Task\]:8080/?anonym=true&app=ZAP](http://[IP from ECS Task]:8080/?anonym=true&app=ZAP)
 - Zap integration with containerized Jenkins is not covered in this tutorial aside from the afore mentioned [plugin setup](#). It is left to the reader to research and do this if desired.



Please sign in

User name

Password

Sign in



Add Sonatype Nexus build step

If you would like to use Sonatype Nexus for build versioning and storage, see the instructions in Appendix A.

Slack Integration

- Refer back to the [Slack Setup Instructions](#) for instructions on adding Slack Notifications as a Post Build process in Jenkins. View the slack channel for messages from Jenkins during/following builds

Nagios Integration

Container Security

Manual Checking with Twistlock CloudDiscovery

Using Twistlock's CloudDiscovery container, instantiated as and ECS services (cloud-discovery) above, we can get a listing of all images and services available in the subnet shared by the container as well as list the open ports for any of our ECS containers/tasks or EC2 instances.

For this portion of the exercise, we will use the curl command and the terminal. If you do not have a terminal/console/command prompt/powershell/bash shell/git shell/ or similar with the curl command. Please study the following screen shots.

- First access the ECS console and navigate to the target page for the cloud-discovery service.

[Clusters](#) > [microcosm](#) > Task: 70d9199f-490a-4f7e-ae25-985145ead22e

Task : 70d9199f-490a-4f7e-ae25-985145ead22e

[Details](#) [Tags](#) [Logs](#)

Cluster	microcosm
Launch type	FARGATE
Platform version	1.3.0
Task definition	cloud-discovery:1
Group	service:cloud-discovery
Task role	ecsTaskExecutionRole
Last status	RUNNING
Desired status	RUNNING
Created at	2019-04-30 09:33:55 -0600
Started at	2019-04-30 09:34:36 -0600

Network

Network mode	awsvpc
ENI Id	eni-04ae8e813349e97e7
Subnet Id	subnet-0ea9bed2074fe558b
Private IP	10.0.0.71
Public IP	13.59.244.184
Mac address	0a:43:40:3d:07:1e

Containers

	Name	Container Id	Status
▶	cloud-discovery	8a084786-8aec-4c8d-a169-db843eb9a3aa	RUNNING

- Copy and paste your Accesskey-ID and secret key, as well as the ip address of the cloud-discovery task (aka container) and enter into the following command in the terminal (note we set previously the username and password for the container to admin:password):
 - curl -k -v -u admin:pass --raw --data '{"credentials": [{"id": "[AWS_ACCESS_KEY]", "secret": "[AWS_SECRET_ACCESS_KEY]"}]} https://[cloud-discovery IP]:9083/discover
 - for example curl -k -v -u admin:pass --raw --data '{"credentials": [{"id": "AKIAW0JKHRHJCDYDAKNE", "secret": "ZNHgDgrWBqnBR+K2AxGncsul7m2S1H0PJk4yqit5"}]}'

```

https://13.59.244.184:9083/discover
mac-topazpomme:microcosm-cloud smorley$ curl -k -v -u admin:pass --raw --data '{"credentials": [{"id": "CDYDAKNE", "secret": "ZNHgDgrWBqnBR+K2AxGncsul7m2SlH0Pjk4yqit5"}]}' https://13.59.244.184:9083/discover
*   Trying 13.59.244.184...
* TCP_NODELAY set
* Connected to 13.59.244.184 (13.59.244.184) port 9083 (#0)
* ALPN, offering h2
* ALPN, offering http/1.1
* Cipher selection: ALL:!EXPORT:!EXPORT40:!EXPORT56:!aNULL:!LOW:!RC4:@STRENGTH
* successfully set certificate verify locations:
*   CAfile: /etc/ssl/cert.pem
*   CApath: none
* TLSv1.2 (OUT), TLS handshake, Client hello (1):
* TLSv1.2 (IN), TLS handshake, Server hello (2):
* TLSv1.2 (IN), TLS handshake, Certificate (11):
* TLSv1.2 (IN), TLS handshake, Server key exchange (12):
* TLSv1.2 (IN), TLS handshake, Server finished (14):
* TLSv1.2 (OUT), TLS handshake, Client key exchange (16):
* TLSv1.2 (OUT), TLS change cipher, Client hello (1):
* TLSv1.2 (OUT), TLS handshake, Finished (20):
* TLSv1.2 (IN), TLS handshake, Finished (20):
* SSL connection using TLSv1.2 / ECDHE-RSA-CHACHA20-POLY1305
* ALPN, server accepted to use http/1.1
* Server certificate:
*   subject: O=Cloud discovery
*   start date: Apr 30 15:34:17 2019 GMT
*   expire date: Apr 29 15:34:17 2020 GMT
*   issuer: O=Cloud discovery
*   SSL certificate verify result: self signed certificate (18), continuing anyway.
* Server auth using Basic with user 'admin'
> POST /discover HTTP/1.1
> Host: 13.59.244.184:9083
> Authorization: Basic YWRtaW46cGFzcw==
> User-Agent: curl/7.54.0
> Accept: */*
> Content-Length: 100
> Content-Type: application/x-www-form-urlencoded
>
* upload completely sent off: 100 out of 100 bytes
Type      Region      ID
ECS       us-east-2    microcosm
ECS       us-east-2    default
Lambda    us-east-2    tstpetclinic
Lambda    us-east-2    MacroProcessor
ECR       us-east-2    sonarqube
ECR       us-east-2    hikkan/jenkins-docker
ECR       us-east-2    sonatype/nexus
ECR       us-east-2    gitlab/gitlab-ce
ECR       us-east-2    gillax/hubot-slack-jenkins
ECR       us-east-2    owasp/zap2docker-stable
* TLSv1.2 (IN), TLS alert, Client hello (1):
* Connection #0 to host 13.59.244.184 left intact

```

- If you prefer to get more details about the images in use, you can have the service return JSON with more information
 - for example `curl -k -v -u admin:pass --raw --data '{"credentials": [{"id": "AKIAW0JKHRHJCDYDAKNE", "secret": "ZNHgDgrWBqnBR+K2AxGncsul7m2SlH0Pjk4yqit5"}]}'`

```

https://13.59.244.184:9083/discover?format=json
mac-topazpomme:microcosm-cloud smorley$ curl -k -v -u admin:pass --raw --data '{"credentials": [{"id": "AKIAWOJXKHRHJCDYDAKNE", "secret": "ZNHgDgwBqnbRrX2AxGnc17mZSL1H0PJK4yqit5"}]} https://13.59.244.184:9083/discover?format=json
*   Trying 13.59.244.184...
* TCP_NODELAY set
* Connected to 13.59.244.184 (13.59.244.184) port 9083 (#0)
* ALPN, offering h2
* ALPN, offering http/1.1
* Cipher selection: ALL:!EXPORT:!EXPORT40:!EXPORT56:!aNULL:!LOW:!RC4:@STRENGTH
* successfully set certificate verify locations:
* CAfile: /etc/ssl/cert.pem
* CApath: none
* TLSv1.2 (OUT), TLS handshake, Client hello (1):
* TLSv1.2 (IN), TLS handshake, Server hello (2):
* TLSv1.2 (IN), TLS handshake, Certificate (11):
* TLSv1.2 (IN), TLS handshake, Server key exchange (12):
* TLSv1.2 (IN), TLS handshake, Server finished (14):
* TLSv1.2 (OUT), TLS handshake, Client key exchange (16):
* TLSv1.2 (OUT), TLS change cipher, Client hello (1):
* TLSv1.2 (OUT), TLS handshake, Finished (20):
* TLSv1.2 (IN), TLS change cipher, Client hello (1):
* TLSv1.2 (IN), TLS handshake, Finished (20):
* SSL connection using TLSv1.2 / ECDHE-RSA-CHACHA20-POLY1305
* ALPN, server accepted to use http/1.1
* Server certificate:
*   subject: O=Cloud discovery
*   start date: Apr 30 15:34:17 2019 GMT
*   expire date: Apr 29 15:34:17 2020 GMT
*   issuer: O=Cloud discovery
*   SSL certificate verify result: self signed certificate (18), continuing anyway.
* Server auth using Basic with user 'admin'
> POST /discover?format=json HTTP/1.1
> Host: 13.59.244.184:9083
> Authorization: Basic YWRtaW46cGFzcw==
> User-Agent: curl/7.54.0
> Accept: */*
> Content-Length: 100
> Content-Type: application/x-www-form-urlencoded
>

* upload completely sent off: 100 out of 100 bytes
{"region": "us-east-2", "type": "ECS", "assets": [{"id": "microcosm", "data": {"hosts": "null"}, {"id": "default", "data": {"hosts": "null"}}}]
{"region": "us-east-2", "type": "Lambda", "assets": [{"id": "tstptclinic", "data": {"codeSha256": "100dRcg1AfDM87MBC1tRPE2v+cw2mEALKjzfMSOMcs=", "codeSize": 333, "description": "test", "functionARN": "arn:aws:lambda:us-east-2:443007076818:function:tstptclinic:$LATEST", "functionName": "tstp etclinic", "handler": "lambda function.lambda", "lastModified": "2019-04-25T21:43:26.392+0000", "masterArn": null, "memorySize": 128, "revis ionId": "4c39e75f-4d02-46d7-bbcc-7ffa0a02d756", "role": "arn:aws:iam::443007076818:role/lambda-s3-role", "runtime": "python3.6", "timeout": 3, "ver sion": "$LATEST"}, {"id": "MacroProcessor", "data": {"codeSha256": "1v1r9MFwCvzqkVvJYCRNXVEurwJTSOSbmDxDORQo=", "codeSize": 939, "description": "IAM User Macro processor function", "functionARN": "arn:aws:lambda:us-east-2:443007076818:function:MacroProcessor:$LATEST", "functionName": "macroProcessor", "handler": "index.lambda_handler", "lastModified": "2019-04-15T20:09:12.610+0000", "masterArn": null, "memorySize": 128, "revisionId": "5200c77d-c4db-4679-9d78-9c9464cd5cd", "role": "arn:aws:iam::443007076818:role/MacroProcessor-lambda-role-us-east-2", "runtime": "python3.6", "timeout": 100, "version": "$LATEST"}]}
{"region": "us-east-2", "type": "ECR", "assets": [{"id": "sonarqube", "data": {"arn": null, "createdAt": "2019-04-10T20:50:52Z", "registryId": "443007076818", "repositoryArn": "arn:ecr:us-east-2:443007076818:repository/sonarqube", "repositoryUri": "443007076818.dkr.ecr.us-east-2.amazonaws.com/sonarqube", "version": "null"}, {"id": "hikkan/jenkins-docker", "data": {"arn": null, "createdAt": "2019-04-10T21:29:41Z", "registryId": "443007076818", "repositoryArn": "arn:aws:ecr:us-east-2:443007076818:repository/hikkan/jenkins-docker", "repositoryUri": "443007076818.dkr.ecr.us-east-2.amazonaws.com/hikkan/jenkins-docker", "version": "null"}, {"id": "sonatype/nexus", "data": {"arn": null, "createdAt": "2019-04-10T21:27:06Z", "registryId": "443007076818", "repositoryArn": "arn:aws:ecr:us-east-2:443007076818:repository/sonatype/nexus", "repositoryUri": "443007076818.dkr.ecr.us-east-2.amazonaws.com/sonatype/nexus", "version": "null"}, {"id": "gitlab/gitlab-ce", "data": {"arn": null, "createdAt": "2019-04-10T21:20:15Z", "registryId": "443007076818", "repositoryArn": "arn:aws:ecr:us-east-2:443007076818:repository/gitlab/gitlab-ce", "repositoryUri": "443007076818.dkr.ecr.us-east-2.amazonaws.com/gitlab/gitlab-ce", "version": "null"}, {"id": "gillax/hubot-slack-jenkins", "data": {"arn": null, "createdAt": "2019-04-10T21:32:20Z", "registryId": "443007076818", "repositoryArn": "arn:aws:ecr:us-east-2:443007076818:repository/gillax/hubot-slack-jenkins", "repositoryUri": "443007076818.dkr.ecr.us-east-2.amazonaws.com/gillax/hubot-slack-jenkins", "version": "null"}, {"id": "owasp/zap2docker-stable", "data": {"arn": null, "createdAt": "2019-04-10T21:32:20Z", "registryId": "443007076818", "repositoryArn": "arn:aws:ecr:us-east-2:443007076818:repository/owasp/zap2docker-stable", "repositoryUri": "443007076818.dkr.ecr.us-east-2.amazonaws.com/owasp/zap2docker-stable", "version": "null"}]}
* TLSv1.2 (IN), TLS alert, Client hello (1):
* Connection #0 to host 13.59.244.184 left intact

```

- From the Cloud-Discovery server, you can inspect open ports for insecurities on individual hosts that are reachable within the subnet, or the internet at large, for open ports that may be insecure. You can get the IP's of your instances through their ECS task pages or EC2 Console instance listings. For example, the following results are returned for either the subnet or public IPs of this Sonarqube container instance when used in this command: curl -k -v -u admin:pass --raw --data '{"subnet": "[ip address of target machine]", "debug": true}' https://[ip address of Cloud-Discovery task]:9083/nmap
 - for example curl -k -v -u admin:pass --raw --data '{"subnet": "10.0.0.93", "debug": true}' https://13.59.244.184:9083/nmap

Task : 70d9199f-490a-4f7e-ae25-985145ead22e

Details Tags Logs

Cluster microcosm
Launch type FARGATE
Platform version 1.3.0
Task definition [cloud-discovery:1](#)
Group service:cloud-discovery
Task role [ecsTaskExecutionRole](#)
Last status RUNNING
Desired status RUNNING
Created at 2019-04-30 09:33:55 -0600
Started at 2019-04-30 09:34:36 -0600

Network

Network mode awsvpc
ENI Id [eni-04ae8e813349e97e7](#)
Subnet Id subnet-0ea9bed2074fe558b
Private IP 10.0.0.71
Public IP 13.59.244.184
Mac address 0a:43:40:3d:07:1e

