Microcosm on AWS

Table of Contents

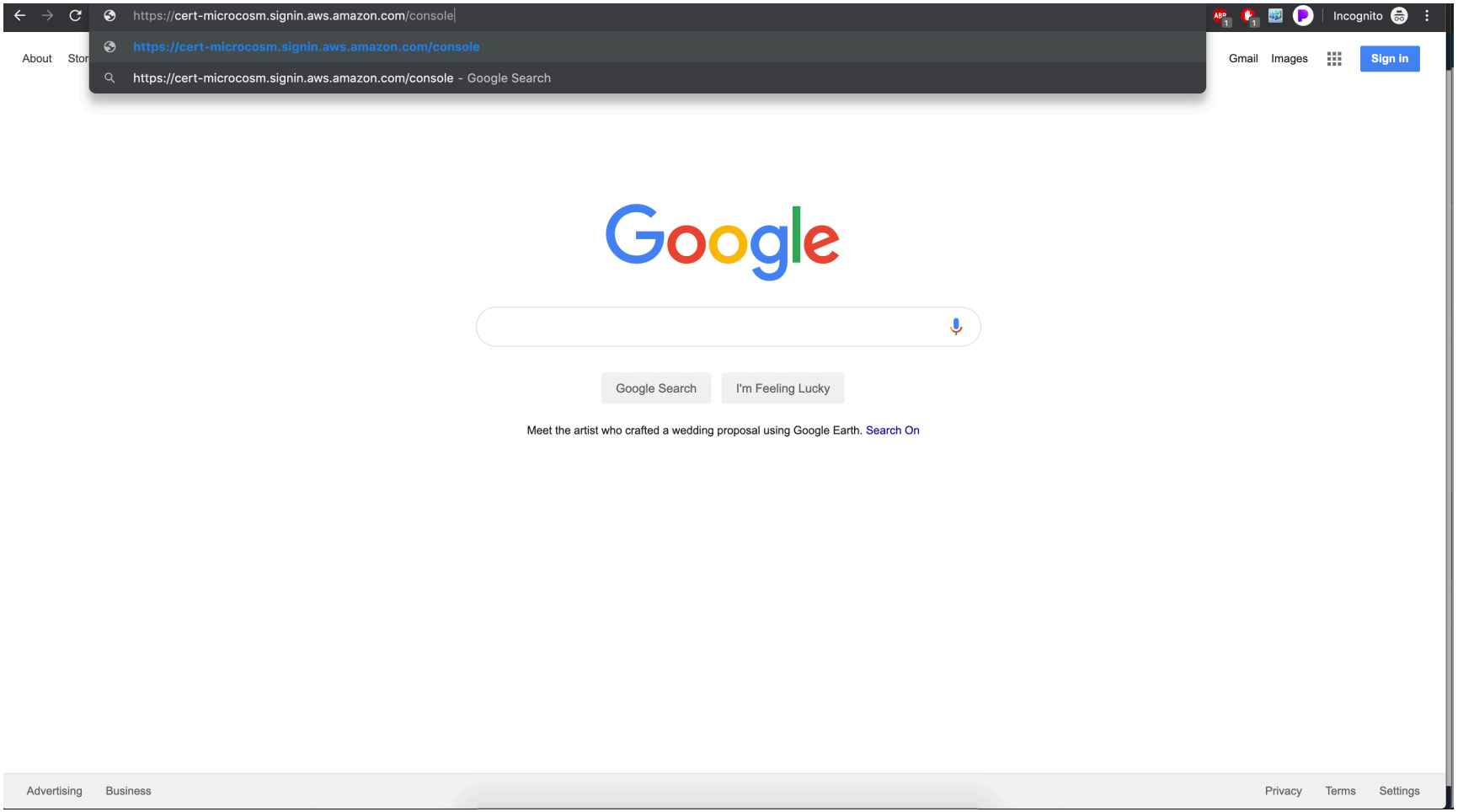
Introduction

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

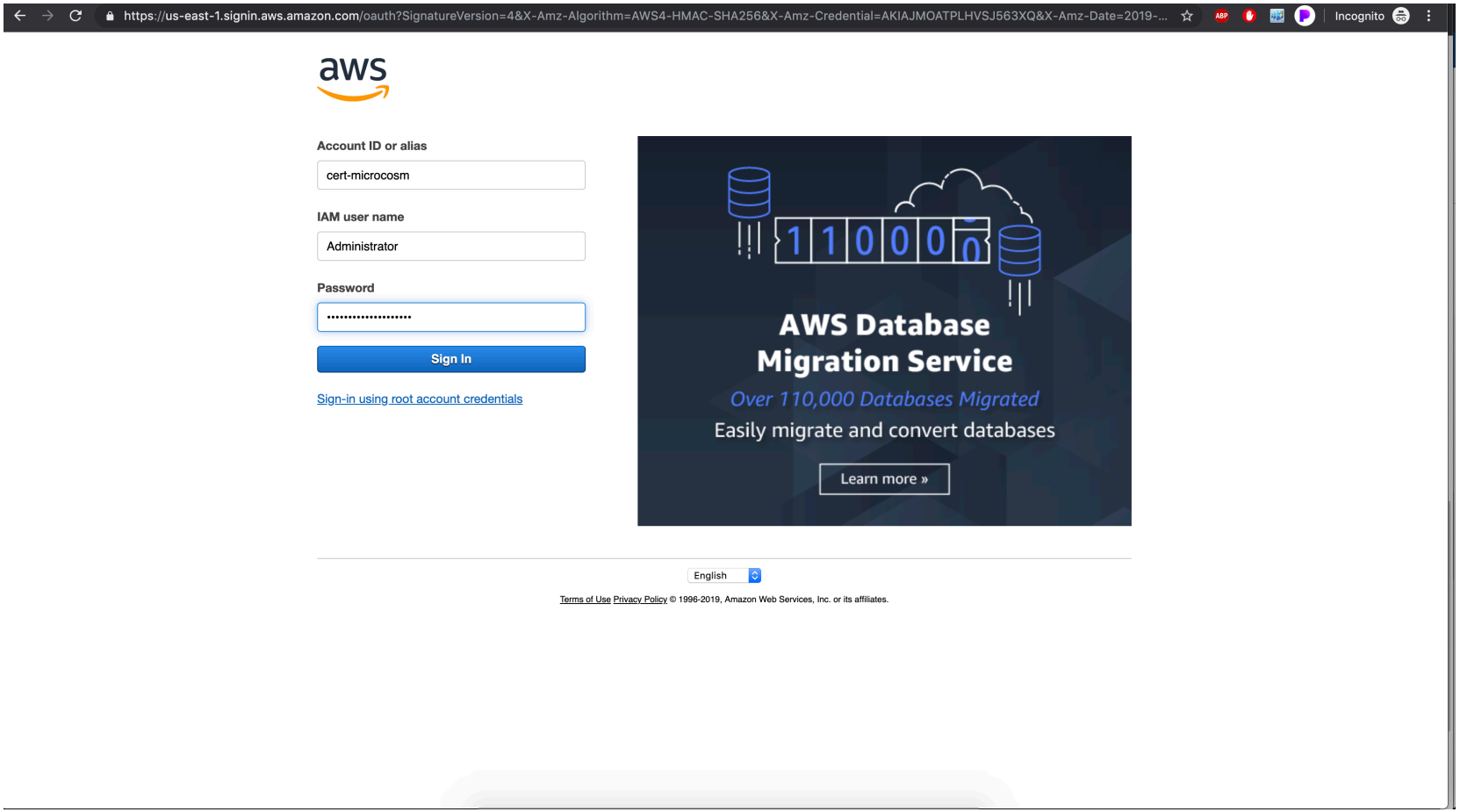
Return to Table of Contents

Login

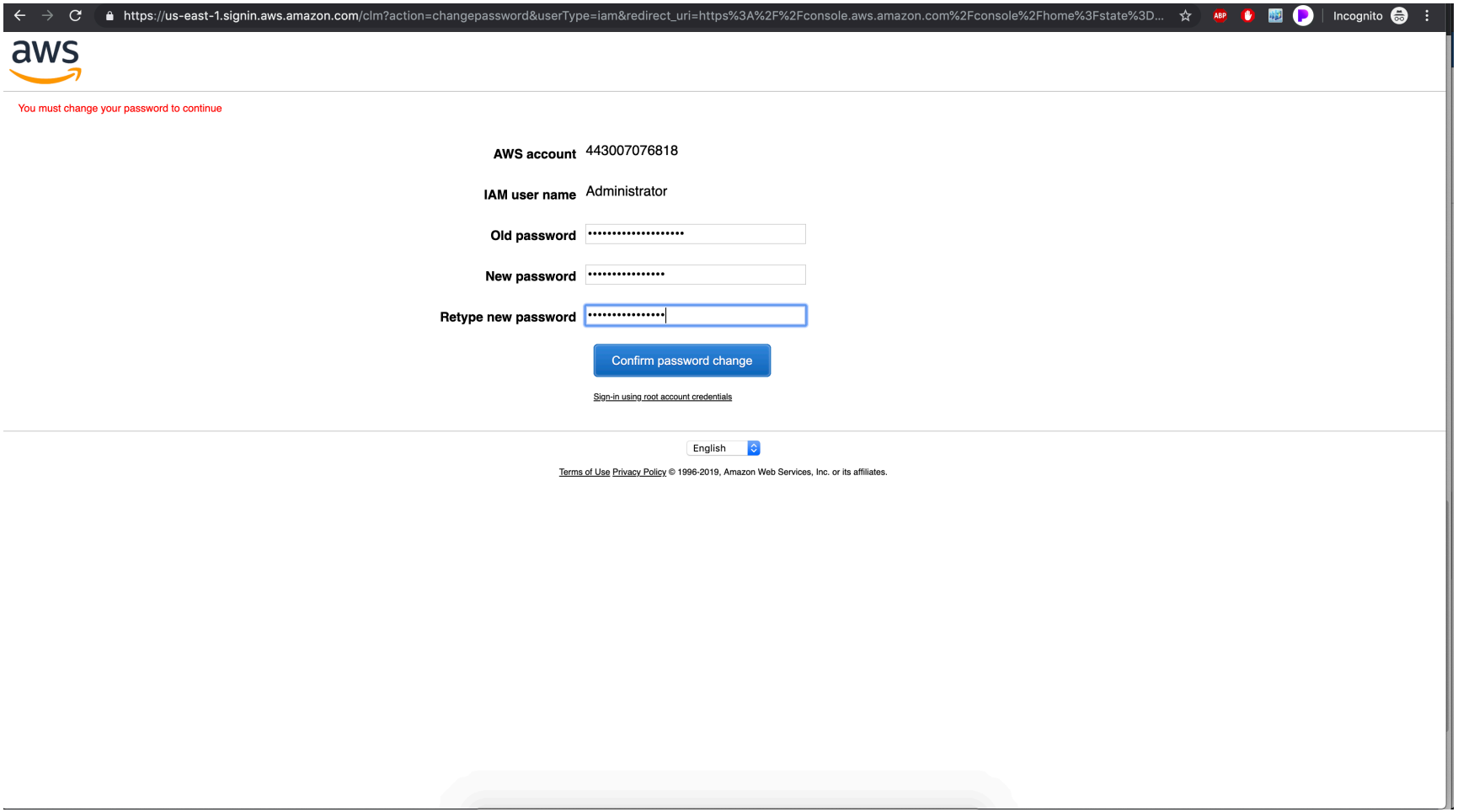
- Navigate to the URL received (eg https://cert-microcosm.signin.aws.amazon.com/console)



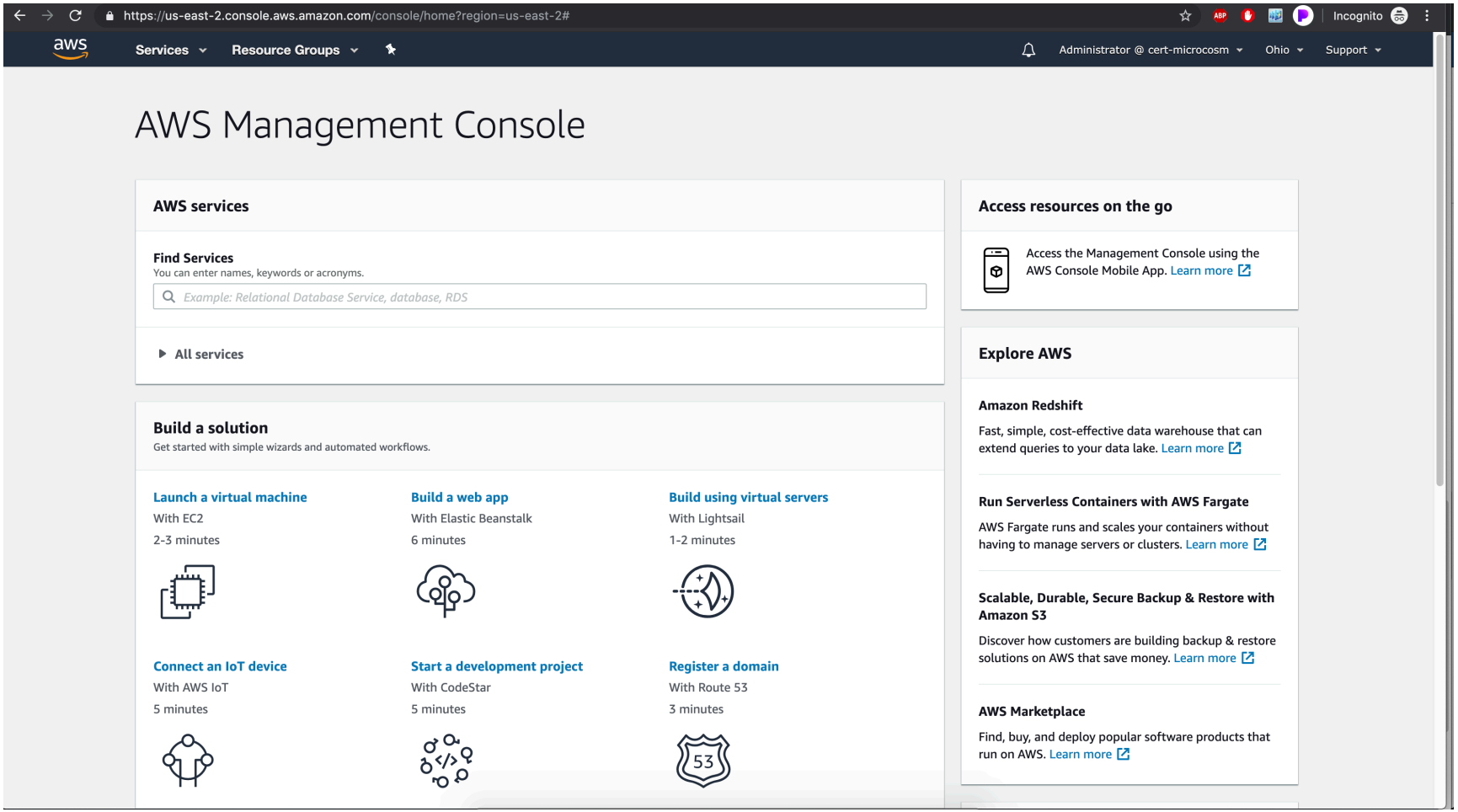
- Enter the credentials received
  - User:[stackname]-StudentUser[number]-[hash]
  - pwd: bzqUrFLxw9HFtB-49eRtf!