Containers

	Name	Container Id	Status
▶	cloud-discovery	8a084786-8aec-4c8d-a169-db843eb9a3aa	RUNNING

```
* Connection #0 to host 13.59.244.184 left intact
mac-topazpomme:microcosm-cloud smorrelle$ curl -k -v -u admin:pass --raw --data '{"ubnet":"10.0.0.93", "debug": true}' https://13.59.244.184:9083/nmap
*   Trying 13.59.244.184...
* TCP_NODELAY set
* Connected to 13.59.244.184 (13.59.244.184) port 9083 (#0)
* ALPN, offering h2
* ALPN, offering http/1.1
* Cipher selection: ALL:!EXPORT:!EXPORT40:!EXPORT56:!aNULL:!LOW:!RC4:@STRENGTH
* successfully set certificate verify locations:
*   CAfile: /etc/ssl/cert.pem
*   CApth: none
* TLSv1.2 (OUT), TLS handshake, Client hello (1):
* TLSv1.2 (IN), TLS handshake, Server hello (2):
* TLSv1.2 (IN), TLS handshake, Certificate (11):
* TLSv1.2 (IN), TLS handshake, Server key exchange (12):
* TLSv1.2 (IN), TLS handshake, Server finished (14):
* TLSv1.2 (OUT), TLS handshake, Client key exchange (16):
* TLSv1.2 (OUT), TLS change cipher, Client hello (1):
* TLSv1.2 (OUT), TLS handshake, Finished (20):
* TLSv1.2 (IN), TLS change cipher, Client hello (1):
* TLSv1.2 (IN), TLS handshake, Finished (20):
* SSL connection using TLSv1.2 / ECDHE-RSA-CHACHA20-POLY1305
* ALPN, server accepted to use http/1.1
* Server certificate:
*   subject: O=Cloud discovery
*   start date: Apr 30 15:34:17 2019 GMT
*   expire date: Apr 29 15:34:17 2020 GMT
*   issuer: O=Cloud discovery
* SSL certificate verify result: self signed certificate (18), continuing anyway.
* Server auth using Basic with user 'admin'
> POST /nmap HTTP/1.1
> Host: 13.59.244.184:9083
> Authorization: Basic YWRtaW46cGFzcw==
> User-Agent: curl/7.54.0
> Accept: */*
> Content-Length: 37
> Content-Type: application/x-www-form-urlencoded
>
* upload completely sent off: 37 out of 37 bytes
<
Host      Port     App          Insecure    Reason
10.0.0.93 80        http         false
10.0.0.93 443       https        false
10.0.0.93 8080      http-proxy   false
10.0.0.93 8081      blackice-icecap false
10.0.0.93 8090      opsmessaging false
10.0.0.93 9000      cslistener   false
10.0.0.93 9083      emc-pp-mgmtsvc false
* TLSv1.2 (IN), TLS alert, Client hello (1):
* Connection #0 to host 13.59.244.184 left intact
```

- When you are finished experimenting with various IPs from your services, stop the task and delete the service when done

Ref: CloudDiscovery (<https://www.twistlock.com/2018/11/13/open-source-cloud-discovery-tool/>)

Automated checking

<https://hub.docker.com/r/twistlocktest/hello-world> (<https://hub.docker.com/r/twistlocktest/hello-world>)

<https://hub.docker.com/r/matthewwaba/twistlock-fargate> (<https://hub.docker.com/r/matthewwaba/twistlock-fargate>)

AWS EC2 Based Code Deploy

Ref. Youtube walkthrough example (<https://www.youtube.com/watch?v=83d5YuG-KiQ>)

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

What are IAM roles?

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

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

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

Additional resources:

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

[Create role](#) [Delete role](#)

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

Create role

Select type of trusted entity



AWS service

EC2, Lambda and others



Another AWS

Belonging to you

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

Choose the service that will use this role

EC2

Allows EC2 instances to call AWS services on your behalf.

Lambda

Allows Lambda functions to call AWS services on your behalf

API Gateway

CodeDeploy

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

Select your use case

CodeDeploy

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

CodeDeploy - ECS

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

CodeDeploy for Lambda

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

- Press Next

Create role

1 2 3 4

Attached permissions policies

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

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

- Add Tags if desired and press Next

Create role

1 2 3 4

Add tags (optional)

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

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

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

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

Permissions **Trust relationships** **Tags (1)** **Access Advisor** **Revoke sessions**

▼ Permissions policies (1 policy applied)

Attach policies

Policy name
▶ AWSCodeDeployRole

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

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.

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

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

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

Step 2: Choose an Instance Type

Amazon EC2 provides a wide selection of instance types optimized to fit different use cases. Instances are virtual servers that can run applications. They have v [Learn more](#) about instance types and how they can meet your computing needs.

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

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

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

- In Step 3, do the following
 - Number of Instance: 2
 - Network: Select your Student VPC
 - Subnet: Select either subnet
 - Auto-Assign public IP: Select Enable (or leave the default if it is Enable)
 - **Under Advanced Detail**, copy and paste the contents of the `addCodeDeploytoEC2Image.sh` file in the optional text box. Make sure the As Text radio button is selected. The purpose of this script is to add the libraries and services necessary to the Ubuntu based VM to be able to deploy to the machine in an automated fashion. Note that the script uses the AWS EAST-2 region (Ohio), change this portion of the S3 url for the region you are using. Additionally, this script installs the services necessary to forward logs to CloudWatch to be used for monitoring and debugging deployments, and program runs. [ref. \(https://aws.amazon.com/blogs/devops/view-aws-codedeploy-logs-in-amazon-cloudwatch-console/\)](https://aws.amazon.com/blogs/devops/view-aws-codedeploy-logs-in-amazon-cloudwatch-console/).

- Press Next

Step 3: Configure Instance Details

Subnet:

Auto-assign Public IP:

Placement group: Add instance to placement group

Capacity Reservation:

IAM role:

Shutdown behavior:

Enable termination protection: Protect against accidental termination

Monitoring: Enable CloudWatch detailed monitoring
Additional charges apply.

Tenancy:
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	<input type="button" value="New network interface"/>	<input type="text" value="subnet-04d355bb02dc06049"/>	<input type="button" value="Auto-assign"/>	<input type="button" value="Add IP"/>	<input type="button" value="Add IP"/>

Add Device

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

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

Step 5: Add Tags

A tag consists of a case-sensitive key-value pair. For example, you could define a tag with key = Name and value = Webserver.

A copy of a tag can be applied to volumes, instances or both.

Tags will be applied to all instances and volumes. [Learn more](#) about tagging your Amazon EC2 resources.

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

Add another tag (Up to 50 tags maximum)

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

- Select Review and Launch.

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

Step 6: Configure Security Group

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

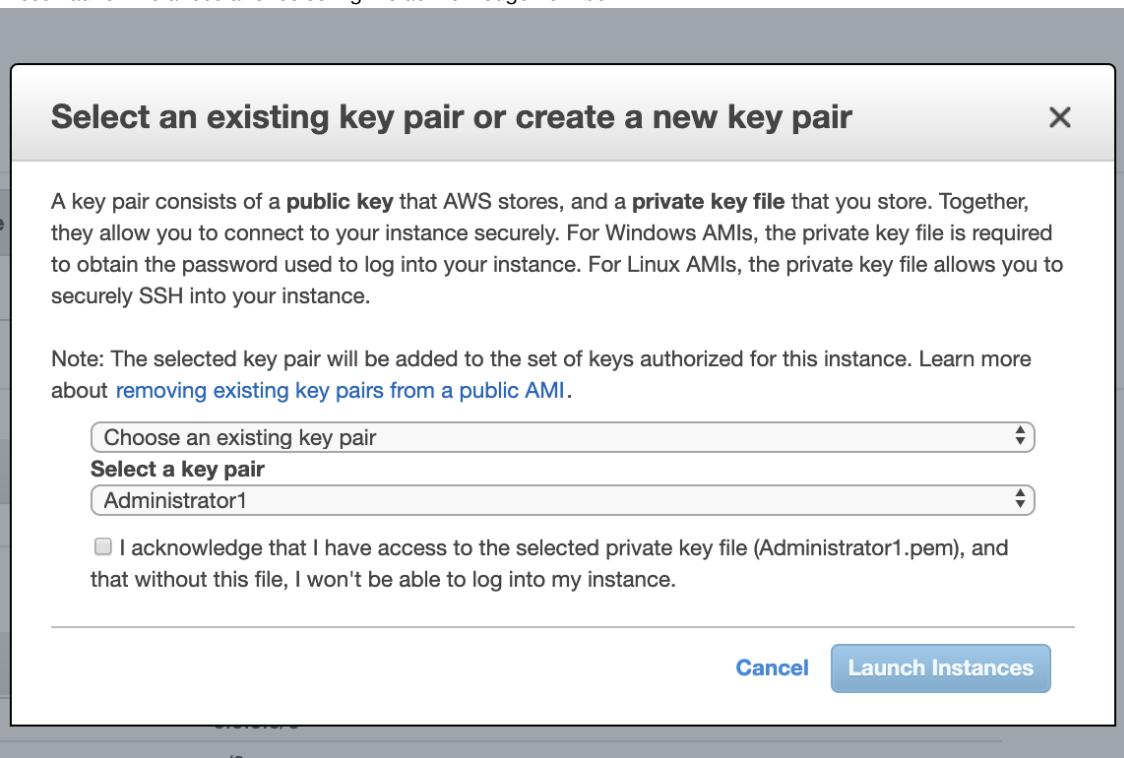
Assign a security group: Create a **new** security group
 Select an **existing** security group

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

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

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

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



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

Name	Instance ID	Instance Type	Availability Zone	Instance State	Status Checks	Alarm Status	Public DNS (IPv4)	IPv4 Public IP	IPv6 IPs	Key Name	Monitoring	Launch Time	Security Groups
Code_Deploy_...	i-03a155360da837b5	t2.large	us-east-2a	running	Initializing	None	ec2-18-218-184-15.eca...	18.218.184.15	-	Administrator1	disabled	April 26, 2019 at 9:14:08 AM...	default
Code_Deploy_...	i-0e2ab0cb45d8bd804	t2.large	us-east-2a	running	Initializing	None	ec2-18-222-113-20.eca...	18.222.113.20	-	Administrator1	disabled	April 26, 2019 at 9:14:08 AM...	default

Load Balancer

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

- Press Next

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

Step 1: Configure Load Balancer

Basic Configuration

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

Name	<input type="text" value="ec2petclinic"/>	<small>⚠ Load balancer names must only contain alphanumeric characters or hyphens, and not start with a hyphen.</small>
Scheme	<input checked="" type="radio"/> internet-facing <input type="radio"/> internal	
IP address type	<input type="text" value="ipv4"/>	

Listeners

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

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

Availability Zones

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

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

Tags

- Press Next for step 2

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

Step 2: Configure Security Settings



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

If your traffic to the load balancer needs to be secure, use the HTTPS protocol for your front-end connection. You can go back to the first step to change 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
<input type="checkbox"/> sg-0e1a263967d2471f2	default	default VPC security group

- Step 4, Create a new Target Group

- Name: create a memorable name (e.g. StudentCodeDeployTG8080)
- Port: 80
- Path: /index.html

- Press Next

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

Step 4: Configure Routing

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

Target group

Target group	i	New target group	▼
Name	i	TGcodedeployec2petclinic	⚠ TargetGroup name cannot contain characters that are not letters, numbers, or underscores.
Target type	<input checked="" type="radio"/> Instance <input type="radio"/> IP <input type="radio"/> Lambda function		
Protocol	i	HTTP	▼
Port	i	8080	

Health checks

Protocol	i	HTTP	▼
Path	i	/petclinic	
▼ Advanced health check settings			
Port	i	<input checked="" type="radio"/> traffic port <input type="radio"/> override	
Healthy threshold	i	5	
Unhealthy threshold	i	2	
Timeout	i	5	seconds
Interval	i	30	seconds
Success codes	i	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
<input type="checkbox"/>	i-03a155360dabf37b5	Code_Deploy_Instances	8080	running	default	us-east-2a
<input type="checkbox"/>	i-0e2a50cb45d6bd604	Code_Deploy_Instances	8080	running	default	us-east-2a

Instances

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

[Add to registered](#) on port 8080

Remove	Instance	Name	State	Security groups	Zone	Subnet ID	Subnet CIDR
<input checked="" type="checkbox"/>	i-03a155360dabf37b5	Code_Deploy_Instances	running	default	us-east-2a	subnet-04d355bb02dc06049	10.0.0.0/24
<input checked="" type="checkbox"/>	i-0e2a50cb45d6bd604	Code_Deploy_Instances	running	default	us-east-2a	subnet-04d355bb02dc06049	10.0.0.0/24

- Step 6, Review and Press Create

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

Step 6: Review

Please review the load balancer details before continuing

▼ Load balancer

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

▼ Security groups

Security groups sg-0e1a263967d2471f2

▼ Routing

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

▼ Targets

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

Create S3 Bucket

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

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

S3 buckets

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

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

Create bucket

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

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

Public access settings for this bucket

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

Manage public access control lists (ACLs) for this bucket

Block new public ACLs and uploading public objects (Recommended) [?](#)

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 has a dark header 'AWS CodeDeploy' and sub-headings 'Automate code deployments to maintain application uptime' and 'AWS CodeDeploy is a fully managed deployment service that automates software deployments to compute services such as Amazon EC2, AWS Lambda, and your on-premises servers. AWS CodeDeploy makes it easier for you to rapidly release new features, helps you avoid downtime during application deployment, and handles the complexity of updating your applications.' To the right, there's a callout box with 'Create AWS CodeDeploy deployment', a sub-section 'Get started with AWS CodeDeploy by creating your first deployment application.', and a large orange 'Create application' button. At the bottom right, there's a 'Pricing (US)' link.

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

The screenshot shows the 'Create application' configuration page. At the top, there's a breadcrumb navigation: 'Developer Tools > CodeDeploy > Applications > Create application'. The main section is titled 'Application configuration'. It has two main fields: 'Application name' (with placeholder 'Enter an application name' and input field containing 'petclinic') and 'Compute platform' (with placeholder 'Choose a compute platform' and dropdown menu showing 'EC2/On-premises'). At the bottom right, there are 'Cancel' and 'Create application' buttons.

- 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

Name X

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](#)

The screenshot shows the AWS CodeDeploy console with the 'petclinicDepGrp' deployment group selected. The 'Deployment group details' section includes:

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

The 'Environment configuration: Amazon EC2 instances' section shows a single entry:

Key	Value
Name	Code_Deploy_Instances

Create Secret Key

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

- Navigate to the IAM Concole

Welcome to Identity and Access Management

IAM users sign-in link:

<https://cert-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="checkbox"/> Find users by username or access key	
<input type="checkbox"/> User name ▾	Groups
<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

Search IAM

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](#)

Sign-in credentials

Summary

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

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

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

Signing certificates None 

Access keys

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

[Create access key](#)

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

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

The screenshot shows the 'Create access key' dialog box from the AWS IAM 'Summary' page. The dialog box has a green header bar with the word 'Success' and a checkmark icon. Below it, a message states: 'This is the **only** time that the secret access keys can be viewed or downloaded. You cannot recover them later. However, you can create new access keys at any time.' At the bottom left of the dialog is a 'Download .csv file' button with a CSV icon. To the right of the dialog is a 'Close' button.

Access key ID	Secret access key
AKIAWOJKHRHJASUZR35N	7ar8nUSG1C2gstsL9x3H6qBf46ta202Man6/wRXF Hide

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
AKIAWOJKHRHJASUZR35N	2019-04-27 07:27 MDT	N/A

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

The screenshot shows a 'Save File' dialog box. The file name field is highlighted with a blue border and contains the text 'accessKeys.csv'. Below the file name field is a dropdown menu set to 'comma-separated values'. At the bottom right of the dialog are 'Cancel' and 'Save' buttons.

- 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
AKIAWOJKHRHJASUZR35N	7ar8nUSG1C2gstsL9x3H6qBf46ta202Man6/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	**/*.yml, **/*.html, **/*.yaml, **/scripts/*.*; **/target/*.war
Exclude Files	./src, ./target
Proxy Host	
Proxy Port	0
Version File	
Appspec.yml per Deployment Group	<input type="checkbox"/>
<input type="radio"/> Register Revision <input type="radio"/> Deploy Revision <input checked="" type="radio"/> Use Access/Secret keys	
<small>If these keys are left blank, the plugin will attempt to use credentials from the default provider chain. That is: Environment Variables, Java System properties, credentials profile file, and finally, EC2 Instance profile.</small>	
Save	Apply
AWS Secret Key	

Forward Build Data to AWS Cloud Watch

- In AWS CloudWatch Console, select logs from the left side bar, then under Actions, select Create Log Group

Never Expire

Create log group

Log Group Name: /jenkins/jobs

Create log group

- Enter /jenkins/jobs then press Create Log Groups

CloudWatch > Log Groups > Streams for /jenkins/jobs

The screenshot shows the AWS CloudWatch Log Groups interface. At the top, there are three buttons: "Search Log Group" (blue), "Create Log Stream" (grey), and "Delete Log Stream" (grey). Below these is a search bar labeled "Filter: Log Stream Name Prefix" with an "x" button. Underneath the search bar is a section titled "Log Streams" with a checkbox. Two log streams are listed: "petclinic/89" and "petclinic/88".

Log Stream
<input type="checkbox"/> petclinic/89
<input type="checkbox"/> petclinic/88

- In Manage Jenkins > Configure System > AWS Logs Publisher, you have to configure the AWS Access Key Id and AWS Secret Key of an account with logs:CreateLogStream and logs:PutLogEvents rights.

AWS CloudWatch Logs Publisher

- Aggregate downstream test results
- Archive the artifacts
- Build other projects
- Deploy artifacts to Maven repository
- JIRA: Update relevant issues
- Publish HTML reports
- Publish Selenium Html Report
- Publish XML Summary Reports
- Publish artifacts to S3 Bucket
- Record fingerprints of files to track usage
- Git Publisher
- Deploy an application to AWS CodeDeploy
- SonarQube analysis with Maven
- Accept GitLab merge request on success
- Add note with build status on GitLab merge requests
- Add vote for build status on GitLab merge requests
- Build other projects (manual step)
- Editable Email Notification
- JIRA: Create issue
- JIRA: Create new version
- JIRA: Mark a version as Released
- JIRA: Move issues matching JQL to the specified version
- Publish build status to GitLab
- Send build artifacts over SSH
- Set GitHub commit status (universal)

Add post-build action ▾

- In your project/job configuration page for Petclinic, go to the section Post-build Actions, click the Add post-build action button and select the item AWS CloudWatch Logs Publisher.

CloudWatch

Dashboards

Alarms

ALARM 0

INSUFFICIENT 0

OK 0

Billing

Events

Rules

Event Buses

Logs

 Insights

 Metrics

Favorites

CloudWatch > Log Groups

Create Metric Filter

Actions ▾

Create log group

Delete log group

Export

Export data to Amazon S3

View all exports to Amazon S3

Subscriptions

Stream to AWS Lambda

Stream to Amazon Elasticsearch Service

Remove Subscription Filter

Log Groups

/aws/lambda/Macros

/aws/lambda/tstpet

/ecs/cloud-discover

/ecs/gitlab

/ecs/hubot-slack

/ecs/jenkins

/ecs/nagios

/ecs/nexus

/ecs/petclinic

- After building in the next step, return to Cloudwatch and view the logs created

AWS CloudWatch Logs Publisher

AWS Access Key ID	AKIAWOJKHRHJCDYDAKNE
AWS Secret Key	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
AWS Region	us-east-2
Log group name	/jenkins/jobs

ref. (<https://wiki.jenkins.io/display/JENKINS/AWS+CloudWatch+Logs+Publisher+Plugin>)

Build and Deploy

- From your Project Page in Jenkins select Build

Jenkins

petclinic



Back to Dashboard



Status



Changes



Workspace



Build Now



Delete Maven project



Configure



Modules



Favorite



SonarQube



Open Blue Ocean



Rename



Embeddable Build Status



Build History

trend

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

Amazon S3 > petclinicdeploy > target

Overview

Type a prefix and press Enter to search. Press ESC

Upload **Create folder** **Download**

Name ▾

#43-3400450200434229806.zip

petclinic.war

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

Developer Tools > CodeDeploy > Deployments > d-9GMQAH25Y

d-9GMQAH25Y

Deployment status

Installing application on your instances 0 of 2 instances updated In progress

Deployment details

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/#54-8254b45117549694950.zip e7ag+bef67edee49a1f6645a1beef4ad5fb	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-03a155360dfb37b5	-	In progress	BeforeBlockTraffic	View events	Apr 26, 2019 11:11 AM	-
i-0e2a50cb45d5bd604	-	In progress	BeforeBlockTraffic	View events	Apr 26, 2019 11:11 AM	-

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

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

Application: petclinic

Deployment ID: d-9GMQAH2SY

Status: In progress

Revision details:

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 ID: d-FRO04J65Y

Deployment group: petclinicDepGrp

Revision details:

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

d-61VZJA35Y

Deployment status

Installing application on your instances 2 of 2 instances updated Succeeded

Deployment details

Application petclinic	Deployment ID d-61VZJA35Y	Status Succeeded
Deployment configuration CodeDeployDefault.AllAtOnce	Deployment group petclinicDepGrp	Initiated by user

Revision details

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

Deployment lifecycle events

Instance ID	Duration	Status	Most recent event	Events	Start time	End time
i-08dbe30c18a614261	59 seconds	Succeeded	AfterAllowTraffic	View events	Apr 26, 2019 3:51 PM	Apr 26, 2019 3:52 PM
i-0cfcb0d38fdf28923	1 minute 0 seconds	Succeeded	AfterAllowTraffic	View events	Apr 26, 2019 3:51 PM	Apr 26, 2019 3:52 PM

The screenshot shows the Pet Clinic application running on the Spring Framework. The header reads "Pet Clinic A Spring Framework Demonstration". Below the header are navigation links: Home, Find owners, Who-am-I, Error, and Help. The main content area features a large image of a puppy and a kitten. At the bottom right is the Spring logo with the text "by Pivotal".

Welcome



- Return to Cloudwatch and view the logs created by the Jenkins build, now forwarded to CloudWatch

AWS CloudWatch Logs Publisher

AWS Access Key ID	AKIAWOJKHRHJCDYDAKNE
AWS Secret Key	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
AWS Region	us-east-2
Log group name	/jenkins/jobs

Note, for the deployment process, if detailed logging is not enabled, it may be necessary to SSH to system or use the AWS CLI to gather additional information. The following screen shots demonstrate doing these actions:

1. Use CLI to retrieve CodeDeploy build status and more detailed failure information

```
...documents/git/microcosm-cloud -- bash ...istrator1.pem ec2-user@18.218.184.15 ...ments/git/microcosm-cloud -- bash + 
tance-id i-03a155360dafb37b5
^C
numa-vpn-10-64-201-76:microcosm-cloud smorley$ aws deploy get-deployment-instance --deployment-id d-USLRD135Y --inst
tance-id i-03a155360dafb37b5
{
    "instanceSummary": {
        "deploymentId": "d-USLRD135Y",
        "instanceId": "arn:aws:ec2:us-east-2:443007076818:instance/i-03a155360dafb37b5",
        "status": "Failed",
        "lastUpdatedAt": 1556307505.62,
        "lifecycleEvents": [
            {
                "lifecycleEventName": "BeforeBlockTraffic",
                "diagnostics": {
                    "errorCode": "Success",
                    "scriptName": "",
                    "message": "Succeeded",
                    "logTail": ""
                },
                "startTime": 1556307193.062,
                "endTime": 1556307193.139,
                "status": "Succeeded"
            },
            {
                "lifecycleEventName": "BlockTraffic",
                "diagnostics": {
                    "errorCode": "Success",
                    "scriptName": "",
                    "message": "Succeeded",
                    "logTail": ""
                },
                "startTime": 1556307194.053,
                "endTime": 1556307500.813,
                "status": "Succeeded"
            },
            {
                "lifecycleEventName": "AfterBlockTraffic",
                "diagnostics": {
                    "errorCode": "Success",
                    "scriptName": "",
                    "message": "Succeeded",
                    "logTail": ""
                },
                "startTime": 1556307501.436,
                "endTime": 1556307501.491,
                "status": "Succeeded"
            },
            {
                "lifecycleEventName": "ApplicationStop",
                "diagnostics": {
                    "errorCode": "Success",
                    "scriptName": "",
                    "message": "Succeeded",
                    "logTail": ""
                },
                "startTime": 1556307502.489,
                "endTime": 1556307502.55,
                "status": "Succeeded"
            }
        ]
    }
}
```

2. Bitnami Tomcat actions:

- the default username for ssh-ing to the Bitnami Tomcat servier is bitnami. Once logged in, sudo does not require a password.

```
mac-topazpommie:microcosm-cloud smorley$ ssh -i ~/.ssh/Administrator1.pem bitnami@52.15.147.245
The authenticity of host '52.15.147.245 (52.15.147.245)' can't be established.
ECDSA key fingerprint is SHA256:zsgB0rsDwUEUVSB1/s0B8Q9dqmd3IQRrT1+S19/7pY.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added '52.15.147.245' (ECDSA) to the list of known hosts.
Welcome to Ubuntu 16.04.6 LTS (GNU/Linux 4.4.0-1079-aws x86_64)

[...]
*** Welcome to the Bitnami Tomcat 8.5.39-3 ***
*** Documentation: https://docs.bitnami.com/aws/infrastructure/tomcat/ ***
*** https://docs.bitnami.com/aws/ ***
*** Bitnami Forums: https://community.bitnami.com/ ***
To run a command as administrator (user "root"), use "sudo <command>".
See "man sudo_root" for details.

bitnami@ip-10-0-0-139:~$ sudo -
sudo: option requires an argument -- 'u'
usage: sudo -h | -K | -k | -V
usage: sudo -v [-AknS] [-g group] [-h host] [-p prompt] [-u user]
usage: sudo -l [-AknS] [-g group] [-h host] [-p prompt] [-U user] [-u user] [command]
usage: sudo -AbEHknPSl [-r role] [-t type] [-C num] [-g group] [-h host] [-p prompt] [-u user] [VAR=value]
      [-i|-s] [<command>]
usage: sudo -e [-AknS] [-r role] [-t type] [-C num] [-g group] [-h host] [-p prompt] [-u user] file ...
bitnami@ip-10-0-0-139:~$ sudo ls /root
snap
bitnami@ip-10-0-0-139:~$
```

- Retrieving the Management console password and starting/restarting/stopping Bitnami manager services like Tomcat:

```
[bitnami@ip-10-0-0-139:/opt/bitnami/apache-tomcat/webapps$ sudo /opt/bitnami/ctlscript.sh restart tomcat
Using CATALINA_BASE:  /opt/bitnami/apache-tomcat
Using CATALINA_HOME:  /opt/bitnami/apache-tomcat
Using CATALINA_TMPDIR: /opt/bitnami/apache-tomcat/temp
Using JRE_HOME:       /opt/bitnami/java
Using CLASSPATH:      /opt/bitnami/apache-tomcat/bin/bootstrap.jar:/opt/bitnami/apache-tomcat/bin/tomcat-juli.jar
Using CATALINA_PID:   /opt/bitnami/apache-tomcat/temp/catalina.pid
NOTE: Picked up JDK_JAVA_OPTIONS: --add-opens=java.base/java.lang=ALL-UNNAMED --add-opens=java.base/java.io=ALL-U
NNAMED --add-opens=java.rmi/sun.rmi.transport=ALL-UNNAMED
Tomcat stopped.
/opt/bitnami/apache-tomcat/scripts/ctl.sh : tomcat stopped
Using CATALINA_BASE:  /opt/bitnami/apache-tomcat
Using CATALINA_HOME:  /opt/bitnami/apache-tomcat
Using CATALINA_TMPDIR: /opt/bitnami/apache-tomcat/temp
Using JRE_HOME:       /opt/bitnami/java
Using CLASSPATH:      /opt/bitnami/apache-tomcat/bin/bootstrap.jar:/opt/bitnami/apache-tomcat/bin/tomcat-juli.jar
Using CATALINA_PID:   /opt/bitnami/apache-tomcat/temp/catalina.pid
Tomcat started.
/opt/bitnami/apache-tomcat/scripts/ctl.sh : tomcat started
[bitnami@ip-10-0-0-139:/opt/bitnami/apache-tomcat/webapps$ cd ~
[bitnami@ip-10-0-0-139:~$ ll
total 40
drwxr-xr-x 4 bitnami bitnami 4096 Apr 26 21:11 .
drwxr-xr-x 4 root    root    4096 Apr  9 09:56 ..
lrwxrwxrwx 1 bitnami bitnami 17 Apr  9 09:58 apps -> /opt/bitnami/apps/
-rw----- 1 bitnami bitnami 113 Apr 26 21:11 .bash_history
-rw-r--r-- 1 bitnami bitnami 220 Aug 31 2015 .bash_logout
-rw-r--r-- 1 bitnami bitnami 3936 Apr  9 09:58 .bashrc
-r----- 1 bitnami bitnami 424 Apr 26 20:59 bitnami_credentials
drwx----- 2 bitnami bitnami 4096 Apr  9 09:47 .cache/
lrwxrwxrwx 1 bitnami bitnami 27 Apr  9 09:58 htdocs -> /opt/bitnami/apache2/htdocs/
-rw-r--r-- 1 bitnami bitnami 655 May 16 2017 .profile
-rw----- 1 root    root    1024 Apr  9 09:56 .rnd
drwx----- 2 bitnami bitnami 4096 Apr 26 20:58 .ssh/
lrwxrwxrwx 1 bitnami bitnami 12 Apr  9 09:58 stack -> /opt/bitnami/
-rw-r--r-- 1 bitnami bitnami 0 Apr 26 21:07 .sudo_as_admin_successful
[bitnami@ip-10-0-0-139:~$ cat bitnami_credentials
Welcome to the Bitnami Tomcat Stack

*****
The default username and password is 'manager' and 'uBXwfWeKwDv'.
*****
You can also use this password to access the databases and any other component the stack includes.

Please refer to https://docs.bitnami.com/ for more details.

[bitnami@ip-10-0-0-139:~$ cd ~
/opt/bitnami/apache-tomcat/webapps
bitnami@ip-10-0-0-139:/opt/bitnami/apache-tomcat/webapps$ ]
```

Note, to access the management console from a public IP, a manager.xml file must be created. This is done in one of the shell scripts called by appspec.yml when a redeployment runs.

Each deployed webapp has a `context.xml` file that lives in

115 `$CATALINA_BASE/conf/[enginemame]/[hostname]`
 (conf/Catalina/localhost by default)

and has the same name as the webapp (`manager.xml` in this case). If no file is present, default values are used.