- Change your password to something you will remember



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



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

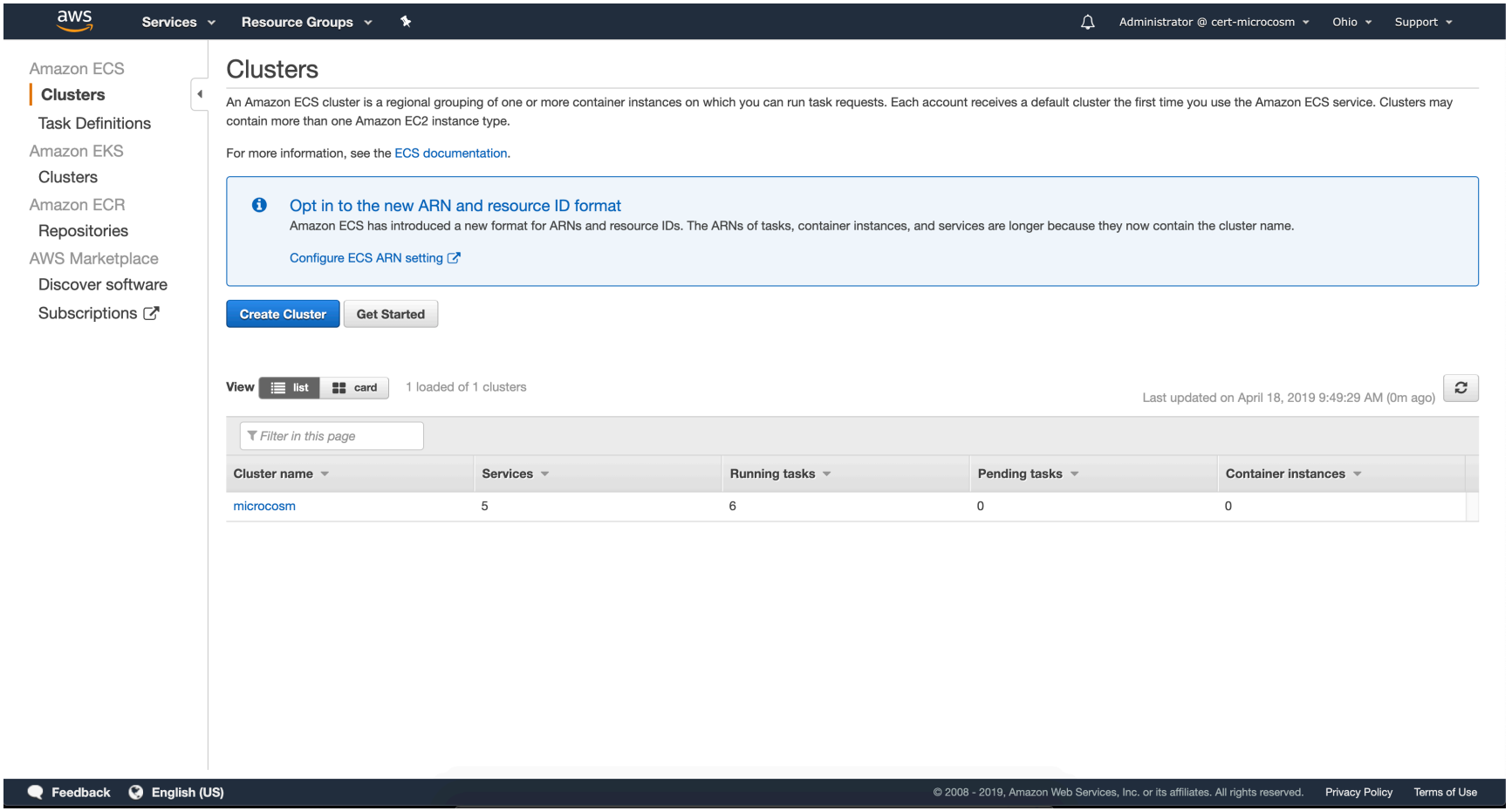
are instantiated as Services - which are grouping of instances of the task definitions. Once a service is started and the related task launches successfully, ECS manages the load balancing (if selected), failure recovery, addressing and other features. For information on the quick start wizard, see Appendix B.

Clusters

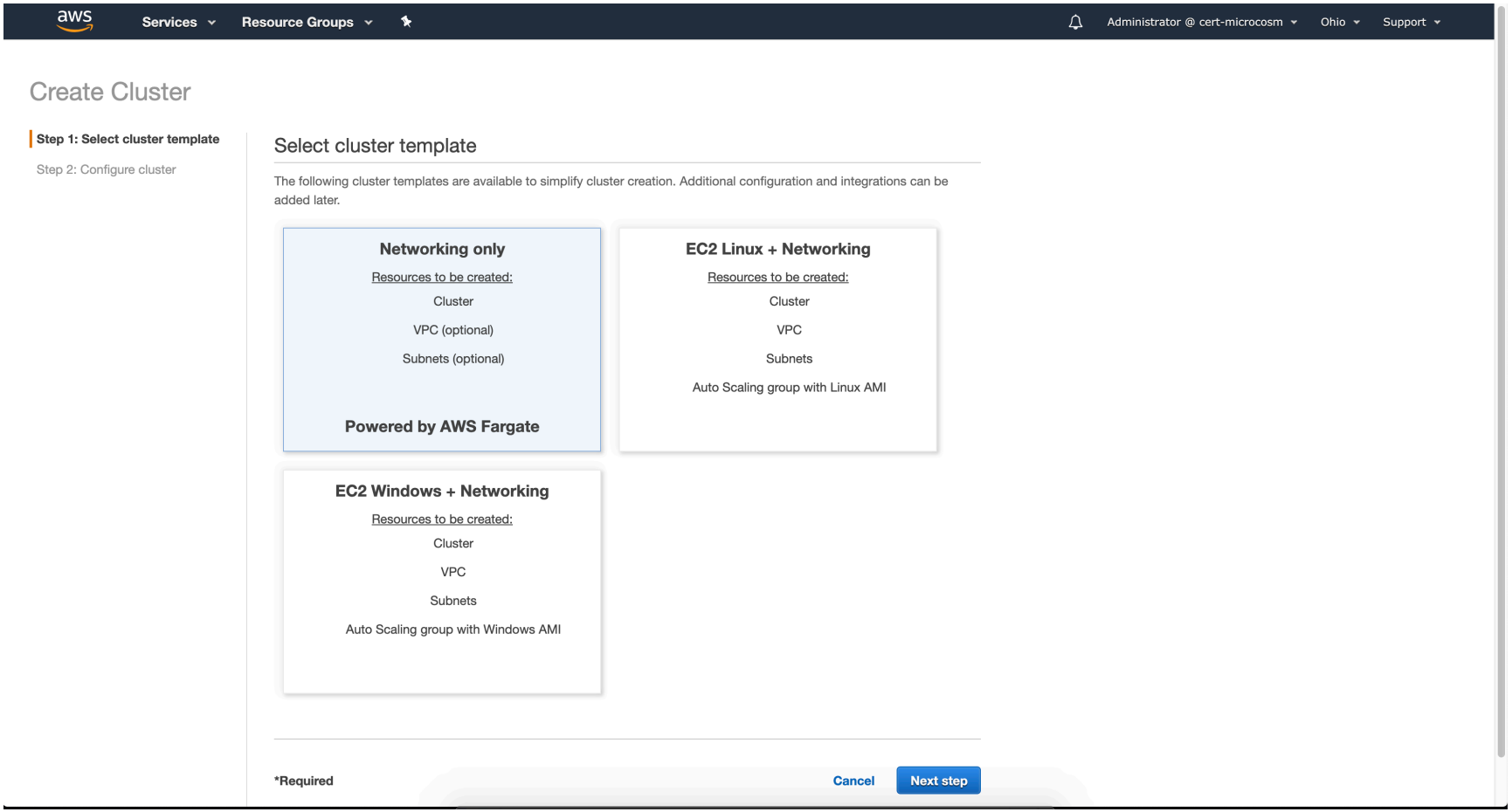
Clusters are a grouping construct within ECS that enable Docker images to be instantiated as containers. The default cluster is created using the get started wizard and will create a VPC, Security groups, networking policies and more behind the scenes - enabling access to and for the container instances (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

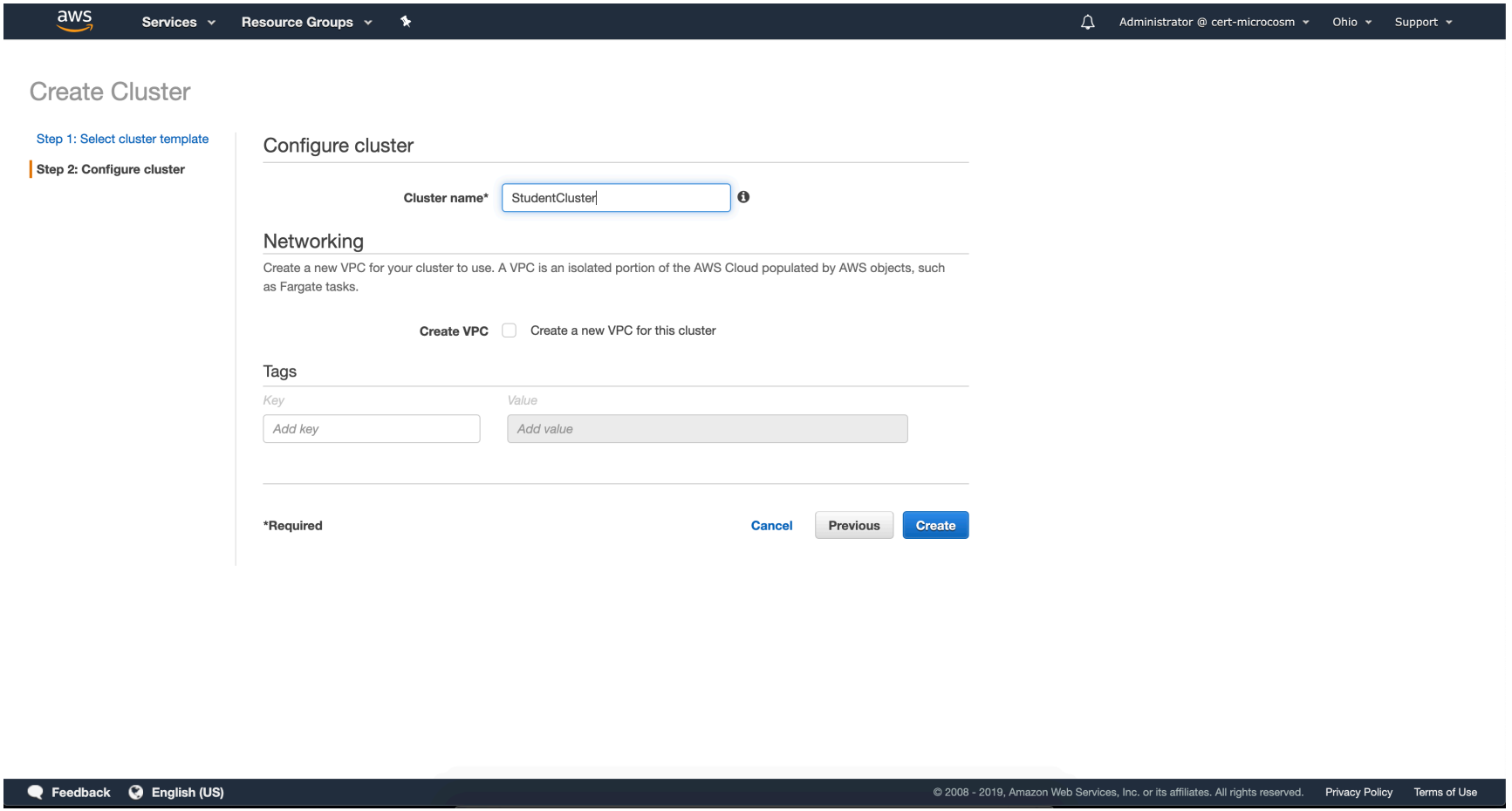
- Select ECS from the default AWS Console or the drop down services menu
- Open the cluster console and select create cluster