So, you need to create a file `conf/Catalina/localhost/manager.xml` and specify the rule you want to allow remote access. For example, the following content of `manager.xml` will allow access from all machines:

```
<Context privileged="true" antiResourceLocking="false"
          docBase="${catalina.home}/webapps/manager">
    <Valve className="org.apache.catalina.valves.RemoteAddrValve" allow="^.*$" />
</Context>
```

Note that the `allow` attribute of the `Valve` element is a regular expression that matches the IP address of the connecting host. Other `Valve` classes cater for other rules (e.g. `RemoteHostValve` for matching host names).

Once the changes above have been made, you should be presented with an authentication dialog when accessing the manager URL. If you enter the details you have supplied in `tomcat-users.xml` you should have access to the Manager.

share improve this answer

edited Mar 30 '17 at 15:13

answered Apr 21 '16 at 15:08

 Kevin Panko
7,004 ● 9 ● 44 ● 55

 Matt Innes
1,668 ● 1 ● 10 ● 12

Note If the deployment seems to be taking much longer than expected and nothing is showing up in the logs: 1. Wait - if it takes longer than and hour+your health check time, it should fail and give you an error. It will likely take longer than expected on the initial run of the deployment. 2. Adjust the health check and de-registration delay times.

Additional Task Definitions

Task/Service Name	Container Name	Image	Ports	Volumes	Mount Points	Environment Variables	Entry Point
petclinic	tomcat8	tomcat:8-jre8	8080	/webapps	[TODO path to apache folder /opt/tomcat/webapps???		
			80				

Create EC2 ECS AIM

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

Create Petclinic ECS Service

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

Set up CodeDeploy Application

Create Build Step in Jenkins

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`

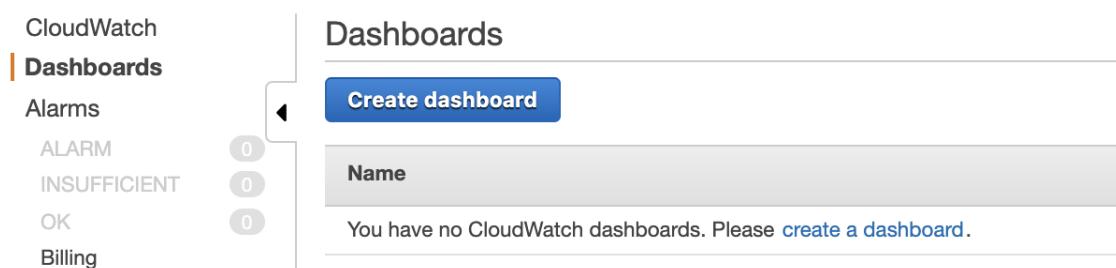
object key if the path within the bucket eg target/petclinic.war

[Return to Table of Contents](#)

CloudWatch Dashboard

In this section we cover using CloudWatch to monitor deployments, builds and other AWS services using CloudWatch Dashboards.

- Navigate to the CloudWatch Console, select Dashboards from the left sidebar, and select Create Dashboard



- Give your dashboard a name and press Create Dashboard

Create new dashboard

Dashboard name:

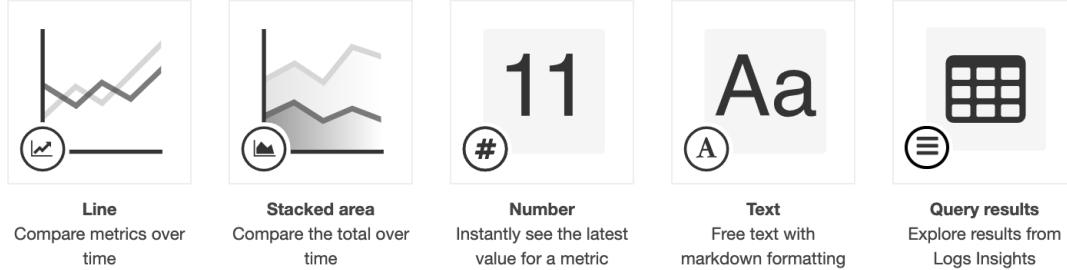
student

Cancel **Create dashboard**

- When prompted select Stacked area, the press Configure

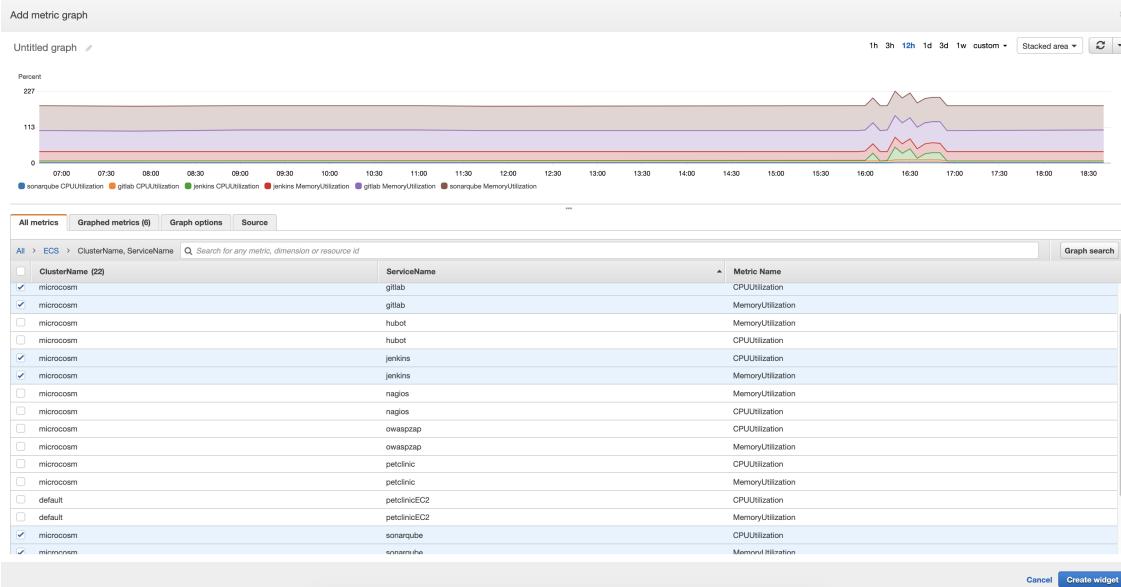
Add to this dashboard

Select a widget type to configure and add to this dashboard.



Cancel **Configure**

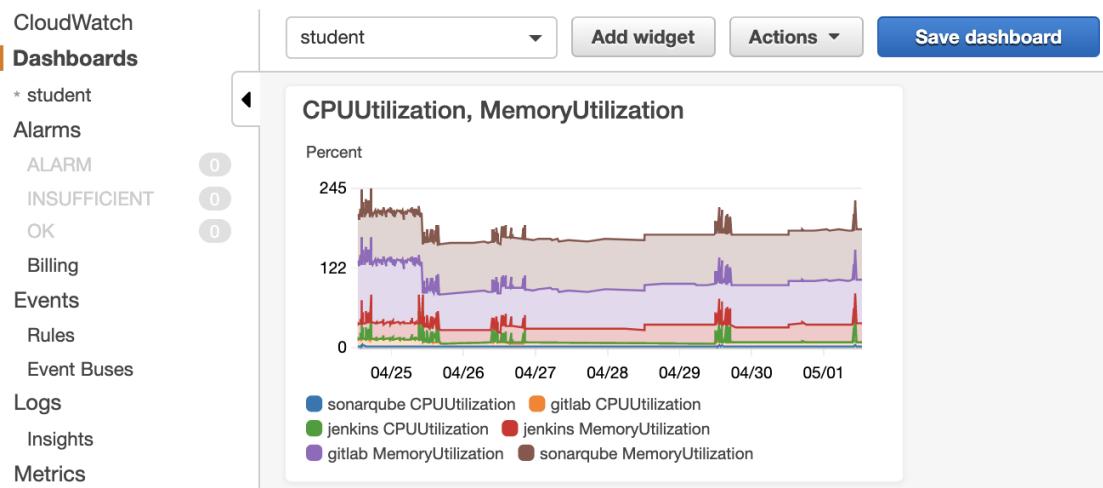
- Select ECS. Sort the rows by Service Name. Select the checkboxes next to your desired services.
- Press create widget



- Back on the dashboard screen, select your dashboard from the list

The screenshot shows the CloudWatch Dashboards interface. On the left, there's a sidebar with a list of services: CloudWatch, Dashboards, student, Alarms, ALARM (0), INSUFFICIENT (0), OK (0). In the center, there's a large 'Create dashboard' button. Below it, there's a 'Name' field with 'student' typed in.

- View the dashboard you created
- Select a time frame from the top right corner
- From the widget setting, feel free to rename or edit the widget



- Add multiple widgets

Add metric graph

Untitled graph

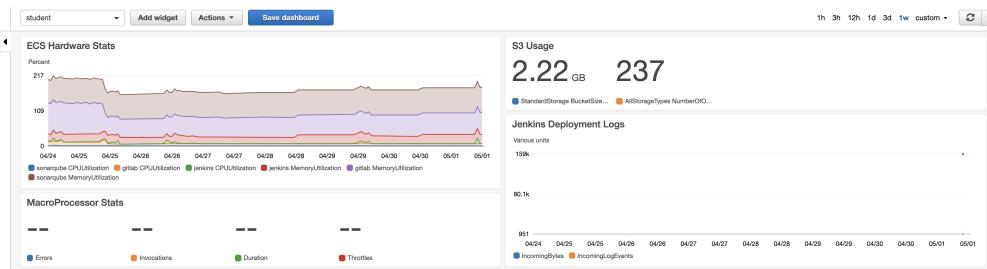
Errors Invocations Duration Throttles

All metrics	Graphed metrics (4)	Graph options	Source
All > Lambda > By Function Name	Search for any metric, dimension or resource id		
<input type="checkbox"/> FunctionName (8)			
<input checked="" type="checkbox"/> MacroProcessor			
<input checked="" type="checkbox"/> MacroProcessor			
<input checked="" type="checkbox"/> MacroProcessor			
<input checked="" type="checkbox"/> MacroProcessor			
Metric Name	Errors	Invocations	Duration
			Throttles

- Explore the possibilities for creating widgets.

- Try creating a graph of CodeDeploy logs and/or jenkins deployment logs.

- Try creating a chart of CPU usage for all of the instances that are more likely to be persistent.



Note that using lambda functions and/or markdown, you are able to display specific errors, logs or log analysis

ref. (<https://aws.amazon.com/blogs/aws/cloudwatch-dashboards-create-use-customized-metrics-views/>)

[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

This screenshot shows the AWS Management Console with the URL <https://us-east-2.console.aws.amazon.com/console/home?region=us-east-2>. The top navigation bar includes 'Services', 'Resource Groups', and a user dropdown for 'Administrator @ cert-microcosm'. The main content area is titled 'AWS Management Console' and contains a sidebar with 'AWS services' (Find Services, Recently visited services: Billing, IAM, EKS, ECR, ECS), 'Build a solution' (Launch a virtual machine, Build a web app, Build using virtual servers), and links to 'Connect an IoT device', 'Start a development project', and 'Register a domain'. The right side features a 'My Billing Dashboard' section with links to 'Access the AWS Console', 'Explore AWS', 'Amazon Redshift', 'Run Serverless Containers with AWS Fargate', 'Scalable, Durable, Secure Backup & Restore with Amazon S3', and 'AWS Marketplace'.

- Explore your billing information, current charges and projections

This screenshot shows the 'Billing & Cost Management Dashboard' at the URL <https://console.aws.amazon.com/billing/home?region=us-east-2#/>. The left sidebar lists 'Cost Management' options: Cost Explorer, Budgets, Cost & Usage Reports, Cost allocation tags, Billing, Bills, Payment history, Credits, Preferences, Billing preferences, Payment methods, Consolidated billing, and Tax settings. The main content area has a 'Billing & Cost Management Dashboard' title. It includes a 'Getting Started with AWS Billing & Cost Management' section with links to AWS Budgets, Cost Explorer, and Reserved Instances (RIs). Below this is a 'Spend Summary' chart showing a month-to-date balance of \$0.50. The chart compares the last month's costs (\$0.00) with the month-to-date costs (\$0.50). To the right is a 'Month-to-Date Spend by Service' chart showing a donut chart with a value of '\$0.5' and a table of costs by service: Route53 (\$0.50), ECS (\$0.00), ECR (\$0.00), S3 (\$0.00), Other Services (\$0.00), Tax (\$0.00), and Total (\$0.50).

[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.

The screenshot shows the Jenkins Manage Jenkins interface under the Sonatype Nexus section. A dropdown menu titled "Add Nexus Repository Manager Server" is open, showing two options: "Nexus Repository Manager 2.x Server" (which is highlighted) and "Nexus Repository Manager 3.x Server".

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

The screenshot shows the Jenkins "Add Credentials" modal window. The "Kind" is set to "Username with password". The "Scope" is "Global (Jenkins, nodes, items, all child items, etc)". The "Username" is "admin", and the "Password" is represented by several dots. The "ID" field contains "nexusadmin". The "Description" field is empty. At the bottom are "Add" and "Cancel" buttons, with "Add" being highlighted.

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

The screenshot shows the Jenkins configuration for a Nexus Repository Manager 2.x Server. The "Display Name" is "Nexus", "Server ID" is "hello1234567890bobmyfriend", "Server URL" is "http://18.221.105.75:8081/nexus", and the "Credentials" dropdown shows "admin/*****" with an "Add" button. Below the form, a message says "Nexus Repository Manager 2.x connection succeeded (2 hosted release Maven 2 repositories)" and there is a "Test connection" button.

- 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

Sonatype™

Artifact Search

Advanced Search

Views/Repositories

Repositories

Help

Welcome

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

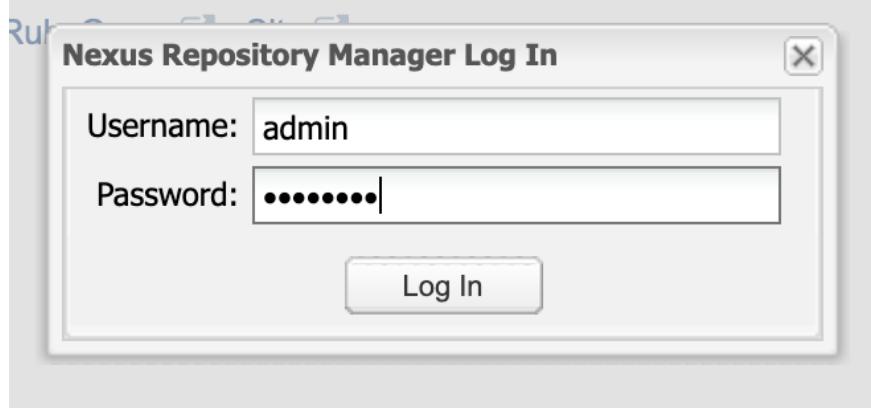
Maven .NET/NuGet Node/npm OSGI P2 RPM/YUM RubyGems Site

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

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	Release	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/shadows/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

- Select Users on the Left

- Click Add User

The screenshot shows the AWS IAM service's 'Users' section. On the left, there's a navigation menu with options like Dashboard, Groups, Roles, Policies, Identity providers, Account settings, Credential report, and Encryption keys. The 'Users' option is selected. In the main content area, there's a search bar at the top labeled 'Find users by username or access key'. Below it is a table with columns: User name, Groups, Access key age, Password age, Last activity, and MFA. One row is visible, showing 'smorley' under 'User name', 'admin' under 'Groups', and '5 days' under both 'Access key age' and 'Password age'. The 'Last activity' and 'MFA' columns show '5 days' and 'Not enabled' respectively. At the bottom of the table, there are buttons for 'Add user' and 'Delete user'.

- 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

The screenshot shows the 'Add user' wizard, step 1 of 5. It's titled 'Set user details'. A note says you can add multiple users at once with the same access type and permissions. The 'User name' field is set to 'Administrator'. There's a link to 'Add another user'. Under 'Select AWS access type', two checkboxes are checked: 'Programmatic access' (which enables access key ID and secret access key) and 'AWS Management Console access' (which enables a password). Under 'Console password*', the 'Custom password' radio button is selected, and the password 'TestPassword1234!@#%' is entered. The 'Show password' checkbox is checked. A note below says 'User must create a new password at next sign-in'. At the bottom, there's a note about required fields and a 'Next: Permissions' button.

- 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

Group	Attached policies
admin	AdministratorAccess

Showing 1 result

Create group Refresh

Search

Cancel Previous Next: Tags

Feedback English (US)

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

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

Create group Cancel Previous Next: Tags

Feedback English (US)

- Select the created group

- Press Next:Tags

Add user

Set permissions

Add user to group

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

Showing 2 results

Cancel Previous Next: Tags

- Add tags if desired
- Click Review

Add tags (optional)

Key	Value (optional)	Remove
administrators		x
Add new key		

You can add 49 more tags.

Cancel Previous Next: Review

- After reviewing, click Create User

Review

User details

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

Permissions summary

Type	Name
Group	Administrators

Tags

Key	Value
administrators	(empty)

Create user

- On the Success Page:

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

- Press close

User	Access key ID	Secret access key	Email login instructions
Administrator	AKIAWOJJKHRHJCJDYDAKNE	ZNHgDgrWBqnBR+K2AxGn 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.

[Return to Table of Contents](#)

Create Key Pair

Key pairs are used for SSH and other authentication with AWS Amazon Machine Image instances (AMIs). * From the EC2 Service Dashboard, select Key Pairs under Network & security on the left * Select Create Key Pair

The screenshot shows the AWS EC2 service dashboard. On the left, a sidebar lists various services: EC2 Dashboard, Events, Tags, Reports, Limits, Instances (Instances, Launch Templates, Spot Requests, Reserved Instances, Dedicated Hosts, Capacity Reservations), AMIs (AMIs, Bundle Tasks), ELASTIC BLOCK STORE (Volumes, Snapshots, Lifecycle Manager), NETWORK & SECURITY (Security Groups, Elastic IPs, Placement Groups, Key Pairs, Network Interfaces), LOAD BALANCING (Load Balancers, Target Groups). The 'Key Pairs' option is selected and highlighted with an orange border. The main content area has a heading 'Select a key pair'. At the top of this area, there are three buttons: 'Create Key Pair' (highlighted in blue), 'Import Key Pair', and 'Delete'. Below these buttons is a search bar with the placeholder 'Filter by attributes or search by keyword'. A message states 'You do not have any Key Pairs in this region.' followed by 'Click the "Create Key Pair" button to create your first Key Pair.' A large blue 'Create Key Pair' button is located at the bottom right of the content area. The browser's address bar shows the URL: https://us-east-2.console.aws.amazon.com/ec2/v2/home?region=us-east-2#KeyPairs:sort=keyName. The top right corner shows the user 'Administrator @ cert-microcosm' and the region 'Ohio'.

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

The screenshot shows the AWS EC2 Dashboard with the 'Key Pairs' section selected. A modal window titled 'Create Key Pair' is open, prompting the user to enter a key pair name ('Administrator1'). The main page displays a message indicating no key pairs are present in the region.

EC2 Dashboard

Events

Tags

Reports

Limits

INSTANCES

Instances

Launch Templates

Spot Requests

Reserved Instances

Dedicated Hosts

Capacity Reservations

IMAGES

AMIs

Bundle Tasks

ELASTIC BLOCK STORE

Volumes

Snapshots

Lifecycle Manager

NETWORK & SECURITY

Security Groups

Elastic IPs

Placement Groups

Key Pairs

Network Interfaces

LOAD BALANCING

Load Balancers

Target Groups

Create Key Pair Import Key Pair Delete

Filter by attributes or search by keyword

You do not have any Key Pairs in this region.

Click the "Create Key Pair" button to create your first Key Pair.

Create Key Pair

Create Key Pair

Key pair name: Administrator1

Cancel Create

Select a key pair

* View key pair details

https://us-east-2.console.aws.amazon.com/ec2/v2/home?region=us-east-2#KeyPairs:sort=keyName

Administrator @ cert-microcosm Ohio Support

Create Key Pair Import Key Pair Delete

Filter by attributes or search by keyword

1 to 1 of 1

Key pair name	Fingerprint
Administrator1	fa:c0:90:4b:a7:bf:82:9f:3d:13:1c:73:dd:83:eb:b4:bd:74:f2:26

Key Pair: Administrator1

Key pair name: Administrator1
Fingerprint: fa:c0:90:4b:a7:bf:82:9f:3d:13:1c:73:dd:83:eb:b4:bd:74:f2:26

Feedback English (US)

Administrator1.pem

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

[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](#)

Requesting Changes to Default AWS Limits

[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

The redesigned AWS CloudFormation console is available now
We've completely redesigned the console to improve the overall look and feel. Try it out now and provide us feedback.

Create a stack

AWS CloudFormation allows you to quickly and easily deploy your infrastructure resources and applications on AWS. You can use one of the templates we provide to get started quickly with applications like WordPress or Drupal, one of the many sample templates or create your own template.

You do not currently have any stacks. Choose **Create new stack** below to create a new AWS CloudFormation stack.

Create new stack

Create a StackSet

A StackSet is a container for AWS CloudFormation stacks that lets you provision stacks across AWS accounts and regions by using a single AWS CloudFormation template.

Create new StackSet

Design a template

Templates tell AWS CloudFormation which AWS resources to provision and how to provision them. When you create a CloudFormation stack, you must submit a template.

To build and view templates, you can use the drag-and-drop tool called AWS CloudFormation Designer. You drag-and-drop the resources that you want to add to your template and drag lines between resources to create connections. To use Designer to create a template or to open and modify a template, choose **Design template**.

Design template

- To create or view a template, on the Select Template page, click Design 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](#).

Design template

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

Select a sample template

Upload a template to Amazon S3
[Choose File](#) No file chosen

Specify an Amazon S3 template URL

Cancel **Next**

- 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)

New types Open Save V2 Sample templates AppSync ApplicationAutoScaling Athena AutoScaling AutoScalingPlans Batch Budgets CertificateManager Cloud9 CloudFormation CloudFront CloudTrail CloudWatch CodeBuild CodeCommit CodeDeploy

File: "Microcosm_modTemplate.template"

LambdaExecutionRole Role

LambdaPolicy Policy

MacroFunction Function

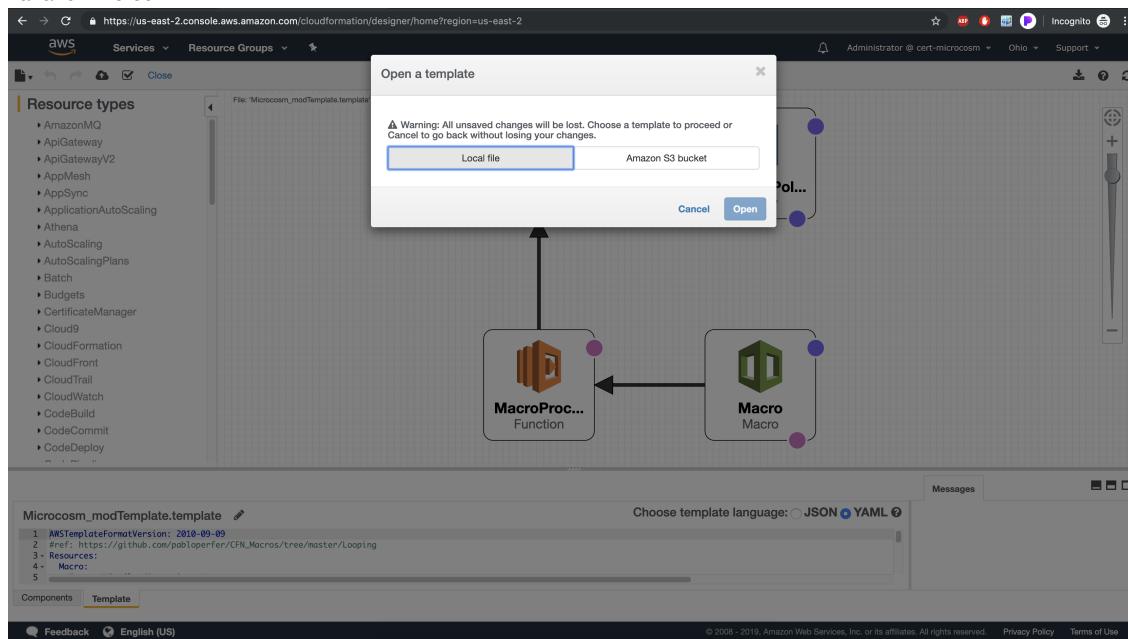
Macro Macro

Choose template language: JSON YAML

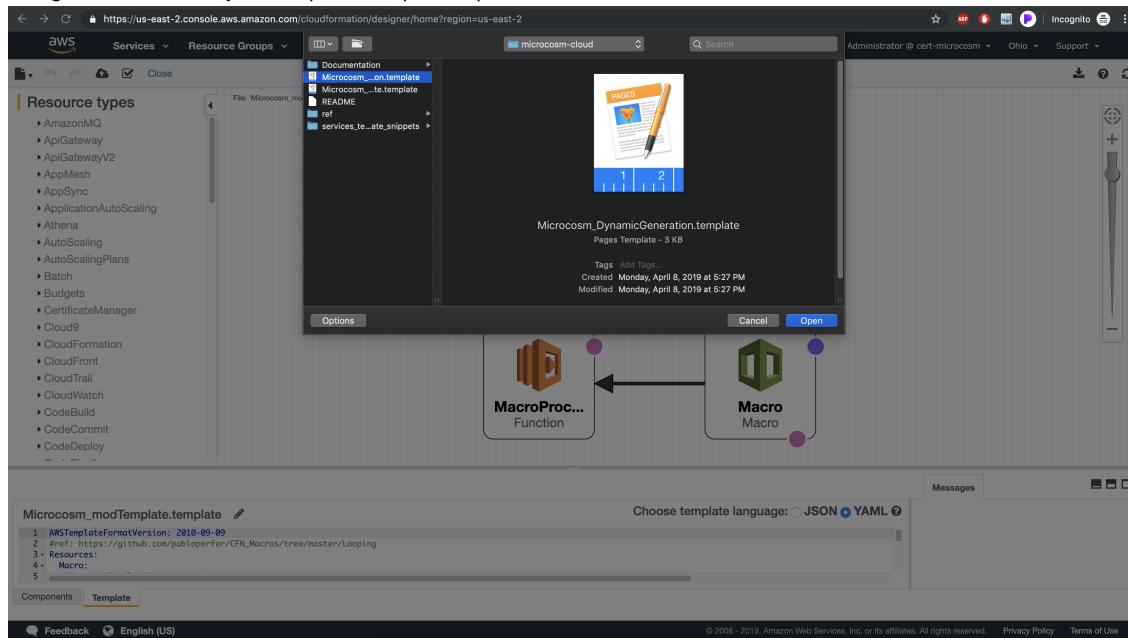
```
1 AWSTemplateFormatVersion: 2010-09-09
2 #Ref: https://github.com/pablopfer/CFN_Macros/tree/master/Looping
3 Resources:
4 - Macro:
5
```

Components Template

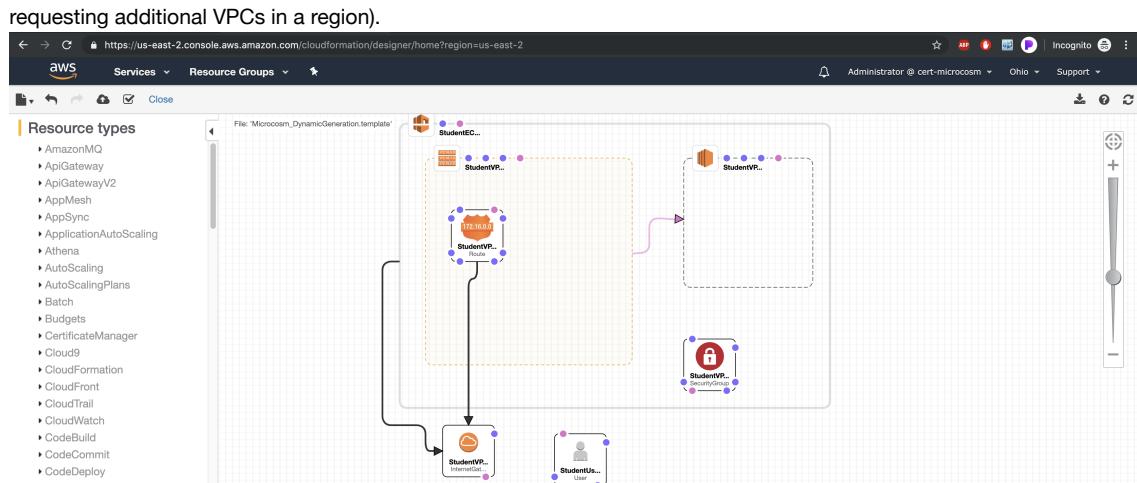
- Select the source of your template. It is good practice to name your templates with the extension `.template` or some variation thereof



- 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



Microcosm_DynamicGeneration.template Edit

Choose template language: JSON YAML Edit

```

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

```

[Components](#) [Template](#)

Feedback English (US)

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

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

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

CloudFormation Stacks Create Stack

Administrator @ cert-microcosm Ohio Support

Create stack

Select Template

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

Design template

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

Select a sample template

Upload a template to Amazon S3 Choose File: Microcosm_mo...te.template

Specify an Amazon S3 template URL

Cancel Next

Feedback English (US)

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

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

- Press Next

Specify Details

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

Stack name

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

- 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

Enter role arn

▼ Rollback Triggers

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

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

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](#)

- 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	Disabled
Termination Protection	Disabled
Timeout	none
Rollback on failure	Yes

Capabilities

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

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

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

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

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

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

CloudFormation > Stacks

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

Showing 1 stack

Overview Outputs Resources Events Template Parameters Tags Stack Policy Change Sets Rollback Triggers

Filter by: Status - Search events

2019-04-11

Status	Type	Logical ID	Status Reason
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	IambdaExecutionRole	

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

- 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

Choose File

Specify an Amazon S3 template URL

Cancel Next

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

- 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.