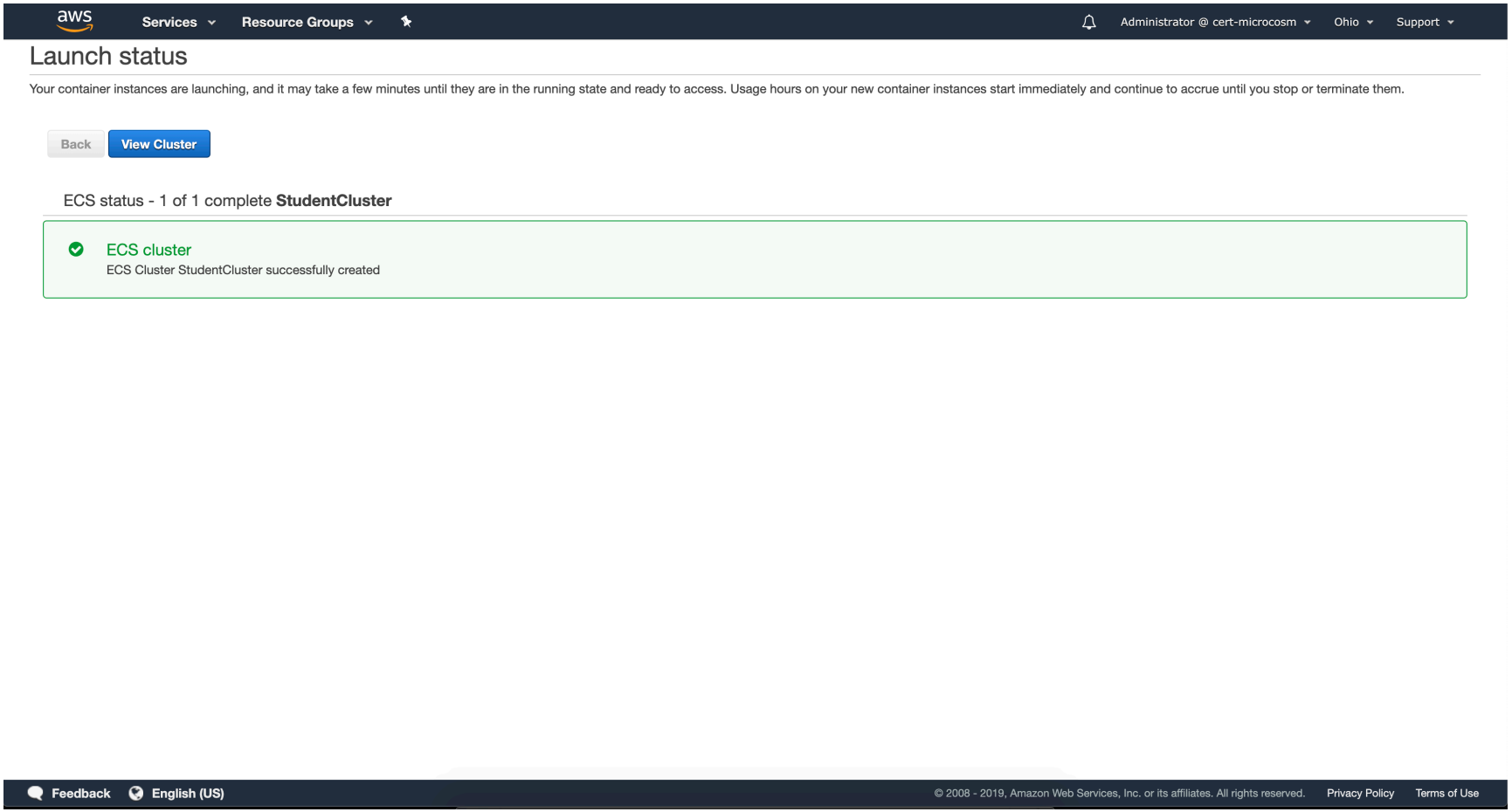
- We are going to be using AWS Fargate, so leave the default selection and click Next Step



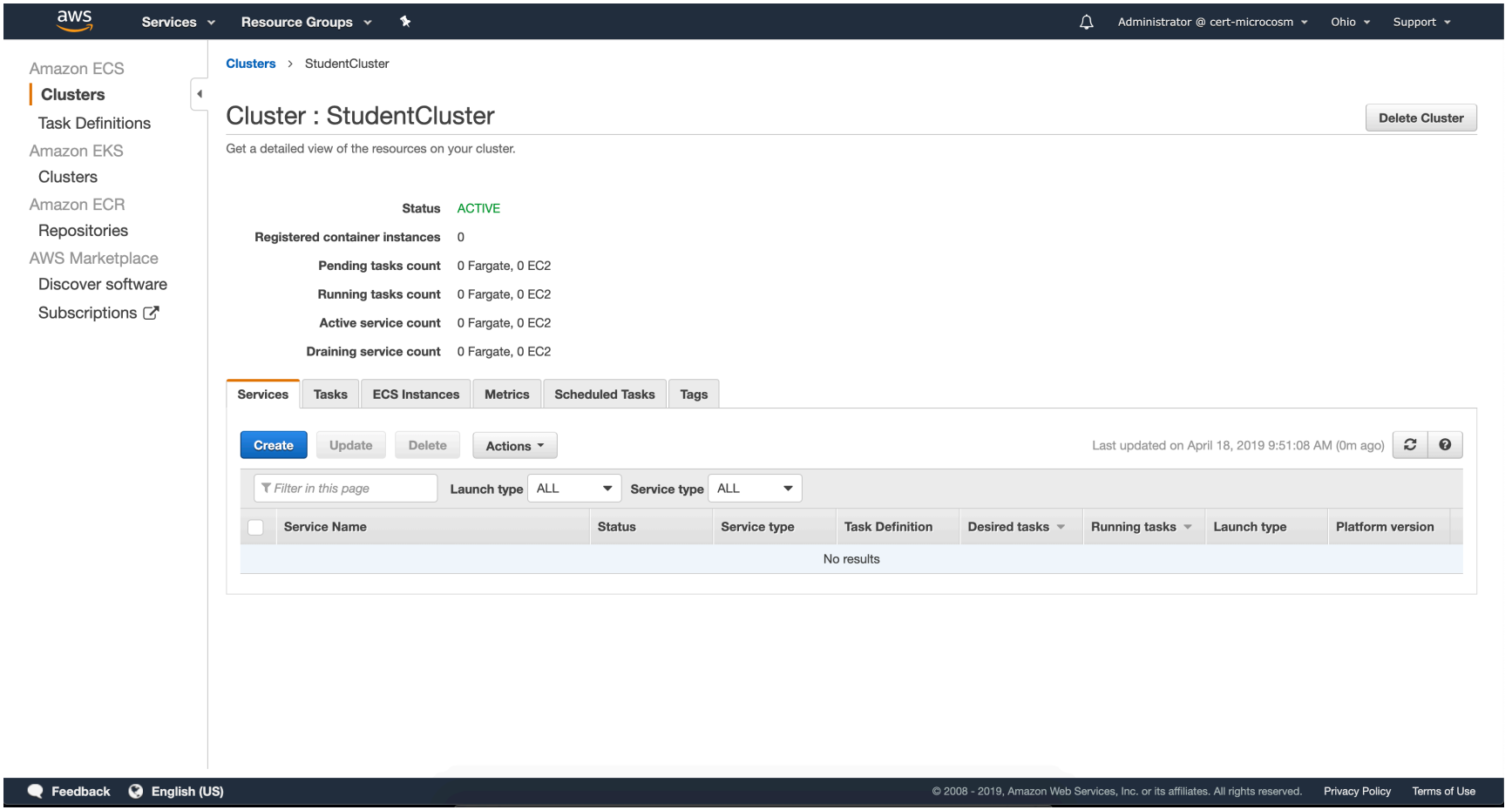
- Enter a Cluster Name and select Create. We already have a VPC in place, so there is no need to check the Create VPC box. If you do not have a VPC in place yet on your own account, select create VPC.



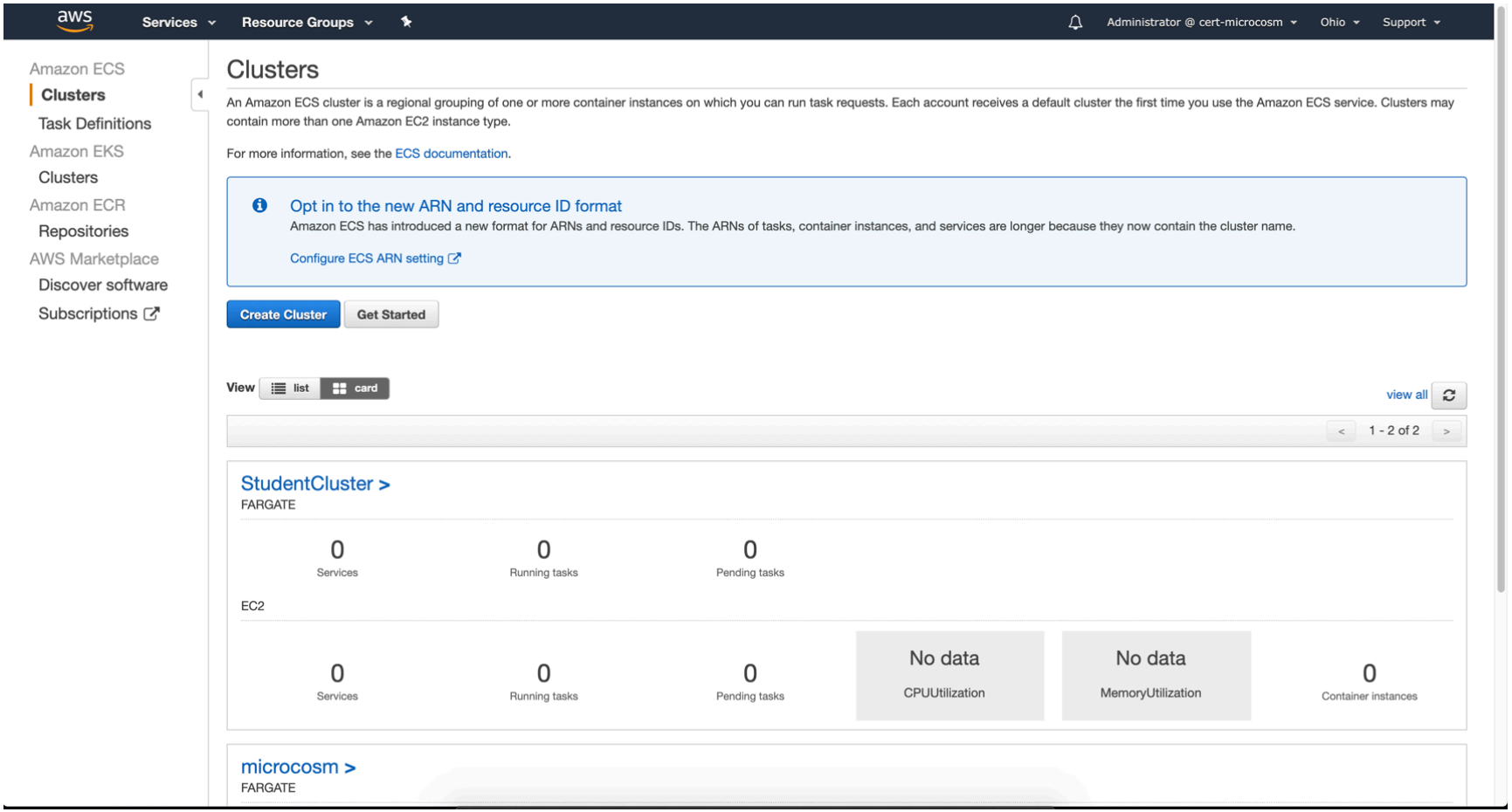
- Once the cluster is created successfully select View Cluster