Stack name

InstanceType WebServer EC2 instance type

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

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

Cancel Previous Next

- 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

Enter role arn

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

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

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

No rollback triggers provided

Advanced

Notification	<input checked="" type="checkbox"/>
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

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

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**

- 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

Functions (1)

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

- 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)

```

    1 import json
    2 import copy
    3
    4 def lambda_handler(event, context):
    5
    6     FinalFragment= event["fragment"]
    7
    8     Number = event["TemplateParameterValues"]["NumberofUsers"]
    9     #I declare a new dictionary to have the Resources object
   10
   11     TemplateDetailsVar = {}
   12     TemplateDetailsVar = FinalFragment["Resources"]
   13
   14     for i in range(int(Number)):
   15         #I modify the IAM User resource logical id in order to add it later to "Resources" object the number of times requested
   16
   17         numStr(i)
   18         TemplateDetailsVar["StudentUser"+num] = copy.deepcopy(TemplateDetailsVar["StudentUser"])
   19         TemplateDetailsVar["StudentEC2VPC"+num] = copy.deepcopy(TemplateDetailsVar["StudentEC2VPC"])
   20
   21         TemplateDetailsVar["StudentVPCSUBNET"+num] = copy.deepcopy(TemplateDetailsVar["StudentVPCSUBNET"])
   22         TemplateDetailsVar["StudentVPCSUBNET"+num]["Properties"]["VpcId"]["Ref"] = "G"+str(i).format(TemplateDetailsVar["StudentVPCSUBNET"+num]["Properties"]["VpcId"])
   23
   24         TemplateDetailsVar["StudentVPCSecurityGroup"+num] = copy.deepcopy(TemplateDetailsVar["StudentVPCSecurityGroup"])
   25         TemplateDetailsVar["StudentVPCSecurityGroup"+num]["Properties"]["VpcId"]["Ref"] = "G"+str(i).format(TemplateDetailsVar["StudentVPCSecurityGroup"+num]["Properties"]["VpcId"])
   26
   27         TemplateDetailsVar["StudentVPCRoutingTable"+num] = copy.deepcopy(TemplateDetailsVar["StudentVPCRoutingTable"])
   28         TemplateDetailsVar["StudentVPCRoutingTable"+num]["Properties"]["VpcId"]["Ref"] = "G"+str(i).format(TemplateDetailsVar["StudentVPCRoutingTable"+num]["Properties"]["VpcId"])
   29         TemplateDetailsVar["StudentVPCRoutingTable"+num]["Dependencies"] = []
   30         dependencies.append("G"+str(i).format(num))
   31         TemplateDetailsVar["StudentVPCRoutingTable"+num]["Properties"]["DependsOn"] = "G"+str(i).format(TemplateDetailsVar["StudentVPCRoutingTable"+num]["Properties"]["DependsOn"])
   32         TemplateDetailsVar["StudentVPCRoutingTable"+num]["DependsOn"] = dependencies
  
```

Throttle Qualifiers Actions Select a test event Test Save

- 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

The screenshot shows the AWS Lambda Monitoring page for the MacroProcessor function. The top navigation bar includes the AWS logo, Services, Resource Groups, and tabs for Lambda, Functions, and MacroProcessor. The main content area is titled "MacroProcessor" and displays "CloudWatch metrics at a glance". It includes four main sections: "Invocations" (Count: 2, Duration: 1 millisecond, 0.64 ms minimum, 0.28 ms average, 0.5 ms maximum), "Duration" (Count: 1, 0.5 ms), "Error count and success rate (%)" (Count: 1, 100%), and "Throttles" (Count: 1). Below these are sections for "IteratorAge" (1) and "DeadLetterErrors" (1). Buttons for "View logs in CloudWatch" and "View traces in X-Ray" are available. A "Save" button is also present.

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

The screenshot shows the AWS CloudWatch Log Stream list for the /aws/lambda/MacroProcessor group. The left sidebar lists various AWS services like CloudWatch, Dashboards, Alarms, Billing, Events, Rules, and Logs. The Logs section is selected. The main pane shows a list of log streams with their last event time. The most recent log stream, "2019/04/11/\$LATEST", is selected and expanded, showing its contents.

Last Event Time	Log Stream
2019-04-11 10:26 UTC-6	2019/04/11/\$LATEST
2019-04-09 12:44 UTC-6	2019/04/09/\$LATEST
2019-04-09 11:53 UTC-6	2019/04/09/\$LATEST
2019-04-09 17:07 UTC-6	2019/04/09/\$LATEST
2019-04-08 17:03 UTC-6	2019/04/08/\$LATEST
2019-04-08 16:55 UTC-6	2019/04/08/\$LATEST
2019-04-08 16:46 UTC-6	2019/04/08/\$LATEST
2019-04-08 16:22 UTC-6	2019/04/08/\$LATEST
2019-04-08 16:12 UTC-6	2019/04/08/\$LATEST
2019-04-08 16:09 UTC-6	2019/04/08/\$LATEST
2019-04-08 16:07 UTC-6	2019/04/08/\$LATEST
2019-04-08 16:04 UTC-6	2019/04/08/\$LATEST
2019-04-08 15:59 UTC-6	2019/04/08/\$LATEST
2019-04-08 15:50 UTC-6	2019/04/08/\$LATEST
2019-04-08 15:48 UTC-6	2019/04/08/\$LATEST
2019-04-08 15:34 UTC-6	2019/04/08/\$LATEST
2019-04-08 15:22 UTC-6	2019/04/08/\$LATEST
2019-04-08 15:02 UTC-6	2019/04/08/\$LATEST
2019-04-08 14:47 UTC-6	2019/04/08/\$LATEST
2019-04-08 14:28 UTC-6	2019/04/08/\$LATEST
2019-04-08 14:13 UTC-6	2019/04/08/\$LATEST
2019-04-08 14:11 UTC-6	2019/04/08/\$LATEST
2019-04-08 13:29 UTC-6	2019/04/08/\$LATEST

- View the logs generated by the lambda function

The screenshot shows the AWS CloudWatch Log Event Viewer for the most recent log stream, "2019/04/11/\$LATEST". The left sidebar shows the CloudWatch Log Stream list. The main pane displays log events from April 11, 2019, at 16:26:34. The events show the start request, print event, print resources, student user properties, end request, and report request details. No newer events were found at the moment.

Time (UTC +0:00)	Message
2019-04-11 16:26:34	START RequestId: f40b8d79-75d1-4182-a8eb-aed854299233 Version: \$LATEST
2019-04-11 16:26:34	PRINT EVENT:
2019-04-11 16:26:34	{"region": "us-east-2", "accountId": "443007076818", "fragment": "AWS::TemplateFormatVersion", "version": "2010-09-", "description": "This template generates the requested number of IAM users that are defined in the StudentUser section.", "studentUser": {"Type": "AWS::IAM::User", "Properties": {"Path": "/", "LoginProfile": {"Password": "Sbzq\$UFLxw9HF-49eRtf"}, "PasswordResetRequired": true}, "Metadata": {"AWS::CloudFormation": {"CreateDependsOn": "IamRole", "UpdateDependsOn": "IamRole"}}, "endRequest": {"requestId": "f40b8d79-75d1-4182-a8eb-aed854299233", "duration": 0.64 ms, "billedDuration": 100 ms, "memorySize": 128 MB, "maxMemoryUsed": 40 MB}}
2019-04-11 16:26:34	REPORT RequestId: f40b8d79-75d1-4182-a8eb-aed854299233 Duration: 0.64 ms Billed Duration: 100 ms Memory Size: 128 MB Max Memory Used: 40 MB

o IAM User

- From the top services menu select IAM
- View the Student user created and its properties

This user information can be given to anyone to allow them to login and access the allowed services

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

User name	Groups	Access key age	Password age	Last activity	MFA
Administrator	Administrators	Yesterday	Yesterday	Today	Not enabled
[REDACTED]	[REDACTED]	6 days	6 days	6 days	Not enabled
VPCCreation-StudentUser-WMCVKHLO8WCF	None	None	Today	None	Not enabled

Text

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

◦ VPC

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

The screenshot shows the AWS VPC Dashboard. The left sidebar lists various VPC components: 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, and Virtual Private Gateways. The main content area displays a table of VPCs with columns: Name, VPC ID, State, IPv4 CIDR, IPv6 CIDR, DHCP options set, Main Route table, and Main Network ACL. Two VPCs are listed:

Name	VPC ID	State	IPv4 CIDR	IPv6 CIDR	DHCP options set	Main Route table	Main Network ACL
vpc-01ca865d17b887b66	vpc-01ca865d17b887b66	available	10.0.0.0/24	-	dopt-5f946034	rtb-0a1f862380212061b	ad-0a03d1f85b31db190
vpc-9dd0c3f5	vpc-9dd0c3f5	available	172.31.0....	-	dopt-5f946034	rtb-d764bcfc	ad-597f8132

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

[Return to Table of Contents](#)

APPENDIX B

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

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

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

- Install the AWS CLI

```
mac-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: docutilts>=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]: AKIAWOKJHRHJCODYDAKNE
AWS Secret Access Key [None]: ZNHgDqrWBqnBR+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 -u AWS -p eyJwYXlsb2FkjoiKzhsTwCtCL2krRjFol3plVzFVcjBlaJNvbWhPU6pMwW50K91XNVJLUUpXR1pHTGFDTQMzQ2VBY3hYZ29XT0R
SS1yZKpjdGvZ0kvWhHZz4Y21EanJTXQFCvZJemdXWVNScEx4Ymg3NDbvbw1JQTFHQ8SwWWdmRnZ1dB1Qa0pws1Nza0hPZkR5N08uSwTYYBJWxOrZ
1EZwdNBc1cydnF5bjy0WDhau014ab3F2bkdoV09hkZM4bzLPNwdBVWtMwmwPsMXhewPbQncycNFBXVDldclck65StGt2enVrSxhmQjFhZ20zek5sMkZhRTFydh
ibEw3NnI2ZGQvNzNgTeUAc3Fr1g1SG1MdF1GY0ozT1RmMkJKckQ1cjlUUkr6T6lxz22ZMRf14eG1PMVVIdjAwaUovb2hEa1VtOFJMcDM3SGhabzJ1MnpjTkRvc
3gvbVNROW5EM2dvZ1Ntl1YwUFhZcm1pdkkvNFJERERxYAS5eXZmcHVSAzV2OUF6MGZuc1d3NnJUSS9IZHZCzzFxevVnC3hnSm08RXY5TUtaOWZzb0pvRF1XSlh
6N22ndwPIRXMoY3ayM2duSEhodUJrN09FVK1INVgvV3NIMEFMaHBvYI2UXmbzJuVa1Rvd1pMd1M0Uk5zL01WMThxb6hRRk4vU1iSw9qaWxNRGxnMHCQkorV
GxaTYN0YFVrcKZMNE5Bu1mdFNSL0NnC2hTSd3aV1KemZIZWcqvYDZZVZiekowT3vtNDRPWTY3RDRzb3JaMTRuRWZ3bk1Iduo5SUNPVEFy0p0JTms0aEVDUuF
JQ2x3SFU1S0p5REZOYXlyRFFeeQQ5cUEGUEJxR314bUrxeFYwd0@0MkozUy8vSE1DTTc3QWerl0UzNFBNaEt1N2E5a3k5YzVPYXQzUGNxelFEUkJaaFYzYzQyd
TRkq1VQUwtDZUuyUDV4WEFLUnJNOUZ4RFJwT315by9IUXXH91wEQxZ0htUvhRx30Uh1Wn5aEVXdnps3bnVHaC9tajVUUU1SVjNRL1YzbW9LMGZ0VW9Hw9
Fwm1xRHrjubHSU9SUTZoZGMzWgxZS9zNu9URfl1SzjvSkk3b1E1yVzdCM09jMXd0Ngi0UhVNXNpYUhjK0l0Nfpob25221vMTN40GxmRVRwheHFLN011Q1g5N
zdDeERVTjdnbksvZSPDd1JHVVHHS8NIousxbw1Lb1VSL1R5V1Zzeke1MVHU3OTk5MXBGu5CK2pmveVQ004ZXVmNkQwaHc2cmw5khkbw1BYUZnhfuUW56buQ
xbmcxUWw0liUuh6aURZm5tNFZG0WnmaVSY21DWGXJX0FMWwpazXBjUFJWM0FkCw9VUe1kd3NtD2NycHnsZHBUC0x5YKj1UVdvRmXanRvT2YzW6p0bzVke
G8zU3R2bmox5kVmU9yUgkxTvh2bUfcLiszMkd4an1CMWY3K29DMzRil3orVkuVUdFnMjTR22FR0pTVxMckJ3PT01LCJkYXRAha2V51jo1QVFQkFiaikI3L21
nd01nNE50d2f1cnhTSV14NeHmbnh1R2MvNDh1Rhd2d0RwT11XWmdBQUF1NHdmQV1KS29aSwh2Y05BUWHb0c4d23RSUJBR4vQmdrcWhra1c5dzBCQndFd0hnW
UpZSVpJQVdVREJBRXVNQKVFRExRYKVi0zNwY0htR3c2bk5BSUJF5SUE3eUUVeWI1Mkp0O0X15MnMzeTRUd3JiNm9nRGv0d1aNCsw0V1wcm1kM0dPL11vai96TU9
BNEpiNvprJHM4QWFqWnRNM09v11IxY1FLNmJsd0iLCJ2ZXJzaw9u1joiMiisInR5cGUio1jEQVrBX0tFWSIsImV4cGlyYXRpB24i0jE1NTQ5njQ40DF9 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 eyJwYXlsb2FkIjoiKzsTwtCl2krRf0l3p1VzFvcjBlajNvbWhPUPgPMW
50k01XNV1XUlwUxPr1HtGFDGTMZq2BVBy3Yh29Xt0RSs1YzKp1dGdVz0kaVhWzz2Y1Ean1JtQxFcvAzJemdXvNsCEx4yMng3DnbV1QfTq58wBmRnZ
11Db1bPaQawpS1Na0zP2kR5n0uSwUvtVvBjWx0z1LzEwDnb1cycdfn5fbjY0WduhA01b3F2kbdoV9khMz4b1zPnWdbVtMwmVsXhwelBpBcnyNfFBXvDdc1
dkeG5SbGt2enVrSxhmqJfZ29zeek5sMkZhRTFydnhibEw3Nn1z2QvNzNgEtu4c3FrA1g1Sc1MdflGy0ozT1RmNmkJckQ1c1j1UuRk6T6C1zX22Mfr14eG1pMvvi
djaAwUvb2hEa1vTmJcMc3SghabJ1Mnp1c3gvbNzR05EM2dvZtNl1YwUhZcm1pdkvNfReRxMa5eXzmcHvSaZoF6MgZuc1d3nnJUSS91Z
JczXfZeVvNchsnMo0RXY5TuTa0Wzb2p0rfX1xsLh6NzDwYrXtm03YAvM2uSehd0Ujrn09Fvkl1Nvqv3NzIMEfMaBhYI2UxmzbjuJa1Rvd1pMd1U0k5z
L011WtMtxbhbRk4uu1iSw9gaWxNrGxnmHFcKqorGxJaYTNYOfVrcKzMN5B8U1mdFNSL0NNc2tSuds3aV1kemZIzWcyVdZzvWziekowT3VtNDRPWTy3RdrzB3
jaTrtRwZ3bK1jd0u5SUNPVEFyK0jtMs0aEvDfUfQJ2x3Qs1U1S0p5REZOYX1yRFeEzG05cU5GUEJxR314bURxeTywd0oMkr0y8vSE1DttC3QWeL0zNfbn
aEt1N2E5a3k5yZpVYXpQzUgxNeUfeUkJaFyZqyDtrK1QvUWtDzuUw4VfEwLJnuOz4RfZt315by91UXvHNg91wEqxZ0htvhnRx3c0U1hwn5aEvdn
p3bnWHaC9tajvuUu1svjNrxL1yzbw9LMGZ0vW9HwmP9WmIxRrhUjBhsu9SutDzGmz2GxrzS9zNu9UrFl1szVskk3b1ey1zdcm99jMxdnG1oUhPvnXNpYUjh
K1o1Nfp0b252z1vrtMn04McrMrvreHfln01i1Qg5NzDdeErVtJdnbfbszS9D1JvhWsh0Sni0Usxbw1Lb1Vsl1R5V1zLek1MVh30T5MxBGdU5C2pkpveVQ0u
o4zxVmNkQwhac2cmw5bkhkbw1Byuznhfu0W55b0UqxmbcJxWwA1liUwaharUzz5tNfZG0wnMaVzsyDwgxX0FmwpazXbjUWf0Mkq9vUe1k3Dnt2Ny
ChnszHbcU05yK1JuvDrVmzXanrT2YzWg0p0bzKveG8zU3RbmoxsPvMpUyUgxKtvh2BuflKsRh4d10nMwy3K29DmzRil3orVkuJwUdFnmJtrZ2FrR0pTkv
XmcKj3P0t1lcLjKyrhxa2V5IjoiVqfFQKzIak13L2ln01nNE5qd2f1ChnsTsv14Nehmhn1R2MvndhiRhd20RwT1LxWmdBu0fInhdqV1ks29asWhY05Bwnn
b0c4d2JrsuJbrEjVQmdrcwhzai5cdzBcqndf0hnwzpSVjPQvDreJbrXvNqKvfReYrykViQzN9y0htr3c2b5kbsujsFE3euUveW11KmP00X15MmzeteRud3
JinNm9Rvg0d1larNcswoIvCim1k0Mdpl1lv196tU9BNEpiVpruhM4QwfqdnRm09v11ixY1flNmjsdz0iLCj22XjzaW9uIjoiMiisInR5cGuio1jeQvrbx0tF
WsIsImw4cGlyXRp24i0je1ntQ5NjQ40df9 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
MES

mac-topazpomme:devops-microcosm smorley$ docker container ls --all
CONTAINER ID        IMAGE               COMMAND             CREATED            STATUS              PORTS
PORTS
NAMES
sonarqube:lts      "/bin/run.sh"        7 days ago         Exited (0) 42 minutes ago
sonarqube
gillax/hubot-slack-jenkins  "/bin/hubot -a slack"    7 days ago         Exited (0) 37 minutes ago
hubot
gitlab/gitlab-ce   "/assets/wrapper"       7 days ago         Exited (0) 37 minutes ago
gitlab
sonatype/nexus     "/bin/sh -c 'java ..."    7 days ago         Exited (143) 37 minutes ago
sonatype_nexus
sonatype_nexus
h1kkan/jenkins-docker:lts  "/sbin/tini -- /usr/..."  7 days ago         Exited (143) 37 minutes ago
jenkins
jenkins
owasp/zap2docker-stable  "zap-webswing.sh"       7 days ago         Exited (0) 44 minutes ago
owaspzap
owaspzap

mac-topazpomme:devops-microcosm smorley$ docker tag sonarqube:lts sonarqube:lts0419