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



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



Task Definitions

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

Microcosm Task Definitions

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

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

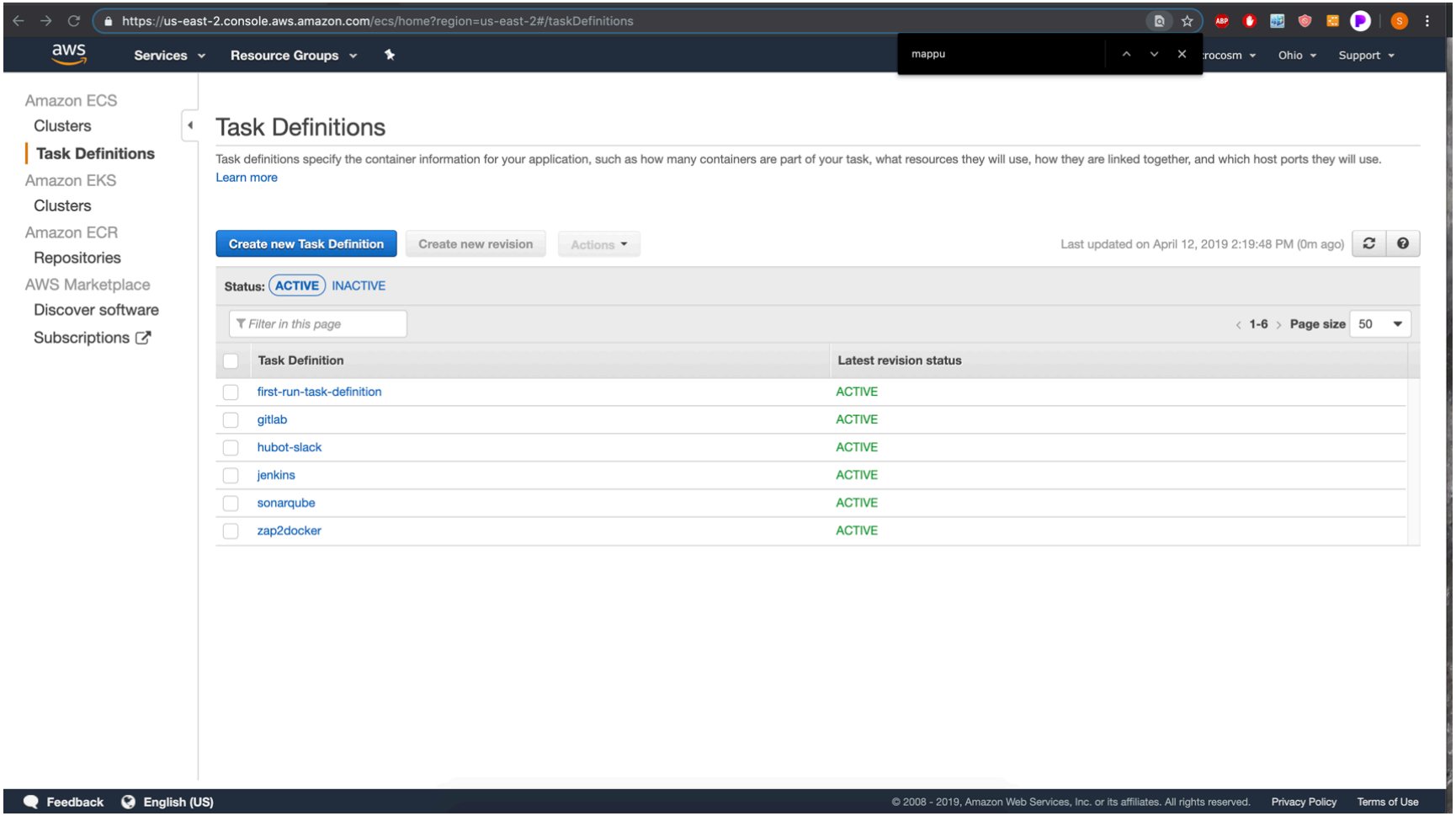
Task/Service Name	Container Name	Image	Ports	Volumes	Mount Poi
jenkins	jenkins	h1kkan/jenkins-docker:lts	8080	jenkins_home	/var/jenkins_home
sonarqube	sonarqube	sonarqube:lts	9000	sonarqube_conf	/opt/sonarqube/c
				sonarqube_data	/opt/sonarqube/d
				sonarqube_extensions	/opt/sonarqube/e
				sonarqube_bundled-plugins	/opt/sonarqube/lib plugins
gitlab	gitlab	gitlab/gitlab-ce	443	gitlab-config	/etc/gitlab
			80	gitlab-logs	/var/log/gitlab
				gitlab-data	/var/opt/gitlab
owaspZAP	owaspzap	owasp/zap2docker-stable	8080		
			8090		
nexus	sonatype_nexus	sonatype/nexus	8081	nexus-data	/sonatype-work

Return to Table of Contents

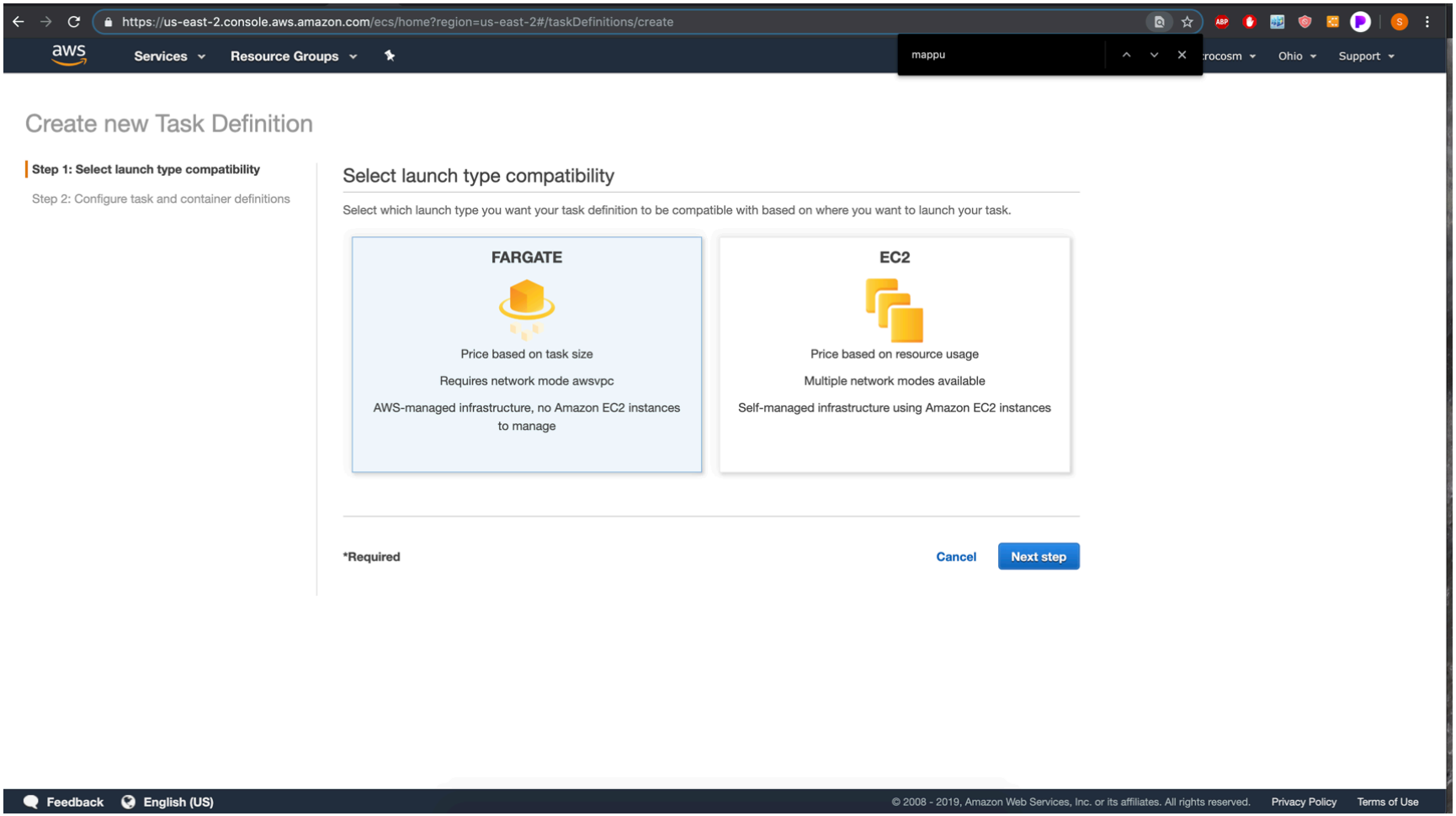
Define a task

Repeat the Following for each desired task definition

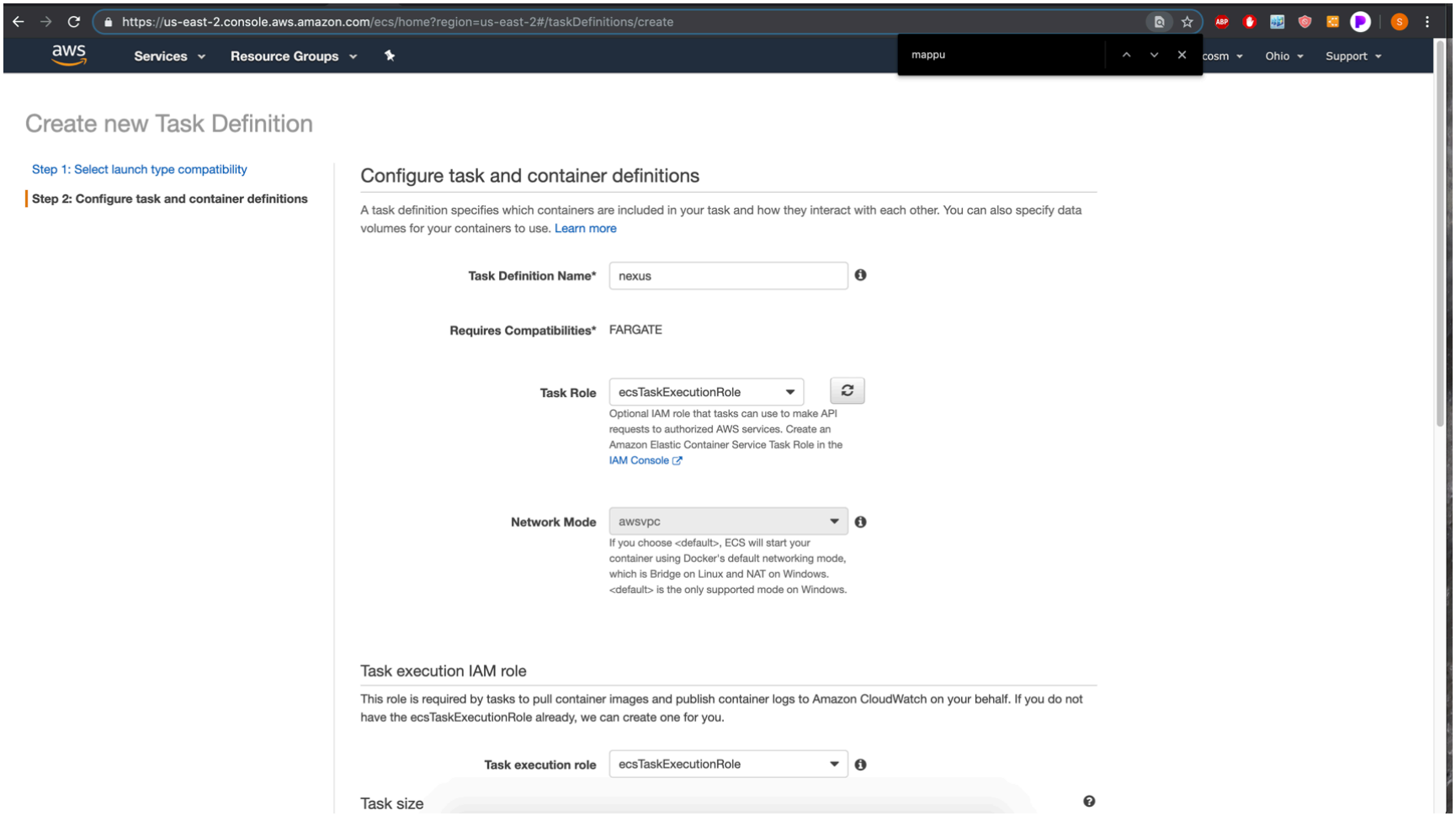
- To define a new task (aka docker container) navigate to the ECS Console and select Task Definitons from the left menu, and press Create new Task Defniniton



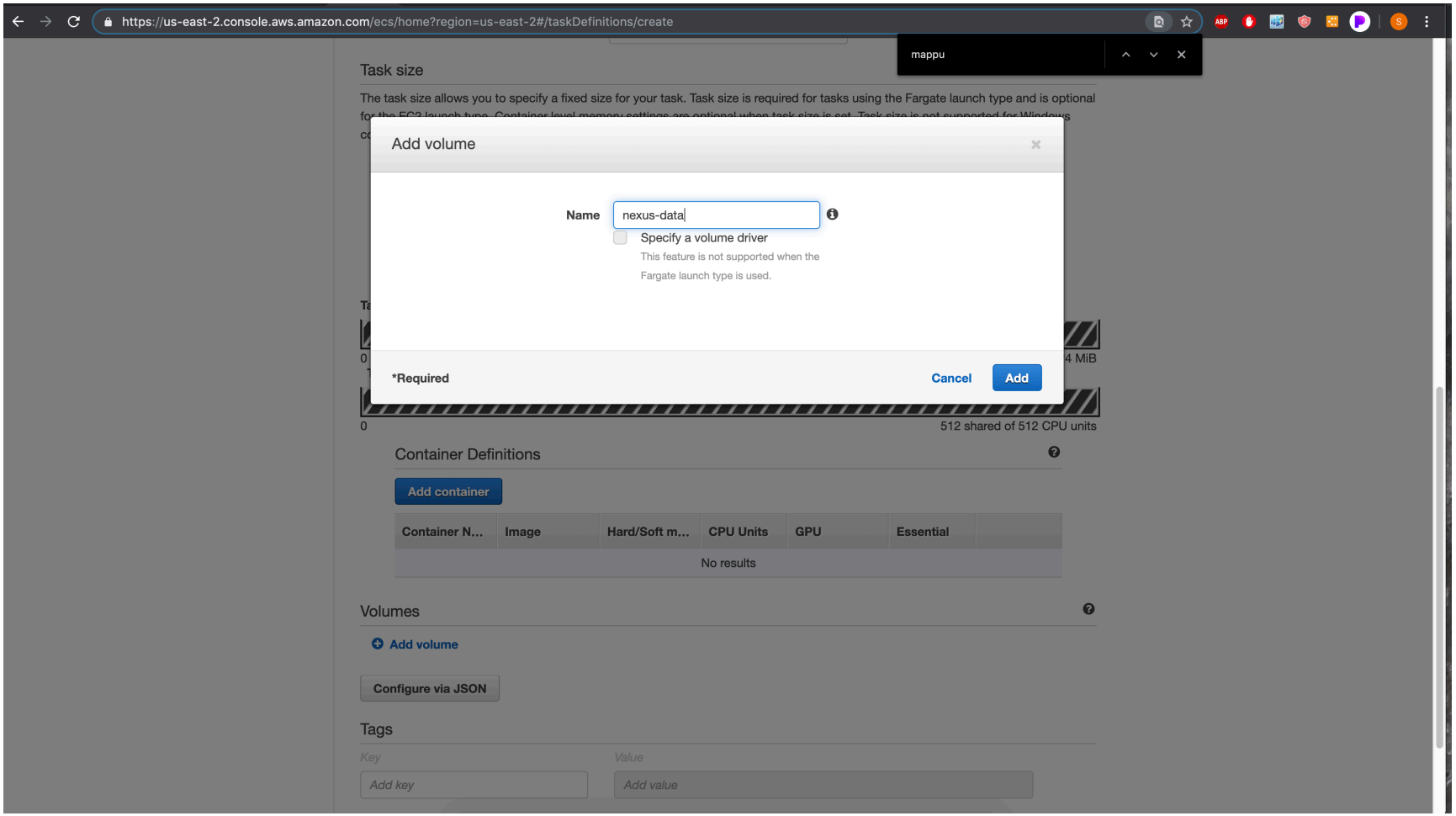
- Select the Fargate Type and press Next Step



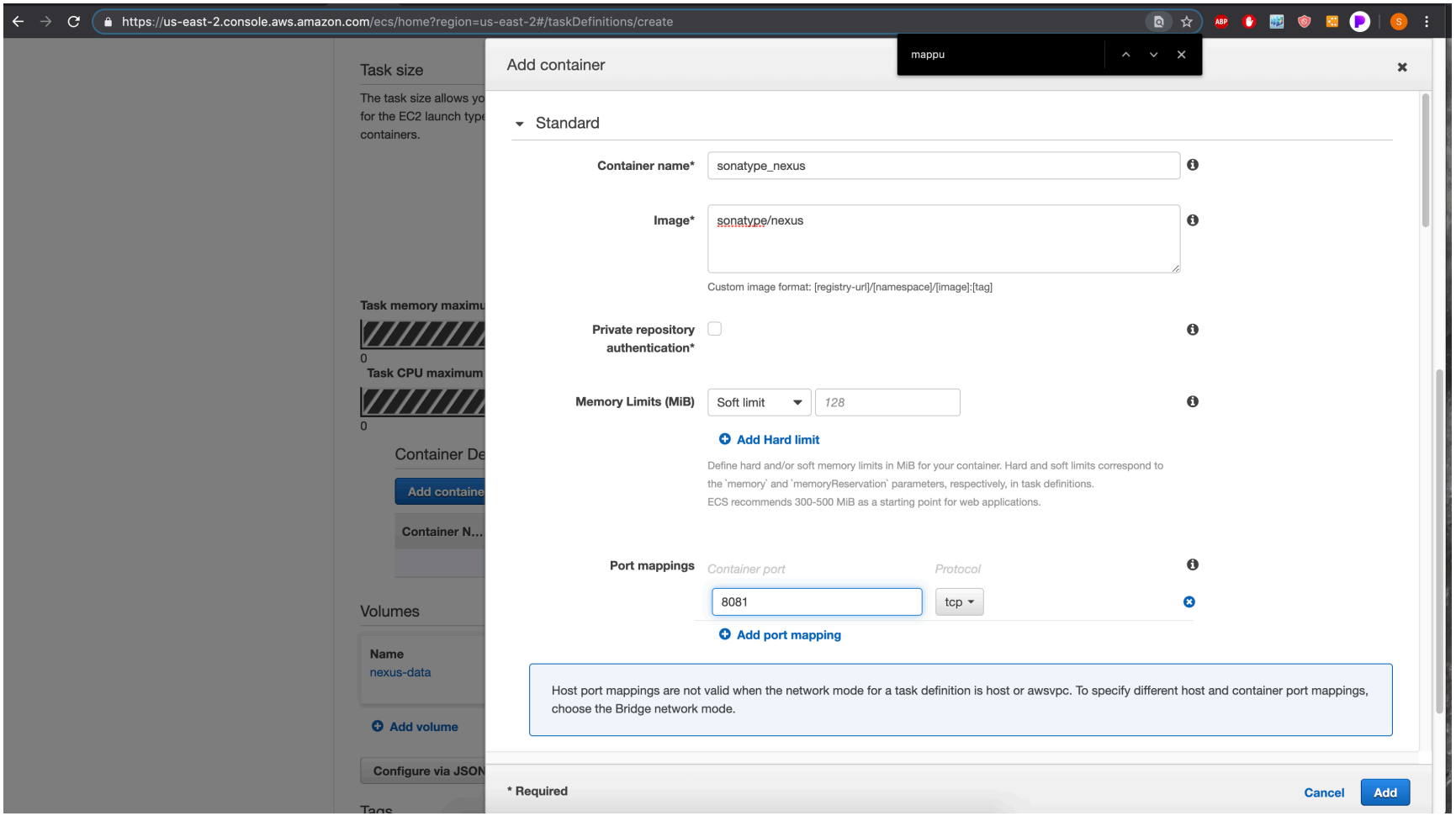
- Enter the task name, and select a Task Role, a Task Execution Role, and sizing parameters



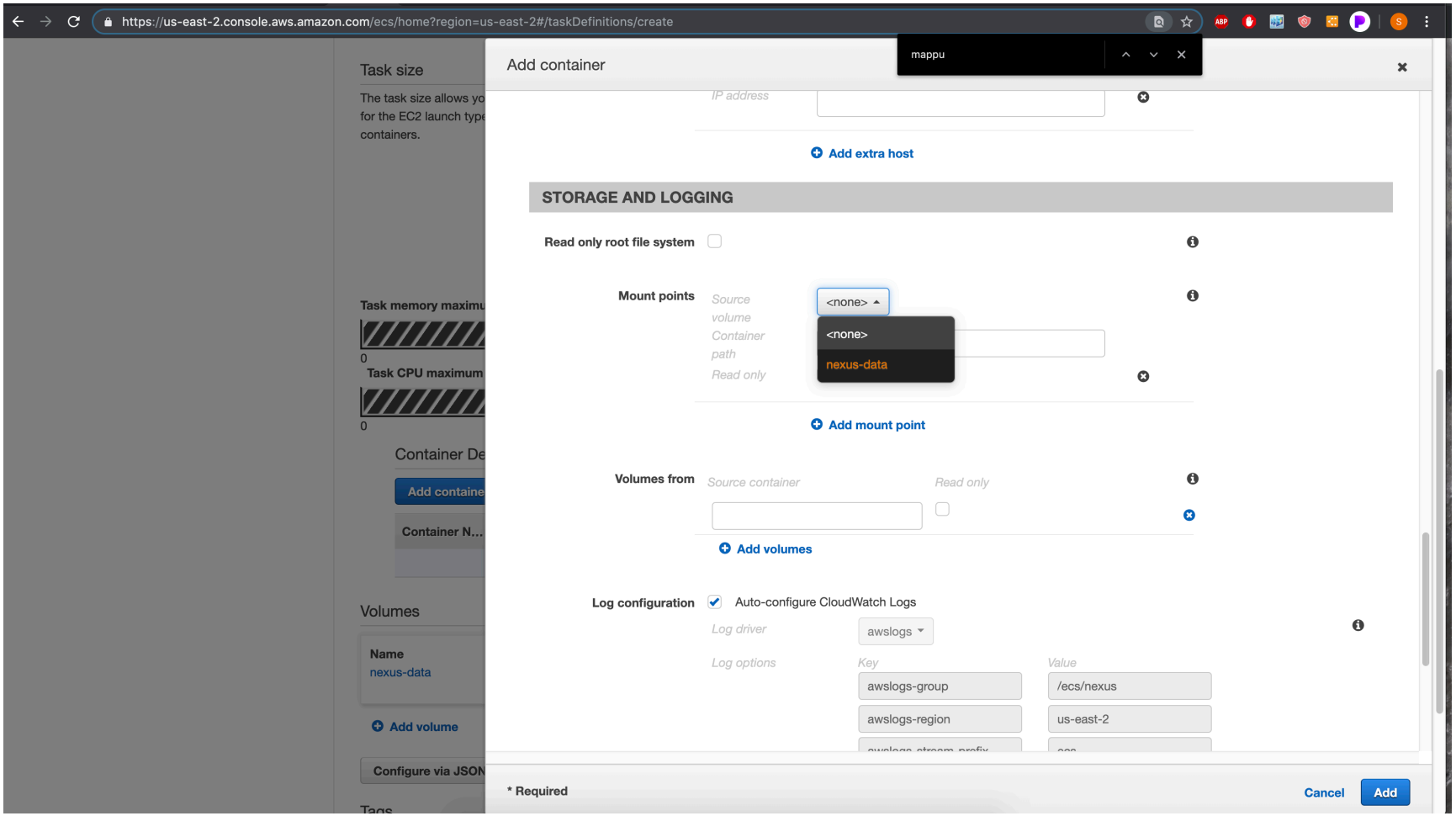
- Select Add Volume, enter the Volume Name and and press add



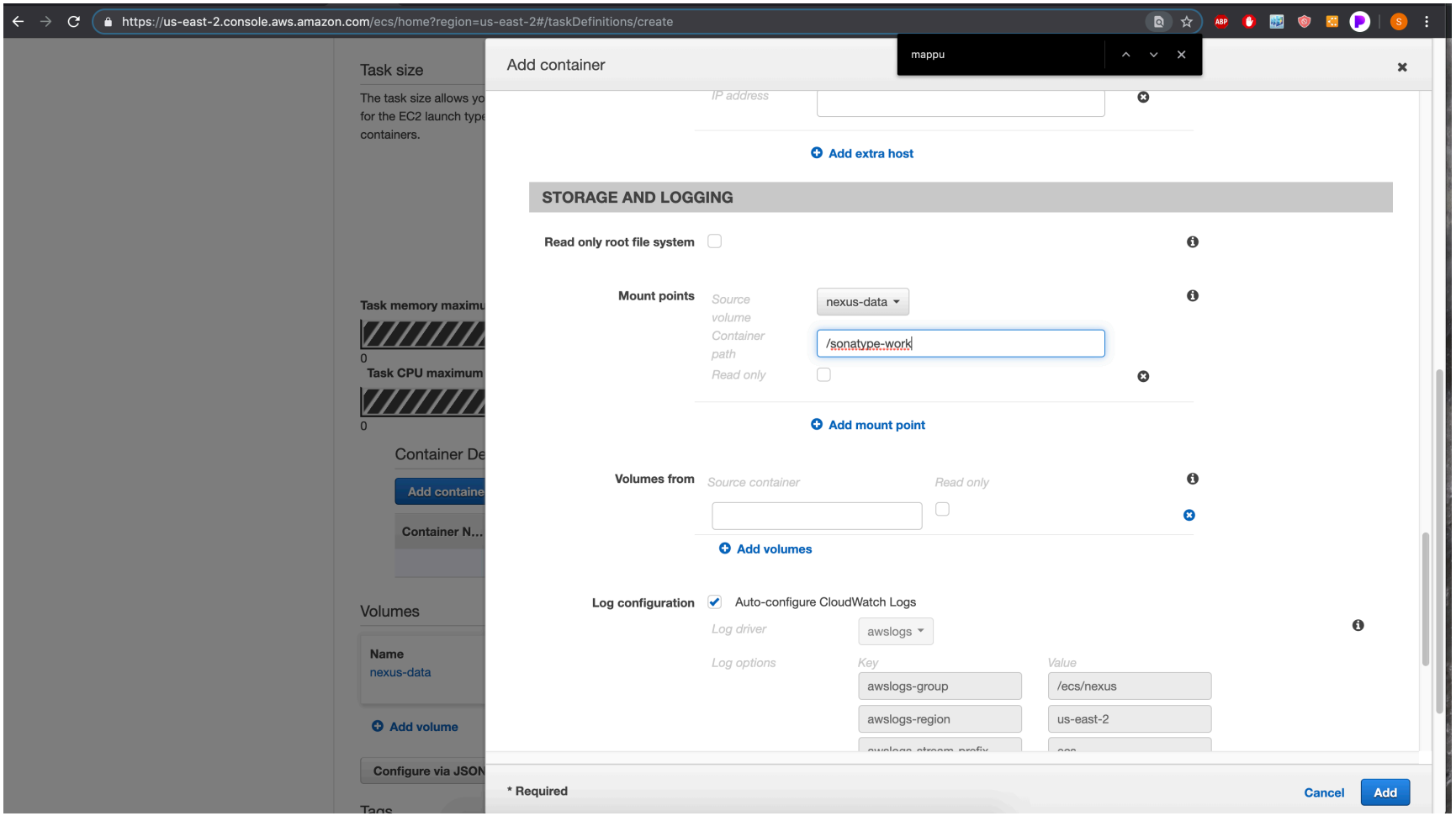
- Select Add Container.
- Enter the Container Name, Image, Port mappings number(s) (select Add port mapping to add additional ports)