```

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

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

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

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

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

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

- Verify tag is correct by relisting images

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

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

```

mac-topazpomme:devops-microcosm smorley$ aws ecr create-repository --repository-name sonarqube
{
    "repository": {
        "repositoryArn": "arn:aws:ecr:us-east-2:443007076818:repository/sonarqube",
        "registryId": "443007076818",
        "repositoryName": "sonarqube",
        "repositoryUri": "443007076818.dkr.ecr.us-east-2.amazonaws.com/sonarqube",
        "createdAt": 1554929452.0
    }
}
[mac-topazpomme:devops-microcosm smorley$ docker tag sonarqube:lts 443007076818.dkr.ecr.us-east-2.amazonaws.com/sonarqube:lts
mac-topazpomme:devops-microcosm smorley$ docker push 443007076818.dkr.ecr.us-east-2.amazonaws.com/sonarqube:lts
The push refers to repository [443007076818.dkr.ecr.us-east-2.amazonaws.com/sonarqube]
68e5087b6fffc: 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 teh AWS ECR Console

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

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

Benefits

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

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](#)

Feedback English (US)

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

Repositories (6)

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

Feedback English (US)

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

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@sha256:4c04456cf...	04/10/19, 3:18:08 PM	285.24	

Feedback English (US)

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

The screenshot shows the AWS ECR console homepage. At the top, there's a navigation bar with 'Services' and 'Resource Groups'. On the right, it shows 'Administrator @ cert-microcosm - Ohio Support'. Below the header, the title 'Amazon Elastic Container Registry' is displayed with the subtitle 'Easily store, manage, and deploy container images'. A subtext explains that ECR is a fully-managed container registry. To the right, there's a 'Create a repository' section with a 'Get Started' button. In the center, there's a diagram titled 'How it works' showing the flow from writing code to deploying containers. To the right of the diagram, there's a 'Pricing (US)' section stating that users pay only for data stored and transferred. Below that is a 'Getting started' section with links to 'What is Amazon ECR?', 'Getting started with ECR', and 'Set up a CI/CD pipeline with ECR'. At the bottom, there are three tabs: 'Fully managed', 'Secure', and 'Launchpad', with 'Launchpad' being the active tab.

The image consists of three vertically stacked screenshots from the AWS ECR console.

Screenshot 1: The first screenshot shows the 'Create repository' page. The 'Repository configuration' section has a 'Repository name' field containing '443007076818.dkr.ecr.us-east-2.amazonaws.com/'. Below the field is a note: 'A namespace can be included with your repository name (e.g. namespace/repo-name.)'. At the bottom are 'Cancel' and 'Create repository' buttons.

Screenshot 2: The second screenshot shows the same 'Create repository' page, but the 'Repository name' field now contains 'microcosm-cloud'. The rest of the interface is identical to the first screenshot.

Screenshot 3: The third screenshot shows the 'Repositories' list after the repository was created. A green banner at the top says 'Successfully created repository'. The 'Repositories (1)' table lists one item: 'microcosm-cloud' with a URI of '443007076818.dkr.ecr.us-east-2.amazonaws.com/microcosm-cloud' and a 'Created at' timestamp of '04/10/19, 11:49:49 AM'. The table includes columns for 'Repository name', 'URI', and 'Created at'. Action buttons for 'View push commands', 'Delete', and 'Create repository' are at the top right of the table.

[Return to Table of Contents](#)

ECS Get Started Wizard

Using Get Started Wizard

Follow the screen shots for reference

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

AWS Services Resource Groups

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

Amazon Elastic Container Service (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](#)

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

AWS Services Resource Groups

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

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](#)

No clusters found [Get Started](#)

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

AWS Services Resource Groups

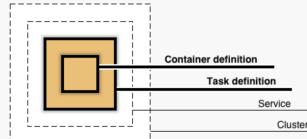
Administrator @ cert-microcosm Ohio Support

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



Container definition

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

sample-app

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

nginx

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

tomcat-webserver

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

custom

image : --
memory : --
cpu : --

[Configure](#)

Task definition

Amazon Container Services

ECR > Repositories > h1kkan/jenkins-docker

Images (1)

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:a2a91bc92...	644.06

[View push commands](#)

Feedback English (US)

Getting Started with Amazon Elastic Container Service

Step 1: Container and Task

Diagram of ECS objects

Container definition

Choose an image for your container before you can define a task definition.

sample-app

image : http2.4
memory : 0.5GB (512)
cpu : 0.25 vCPU (256)

tomcat-webserver

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

Task definition

Edit container

Standard

Container name* jenkins

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

Custom image format: [registry-uri]/[namespace]/[image]:[tag]

Private repository authentication*

Memory Limits (MiB) Soft limit 128

Add Hard limit

Define hard and/or soft memory limits in MiB for your container. Hard and soft limits correspond to the 'memory' and 'memoryReservation' parameters, respectively, in task definitions. ECS recommends 300-500 MiB as a starting point for web applications.

Port mappings

Container port	Protocol
8080	tcp

Add port mapping

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 before you can define a task definition.

sample-app

image : http2.4
memory : 0.5GB (512)
cpu : 0.25 vCPU (256)

tomcat-webserver

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

Task definition

Edit container

STORAGE AND LOGGING

Read only root file system

Mount points

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

Add mount point

Volumes from

Log configuration

Auto-configure CloudWatch Logs

Log driver awslogs

Log options

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

Add key Add value

* Required

Cancel Update

sample-app

image : http2.4
memory : 0.5GB (512)
cpu : 0.25 vCPU (256)

nginx

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

tomcat-webserver

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

jenkins

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)

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

<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
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
Number of desired tasks	1

Cluster

Cluster name	microcosm
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>

AWS Services Resource Groups Administrator @ cert-microcosm Ohio Support

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	pending
Cluster microcosm	complete ✓
Task definition first-run-task-definition:3	complete ✓
Service jenkins-service	pending
Additional AWS service integrations	pending
Log group The log group [/ecs/first-run-task-definition] already exists	complete ✓
CloudFormation stack	pending
VPC	pending
Subnet 1	pending
Subnet 2	pending
Security group	pending

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

AWS Services Resource Groups Administrator @ cert-microcosm Ohio Support

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

Launch Status

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

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

Additional features that you can add to your service after creation

Scale based on metrics

You can configure scaling rules based on CloudWatch metrics

Preparing service : 9 of 9 complete

ECS resource creation	complete ✓
Cluster microcosm	complete ✓
Task definition first-run-task-definition:3	complete ✓
Service jenkins-service	complete ✓
Additional AWS service integrations	complete ✓
Log group The log group [/ecs/first-run-task-definition] already exists	complete ✓
CloudFormation stack EC2ContainerService-microcosm	complete ✓
VPC vpc-0530aacf15c751393	complete ✓
Subnet 1 subnet-0521ea54a64435d4	complete ✓
Subnet 2 subnet-0e50c4d05af5052a7	complete ✓
Security group sg-06107d8d65ef6b2fd	complete ✓

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

AWS Services Resource Groups Administrator @ cert-microcosm Ohio Support

Clusters > microcosm > Service: jenkins-service

Service : jenkins-service

Cluster: microcosm Status: ACTIVE Task definition: first-run-task-definition:3 Service type: REPLICAS 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-0521ea5a64435dd4, subnet-0e50c4d05af5052a7
Security groups*	sg-0f607d8d65ef6b2fd
Auto-assign public IP	ENABLED

Clusters > microcosm > Service: jenkins-service

Service : jenkins-service

Cluster: microcosm Status: ACTIVE Task definition: first-run-task-definition:3 Service type: REPLICAS 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

Clusters > microcosm > Service: jenkins-service

Service : jenkins-service

Cluster: microcosm Status: ACTIVE Task definition: first-run-task-definition:3 Service type: REPLICAS 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

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

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

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-06504d05af5052a7
Private IP: 10.0.1.191
Public IP: 18.191.252.84
Mac address: 06:5e:3e:bd:77:c4

Containers

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

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

aws Services Resource Groups

Administrator @ cert-microcosm Ohio Support

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

Task Definition Latest revision status

first-run-task-definition ACTIVE

Feedback English (US)

Amazon ECS

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

microcosm > FARGATE

1 Services 1 Running tasks 0 Pending tasks

EC2

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

Feedback English (US)

Everything Created By Get Started Wizard

Follow the screen shots for reference

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

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: Disabled
Termination protection: Disabled
Drift status: NOT_CHECKED [View details](#)
Last drift check time:
IAM role:
Description: AWS CloudFormation template to create a new ECS Fargate First Run stack

Outputs

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

Resources

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

Logical ID	Physical ID	Type	Drift Status	Status	Status Reason
AttachGateway	EC2Co-Attac-1AVZP5KOSFL1P	AWS::EC2::VPCGatewayAttachment	NOT_CHECKED	CREATE_COMPLETE	
EcsSecurityGroup	sg-06f07d8d65ef6b2fd	AWS::EC2::SecurityGroup	NOT_CHECKED	CREATE_COMPLETE	
InternetGateway	igw-0db5c3b0358004693	AWS::EC2::InternetGateway	NOT_CHECKED	CREATE_COMPLETE	

Events

Filter by: Status

Date	Status	Type	Logical ID	Status Reason
2019-04-11	CREATE_COMPLETE	AWS::CloudFormation::Stack	EC2ContainerService-microcosm	
08:46:35 UTC-0600	CREATE_COMPLETE	AWS::EC2::SubnetRouteTableAssociation	PublicSubnet1RouteTableAssociation	
08:46:30 UTC-0600	CREATE_COMPLETE	AWS::EC2::SubnetRouteTableAssociation	PublicSubnet2RouteTableAssociation	
08:46:29 UTC-0600	CREATE_COMPLETE	AWS::EC2::Route	PublicRouteVialgw	
08:46:15 UTC-0600	CREATE_IN_PROGRESS	AWS::EC2::SubnetRouteTableAssociation	PublicSubnet1RouteTableAssociation	Resource creation initiated