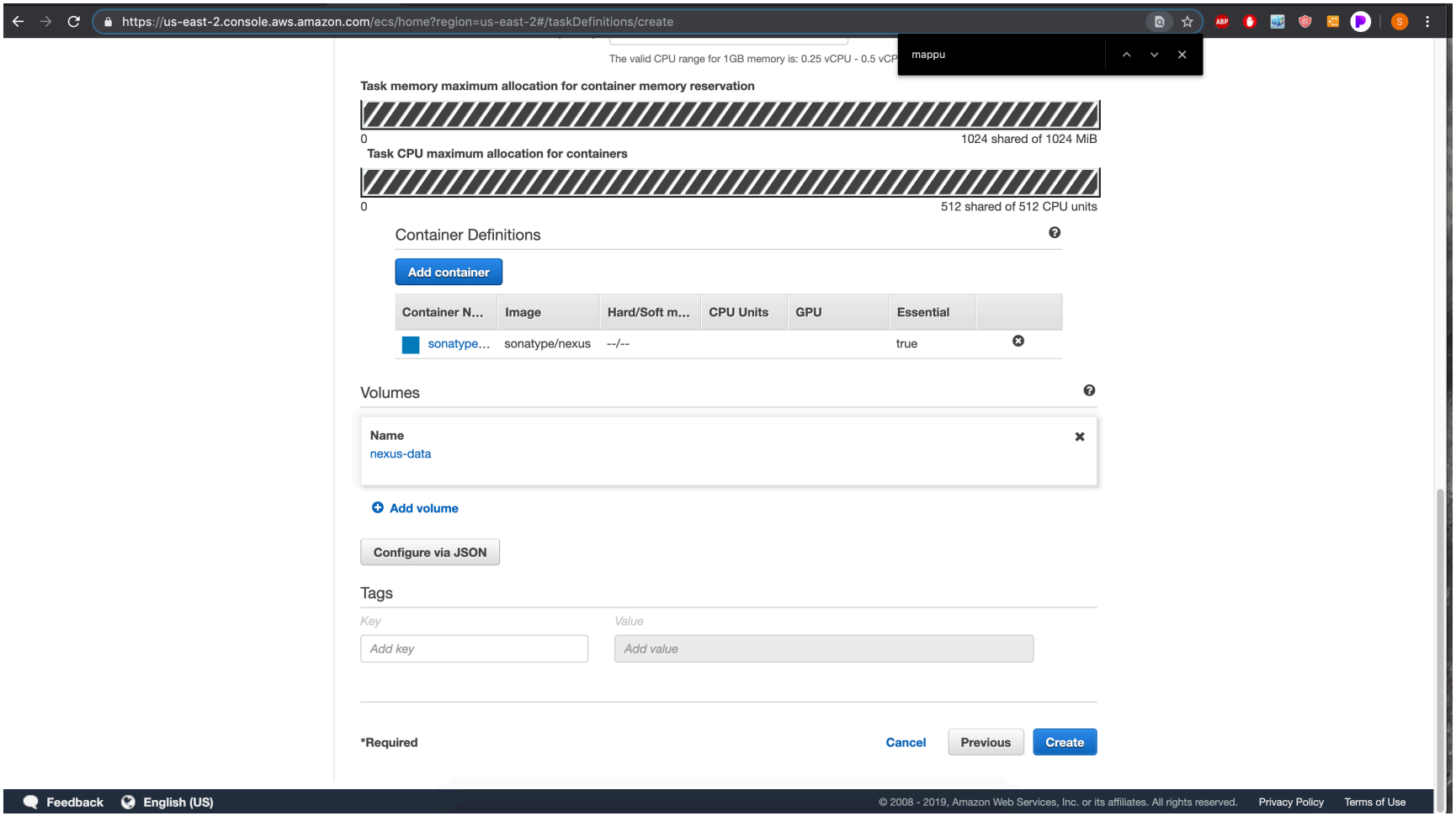
- Under Mount Points, if applicable, select a volume entered previously from the drop down



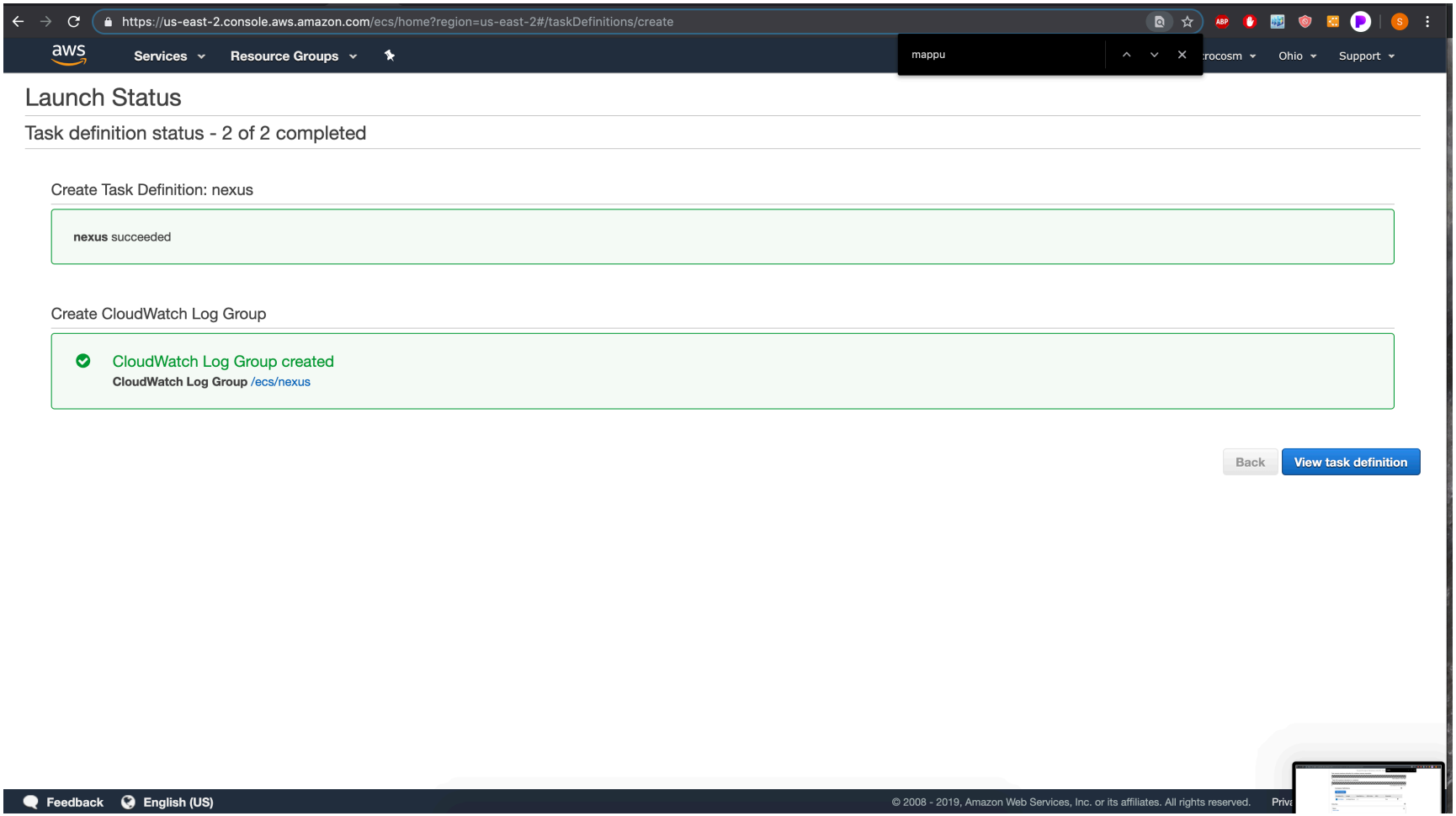
- Enter the mount point (s) (Select Add mount point if additional are required).
- Make sure Auto-configure CloudWatch Logs is selected.
- Press Add.



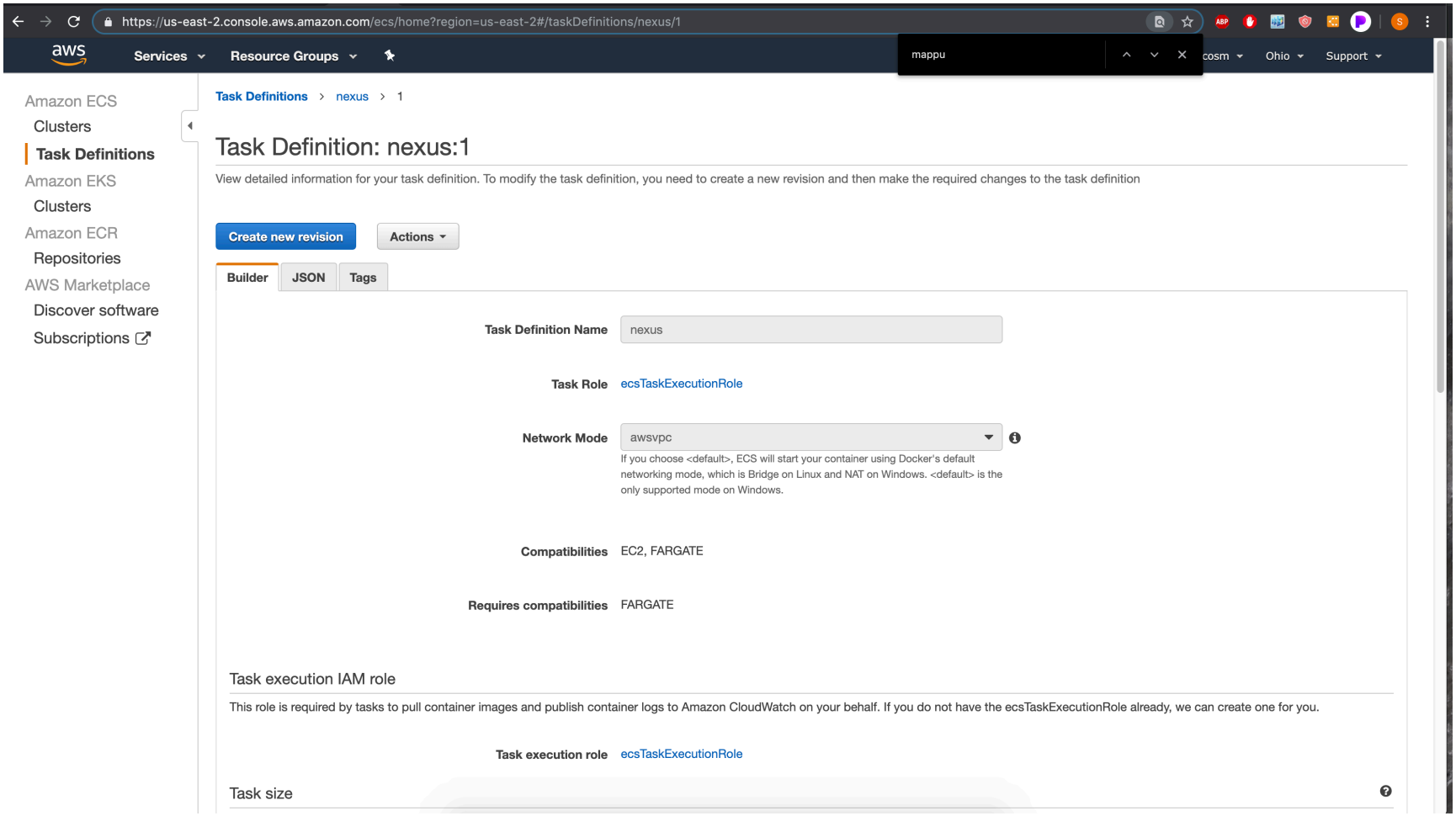
- Add Tags if desired and press create



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



- Review the task definition, if desired



Return to Table of Contents

Services

Services are constructs within ECS that allow for the management, grouping, load balancing, fault tolerance, etc. of tasks. When a service is created based on a task definition, ECS will instantiate tasks (containers) automatically and

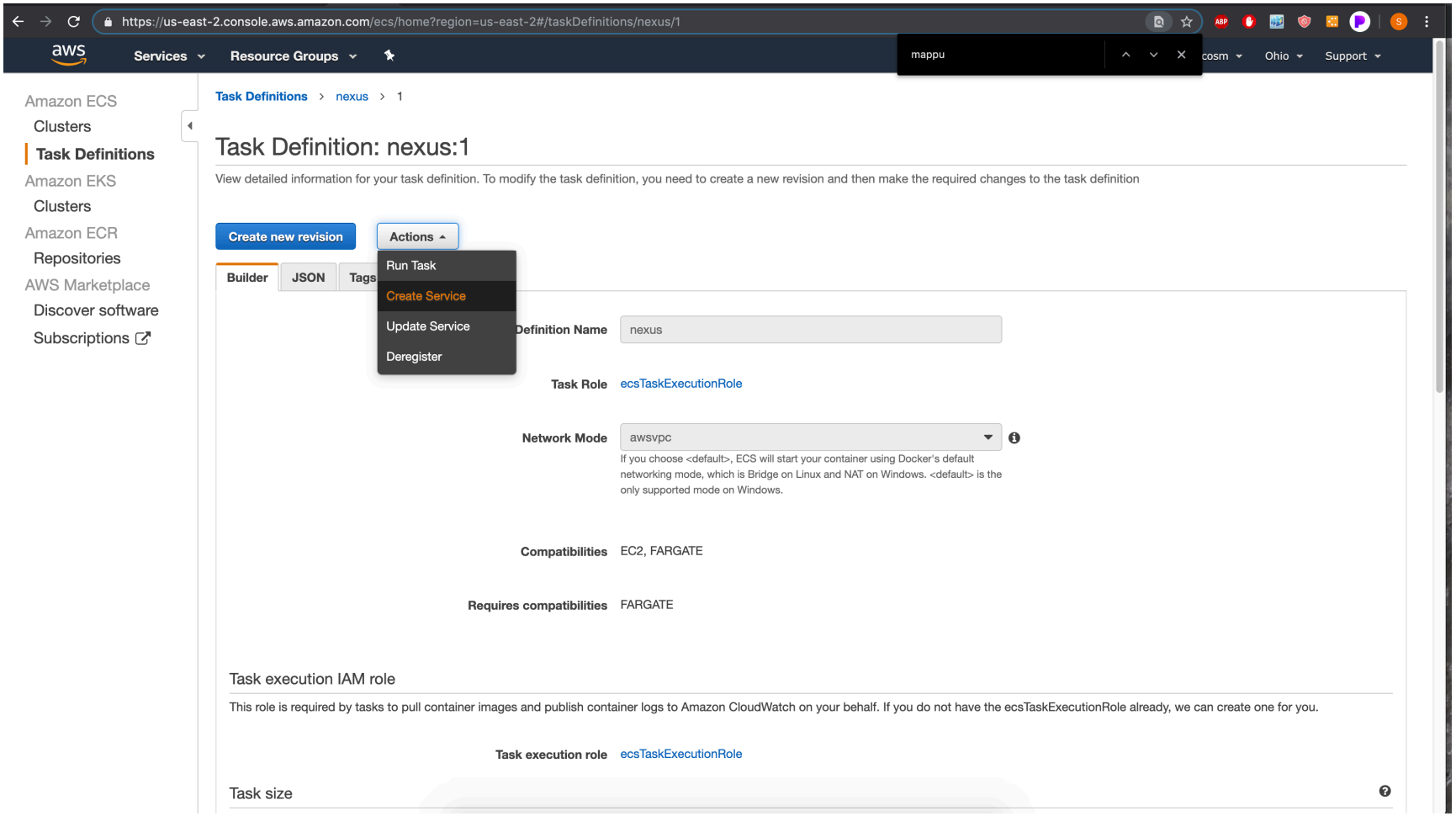
re-stand them up if they fail. It is possible to have multiple different task definitons, of the same or different types, within a single service. Note, it is possible to have tasks not managed by a service, these have no fault tolerance nor take advantage of any other service properties. In this example 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 the set the number of tasks to 0, else the service will restart the tasks before the service can be deleted.

####Create a Service

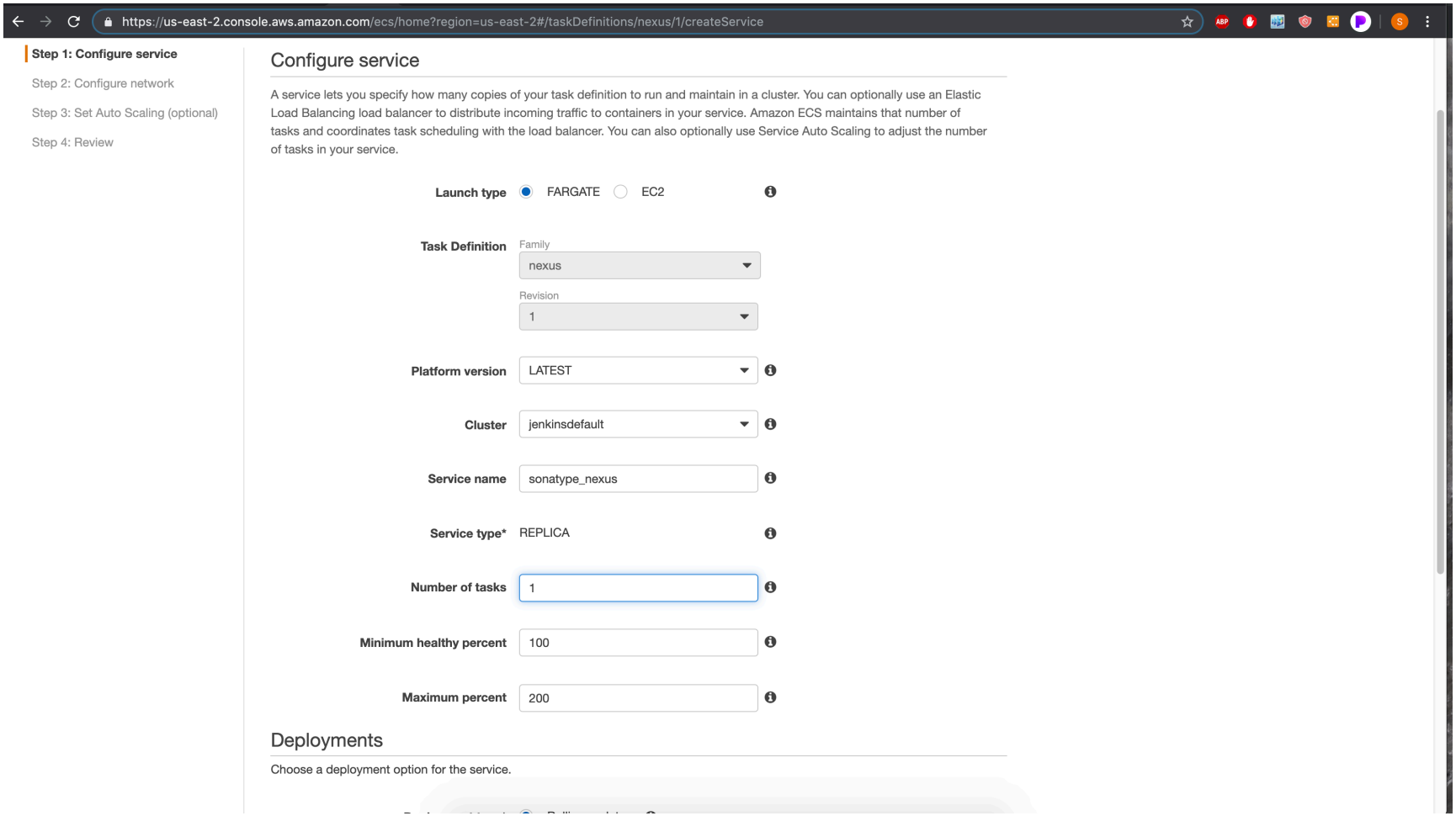
Repeat the Following for each desired service definition

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

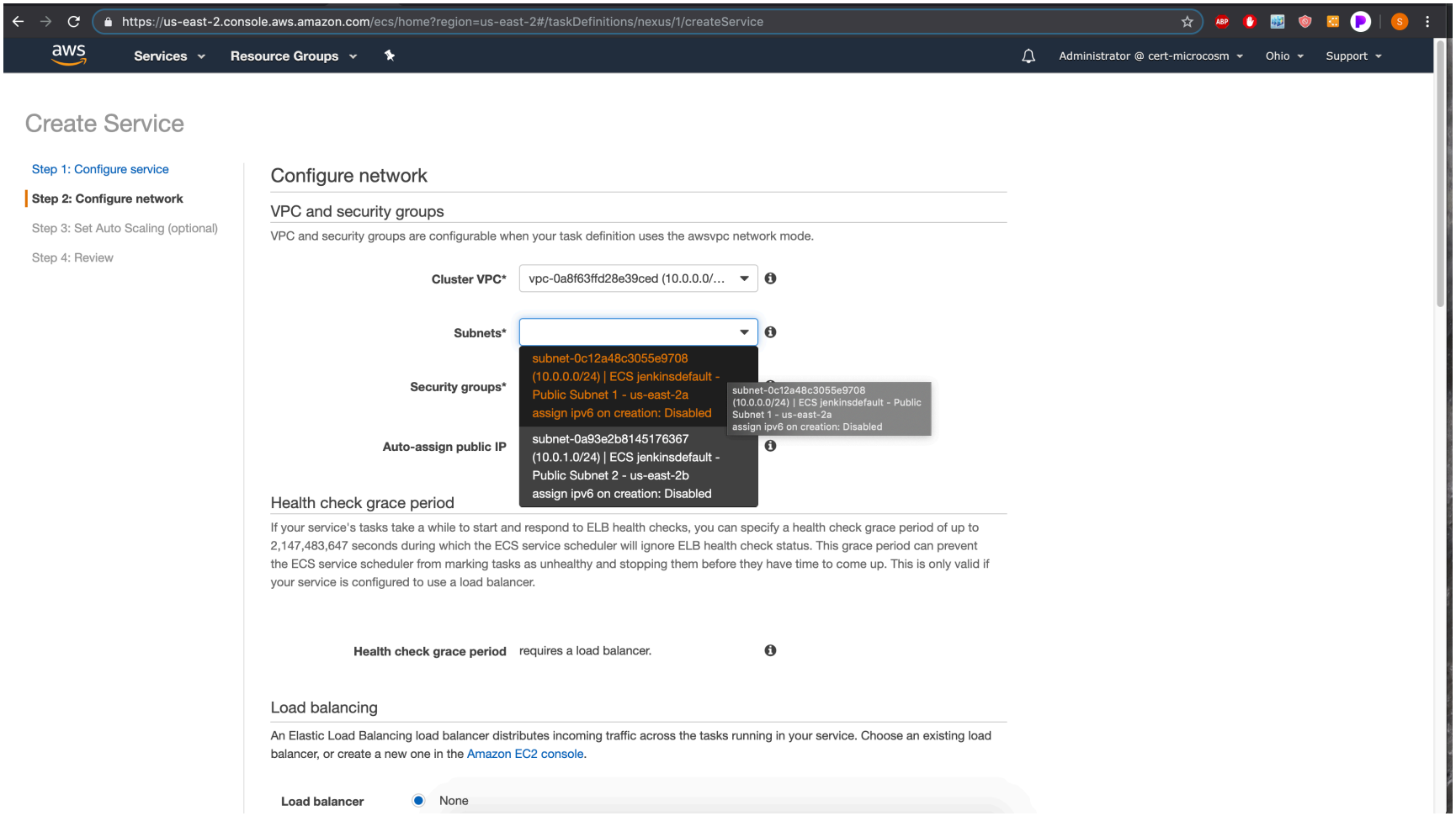
- Select a task definition and, under the Actions drop down, select Create Service



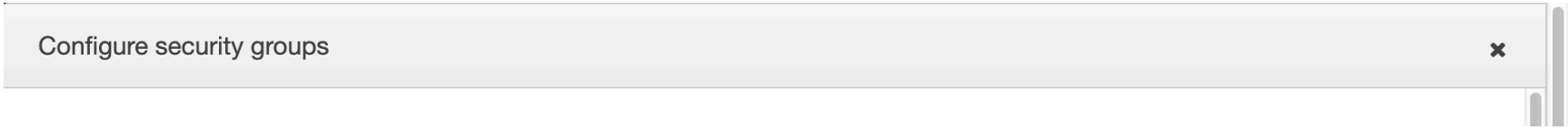
- For Launch type select Fargate
- Select the cluster you created earlier
- Enter the service name from the table above (though it can be anything)
- Enter 1 for the number of tasks (we only need 1 instance of each container/task defnition for the present practice)
- Press Next Step



- 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



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



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

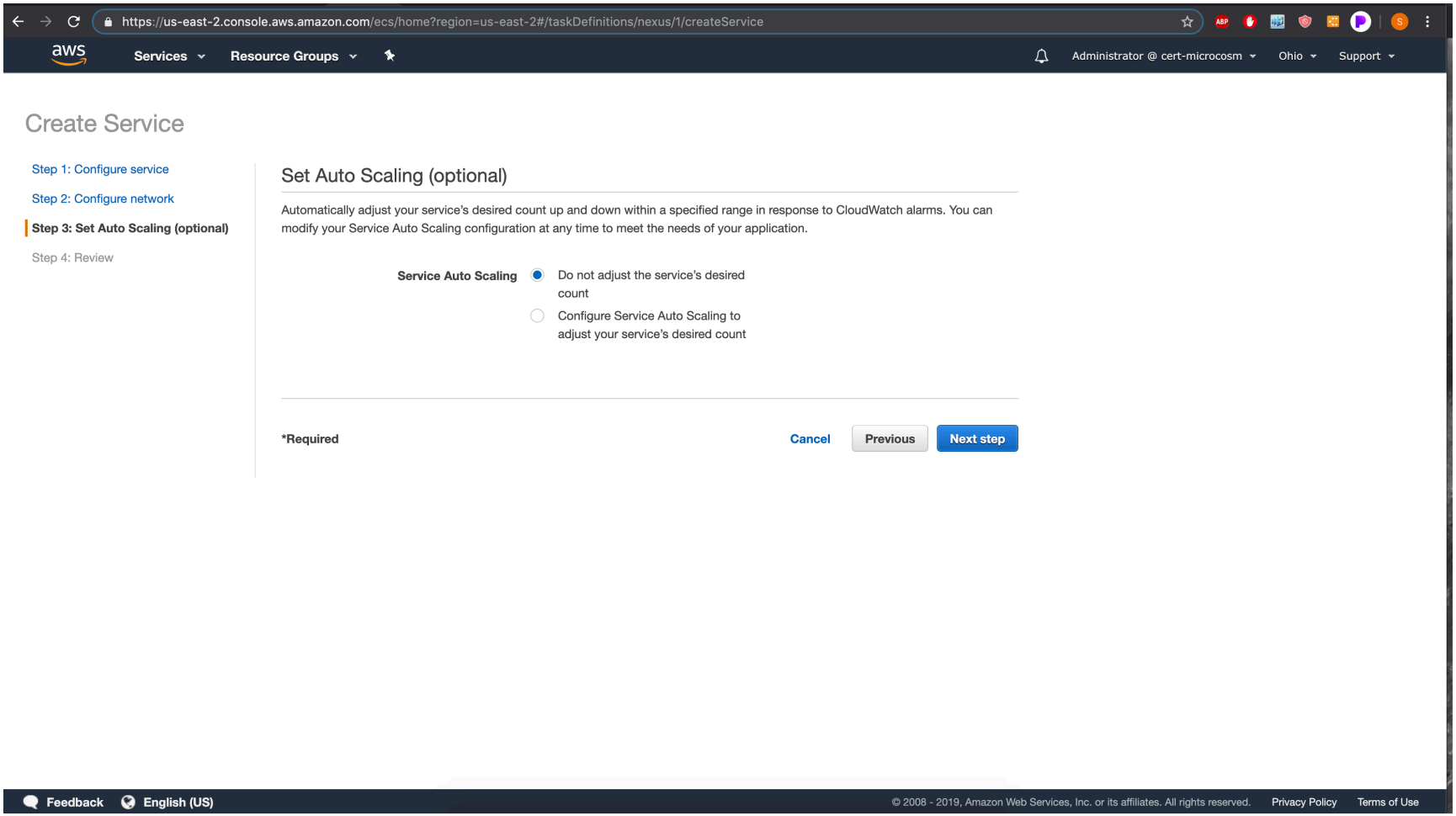
	Security group ID	Name	Description	Actions
☑	sg-016e9fba44d930d92	Dynamic2-StudentVPCSecurityGro…	Student VPC Security Group	Copy to new
☐	sg-050f7aa077e6edfa2	gitlab-2784	2019-04-16T15:33:53.395Z	Copy to new
☐	sg-0793fec6a9c4a1c01	default	default VPC security group	Copy to new