08:46:15 UTC-0600	CREATE_IN_PROGRESS	AWS::EC2::SubnetRouteTableAssociation	PublicSubnet2RouteTableAssociation	Resource creation initiated
08:46:14 UTC-0600	CREATE_IN_PROGRESS	AWS::EC2::SubnetRouteTableAssociation	PublicSubnet1RouteTableAssociation	
08:46:14 UTC-0600	CREATE_IN_PROGRESS	AWS::EC2::Route	PublicRouteVialgw	Resource creation initiated
08:46:13 UTC-0600	CREATE_IN_PROGRESS	AWS::EC2::Route	PublicRouteVialgw	
08:46:10 UTC-0600	CREATE_COMPLETE	AWS::EC2::Subnet	PublicSubnet1Az1	
08:46:10 UTC-0600	CREATE_COMPLETE	AWS::EC2::Subnet	PublicSubnet1Az2	
08:46:09 UTC-0600	CREATE_COMPLETE	AWS::EC2::VPCGatewayAttachment	AttachGateway	
08:45:59 UTC-0600	CREATE_COMPLETE	AWS::EC2::SecurityGroup	EcsSecurityGroup	

08:45:29 UTC-0600	CREATE_IN_PROGRESS	AWS:CloudFormation::Stack	EC2ContainerService-microcosm	User Initiated																																																																																																				
Template																																																																																																								
Parameters																																																																																																								
<table border="1"> <thead> <tr> <th>Key</th><th>Value</th><th>Resolved Value</th><th></th><th></th></tr> </thead> <tbody> <tr><td>AsgMaxSize</td><td>1</td><td></td><td></td><td></td></tr> <tr><td>CreateElasticLoadBalancer</td><td>false</td><td></td><td></td><td></td></tr> <tr><td>EcsAmiId</td><td>ami-044120f0dd7ed0fb4</td><td></td><td></td><td></td></tr> <tr><td>EcsClusterName</td><td>microcosm</td><td></td><td></td><td></td></tr> <tr><td>EcsEndpoint</td><td></td><td></td><td></td><td></td></tr> <tr><td>EcsInstanceType</td><td>t2.micro</td><td></td><td></td><td></td></tr> <tr><td>EcsPort</td><td>8080</td><td></td><td></td><td></td></tr> <tr><td>ElbHealthCheckTarget</td><td>HTTP:80/</td><td></td><td></td><td></td></tr> <tr><td>ElbPort</td><td>80</td><td></td><td></td><td></td></tr> <tr><td>IamRoleInstanceProfile</td><td>ecsInstanceRole</td><td></td><td></td><td></td></tr> <tr><td>IsFargate</td><td>true</td><td></td><td></td><td></td></tr> <tr><td>KeyName</td><td></td><td></td><td></td><td></td></tr> <tr><td>SourceCidr</td><td>0.0.0.0/0</td><td></td><td></td><td></td></tr> <tr><td>SubnetCidrBlock1</td><td>10.0.0.0/24</td><td></td><td></td><td></td></tr> <tr><td>SubnetCidrBlock2</td><td>10.0.1.0/24</td><td></td><td></td><td></td></tr> <tr><td>TargetGroupName</td><td>ECSFirstRunTargetGroup</td><td></td><td></td><td></td></tr> <tr><td>TargetType</td><td>ip</td><td></td><td></td><td></td></tr> <tr><td>VpcAvailabilityZones</td><td>us-east-2a,us-east-2b,us-east-2c</td><td></td><td></td><td></td></tr> <tr><td>VpcCidrBlock</td><td>10.0.0.0/16</td><td></td><td></td><td></td></tr> </tbody> </table>					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			
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>

Clusters > microcosm

Cluster : microcosm

Status **ACTIVE**

Get a detailed view of the resources on your cluster.

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)

Filter in this page	Launch type	ALL	Service type	ALL				
<input type="checkbox"/> Service Name	Status	ACTIVE	Service type	REPLICA	Task Definition	first-run-task-de...	Desired tasks	1
<input type="checkbox"/> jenkins-service							Running tasks	1
							Launch type	FARGATE
							Platform version	LATEST(1.3.0)

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

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

You are using the following Amazon VPC resources

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

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#vpcs:sort=VpcId>

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 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 micro...	vpc-0530acf15c751393	available	10.0.0.0/16	-	dopt-5f946034	rtb-089b8ba33139a46fa	acl-07738971cf03937c	defi
vpc-9dd0c3f5	vpc-9dd0c3f5	available	172.31.0....	-	dopt-5f946034	rtb-d764bc	acl-597f8132	defi

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#subnets:sort=SubnetId>

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 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 micro...	subnet-0521ea54a64435dd4	available	vpc-0530acf15c751393 ...	10.0.0.0/24	250	-	us-east-2a	use2-az1	rtb-0
ECS micro...	subnet-0e50c4d05ef5052a7	available	vpc-0530acf15c751393 ...	10.0.1.0/24	251	-	us-east-2b	use2-az2	rtb-0
	subnet-0ecabb74	available	vpc-9dd0c3f5	172.31.16.0/20	4091	-	us-east-2b	use2-az2	rtb-d
	subnet-73d1fea1b	available	vpc-9dd0c3f5	172.31.0.0/20	4091	-	us-east-2a	use2-az1	rtb-d
	subnet-f1c1fb5d	available	vpc-9dd0c3f5	172.31.32.0/20	4091	-	us-east-2c	use2-az3	rtb-d

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#RouteTables:sort=routeTableId>

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

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

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

Select an internet gateway above

<https://us-east-2.console.aws.amazon.com/vpc/home?region=us-east-2#SecurityGroups:sort=groupId>

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

https://us-east-2.console.aws.amazon.com/ec2/v2/home?region=us-east-2#Home:

Resources

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

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

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): No events

Availability Zone Status:

- us-east-1a: Availability zone is operating normally
- us-east-1b: Availability zone is operating normally
- us-east-1c: Availability zone is operating normally

[Service Health Dashboard](#)

Scheduled Events

AWS Marketplace

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

- Barracuda CloudGen Firewall for AWS - PAYG
- Matillion ETL for Amazon Redshift

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](#)

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](#)

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

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 8:59:55 AM (0m ago)

Status: ACTIVE INACTIVE	Filter in this page	Latest revision status
<input type="checkbox"/> Task Definition		1-1 > Page size 50
<input type="checkbox"/> first-run-task-definition		ACTIVE

Feedback **English (US)**

© 2008 - 2019, Amazon Web Services, Inc. or its affiliates. All rights reserved. [Privacy Policy](#) [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. You can use a cluster to run tasks on a single Amazon EC2 instance or across multiple regions.

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 fully supported, and the legacy format will be removed in a future release.

[Configure ECS ARN setting](#)

[Create Cluster](#)

[Get Started](#)

View

list

card

[microcosm >](#)

FARGATE

10

Services

4

Running tasks

EC2

0

Services

0

Running tasks

- Scroll down and select your service

[Clusters](#) > microcosm

Cluster : microcosm

Get a detailed view of the resources on your cluster.

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

Services	Tasks	ECS Instances	Metrics	Scheduled Tasks	Tags																								
Create	Update	Delete	Actions ▾																										
<div style="display: flex; justify-content: space-between;"> <input type="text"/> Filter in this page Launch type ALL Service type All </div> <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th><input type="checkbox"/></th> <th>Service Name</th> <th>Status</th> </tr> </thead> <tbody> <tr> <td><input type="checkbox"/></td> <td>sonarqube</td> <td>ACTIVE</td> </tr> <tr> <td><input type="checkbox"/></td> <td>gitlab</td> <td>ACTIVE</td> </tr> <tr> <td><input type="checkbox"/></td> <td>hubot</td> <td>ACTIVE</td> </tr> <tr> <td><input type="checkbox"/></td> <td>owaspzap</td> <td>ACTIVE</td> </tr> <tr> <td><input type="checkbox"/></td> <td>nagios</td> <td>ACTIVE</td> </tr> <tr> <td><input type="checkbox"/></td> <td>sonatype_nexus</td> <td>ACTIVE</td> </tr> <tr> <td><input type="checkbox"/></td> <td>cloud-discovery</td> <td>ACTIVE</td> </tr> </tbody> </table>						<input type="checkbox"/>	Service Name	Status	<input type="checkbox"/>	sonarqube	ACTIVE	<input type="checkbox"/>	gitlab	ACTIVE	<input type="checkbox"/>	hubot	ACTIVE	<input type="checkbox"/>	owaspzap	ACTIVE	<input type="checkbox"/>	nagios	ACTIVE	<input type="checkbox"/>	sonatype_nexus	ACTIVE	<input type="checkbox"/>	cloud-discovery	ACTIVE
<input type="checkbox"/>	Service Name	Status																											
<input type="checkbox"/>	sonarqube	ACTIVE																											
<input type="checkbox"/>	gitlab	ACTIVE																											
<input type="checkbox"/>	hubot	ACTIVE																											
<input type="checkbox"/>	owaspzap	ACTIVE																											
<input type="checkbox"/>	nagios	ACTIVE																											
<input type="checkbox"/>	sonatype_nexus	ACTIVE																											
<input type="checkbox"/>	cloud-discovery	ACTIVE																											

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

[Clusters](#) > [microcosm](#) > Service: cloud-discovery

Service : cloud-discovery

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

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

Number of tasks i

Minimum healthy percent i

Maximum percent i

Cancel **Skip to review**

- Press Update service and the View Service

Cancel **Previous** **Update Service**

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

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

Update **Delete**

- Press Delete in the top right corner
- Check the Delete Discovery box and enter delete me in the prompt and press Delete

Delete Service x

Are you sure you want to delete the service cloud-discovery? If the service is not scaled down to 0, it will be scaled down before it is deleted.

Delete the service discovery service "cloud-discovery" created with this service.

You can delete the service discovery namespace associated with this service using the [AWS Cloud Map console](#).

Enter the phrase "delete me" into the field below to confirm deletion.

Cancel **Delete**

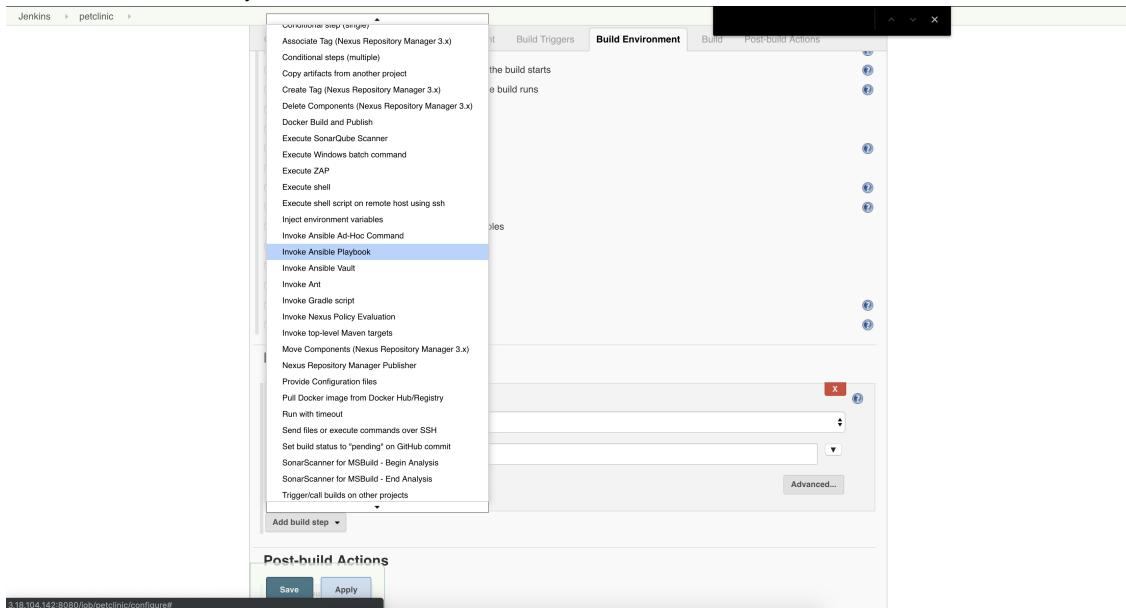
Notes:

- 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

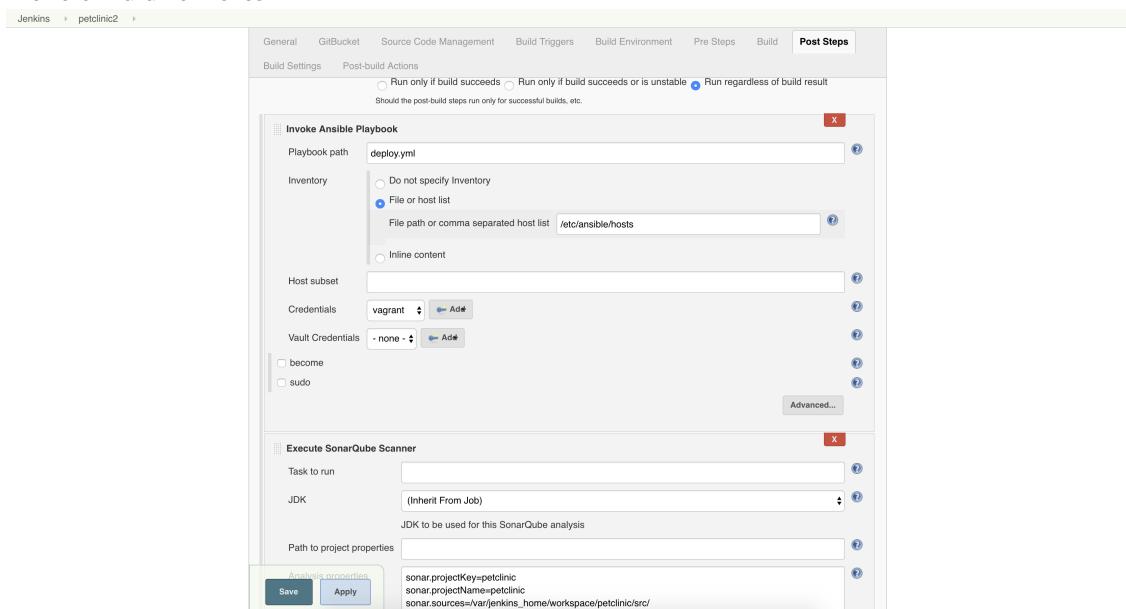
Ansible Deployment

To integrate deployment of the test project using Ansible do the following:

- On the project page, select Configure
 - Scroll down to the Post Steps section and click Add build step
 - Select Invoke Ansible Playbook



- Playbook path: deploy.yml
 - Inventory: File or host list: /etc/ansible/hosts
 - Beside Credentials, click Add -> Jenkins
 - Select “SSH Username with private key”
 - Username: vagrant
 - Private Key: “From a file on Jenkins master”: /etc/ansible/vagrant_id_rsa (If this option is not available, you may need to enter the private key directly, or leave it blank to use the default)
 - Credentials: select ‘vagrant’
 - Click Apply and then click Save
 - Then click Build Now to test



that makes this work with AWS

```
[INFO] -----  
[INFO] BUILD SUCCESS  
[INFO] -----  
[INFO] Total time: 27.562 s  
[INFO] Finished at: 2019-04-17T17:14:37Z  
[INFO] -----  
Waiting for Jenkins to finish collecting data  
[JENKINS] Archiving /var/jenkins_home/workspace/petcclinic2/pom.xml to org.springframework.samples/spring-petcclinic/1.0.0-SNAPSHOT/spring-petcclinic-1.0.0-SNAPSHOT.pom  
[JENKINS] Archiving /var/jenkins_home/workspace/petcclinic2/target/petcclinic.war to org.springframework.samples/spring-petcclinic/1.0.0-SNAPSHOT/spring-petcclinic-1.0.0-SNAPSHOT.war  
channel stopped  
[petclinic2] $ ansible-playbook deploy.yml -i /etc/ansible/hosts -f 5 --private-key /tmp/ssh7186750522361202959.key -u vagrant  
[WARNING]: Unable to parse /etc/ansible/hosts as an inventory source  
[WARNING]: No inventory was parsed, only implicit localhost is available  
[WARNING]: provided hosts list is empty, only localhost is available. Note  
that the implicit localhost does not match 'all'  
[WARNING]: Could not match supplied host pattern, ignoring: DevOps  
  
PLAY [DevOps] *****  
skipping: no hosts matched  
  
PLAY RECAP *****  
Finished: SUCCESS
```

[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.

TODOTODOTODOTODOTODO - Screenshots

[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 takes 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](#)