Inbound rules for selected security groups

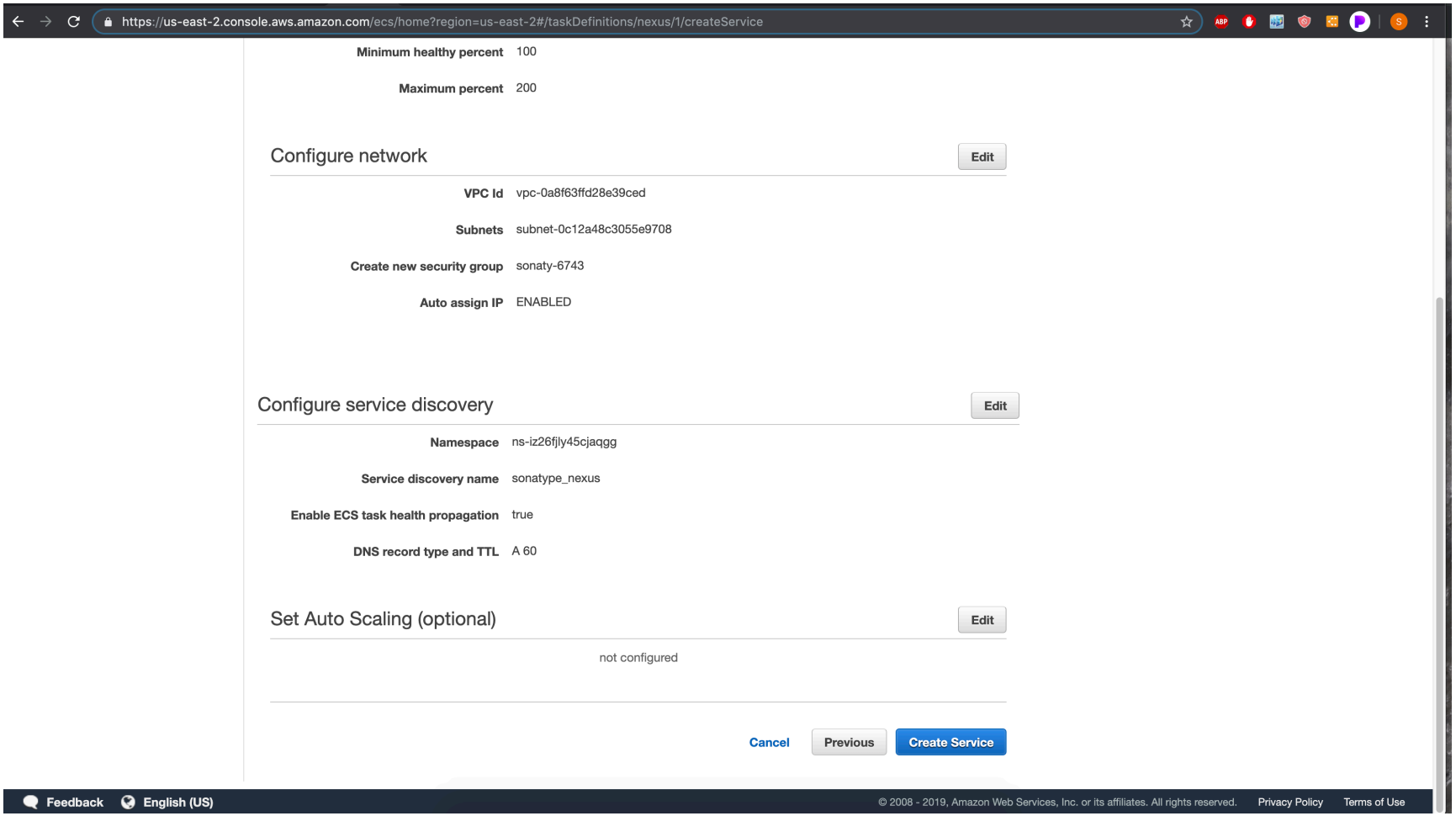
Security group ID	Type	Protocol	Port range	Source
sg-016e9fba44d930d92	HTTP	TCP	80	0.0.0.0/0
sg-016e9fba44d930d92	customtcp	TCP	9000	0.0.0.0/0
sg-016e9fba44d930d92	customtcp	TCP	8080	0.0.0.0/0
sg-016e9fba44d930d92	SSH	TCP	22	0.0.0.0/0
sg-016e9fba44d930d92	customtcp	TCP	50000	0.0.0.0/0
sg-016e9fba44d930d92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

Cancel Save

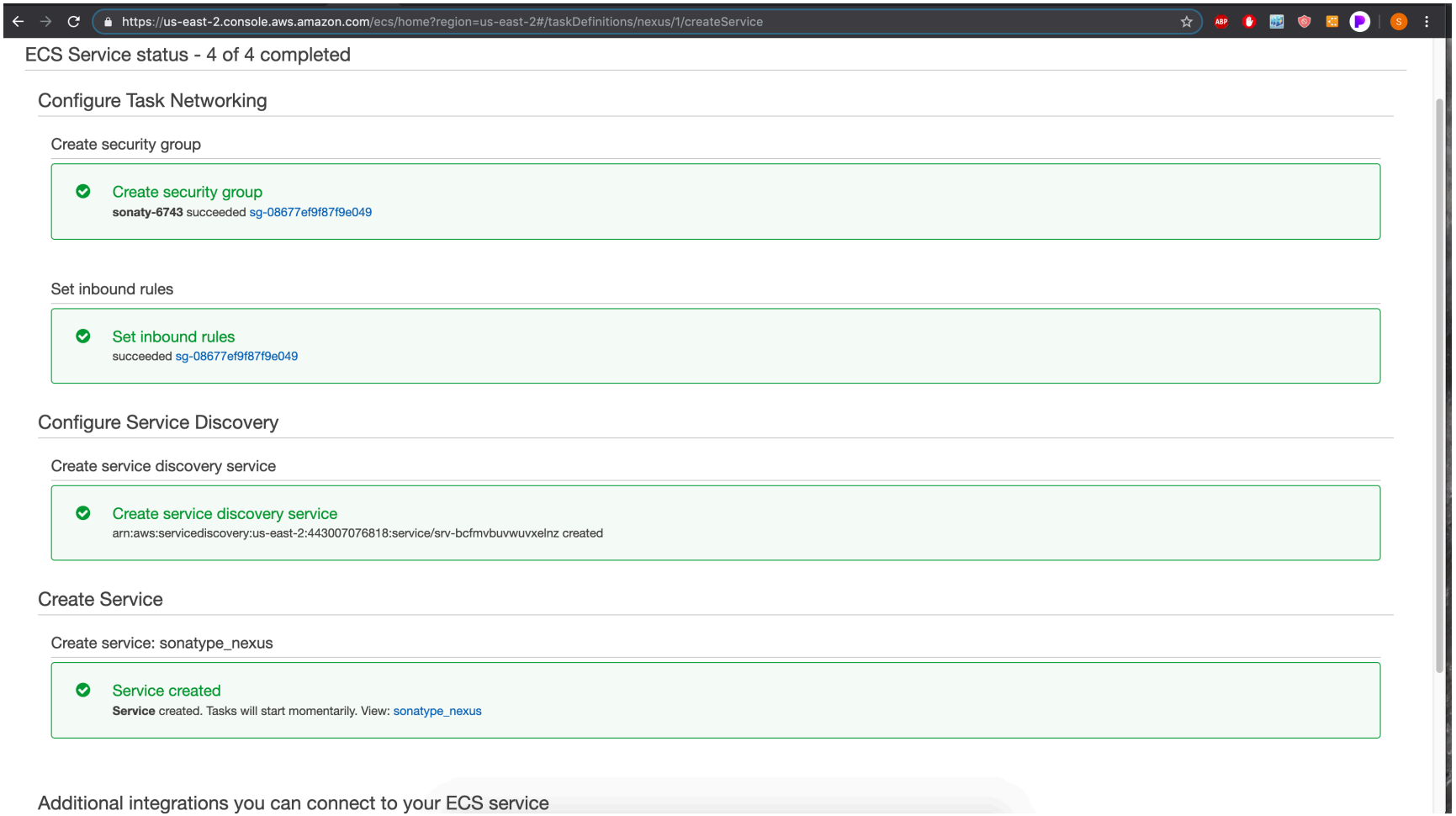
- Select Next Step



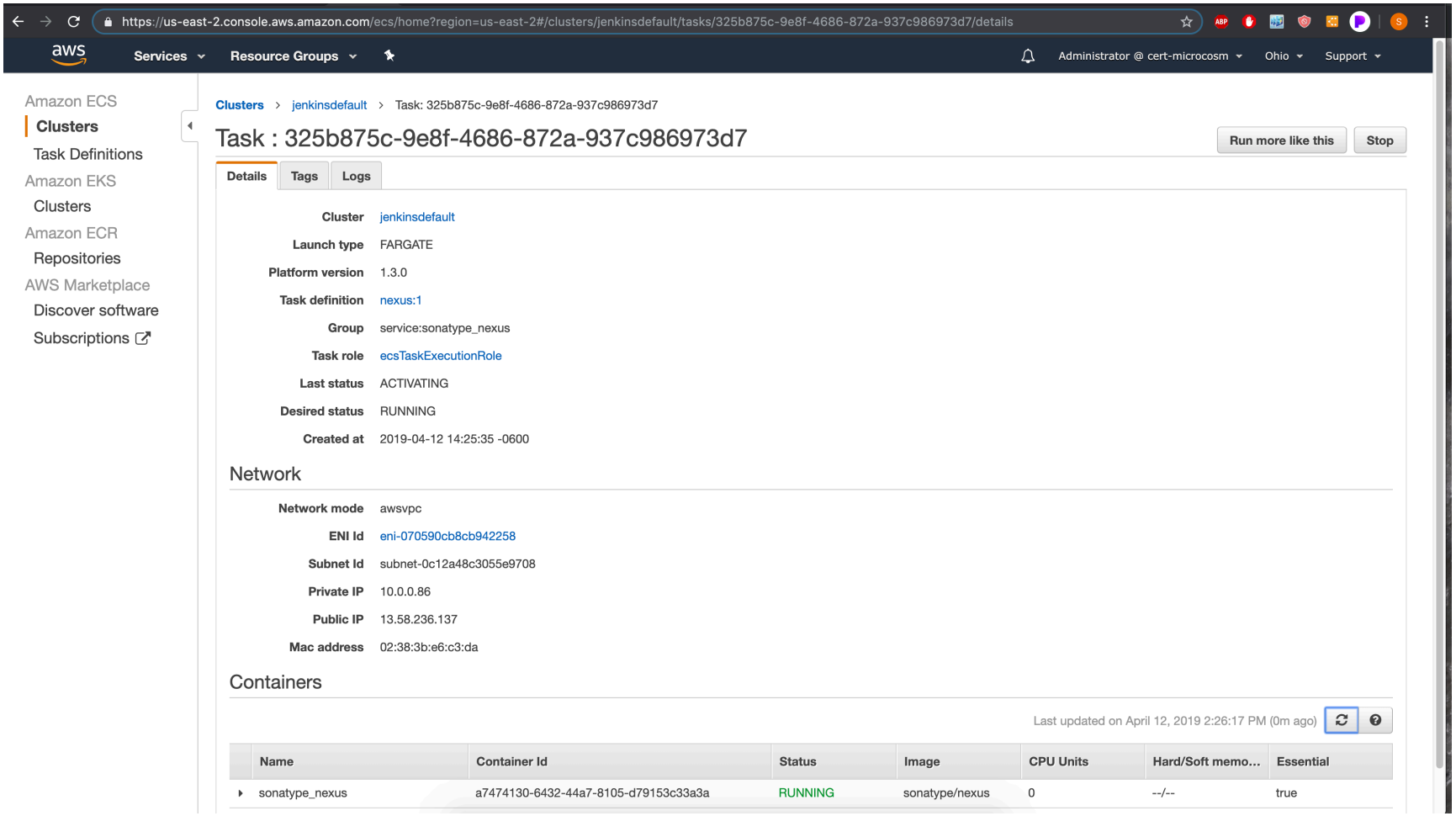
- Review and press Create Service



- Once the service is successfully created, select View Service

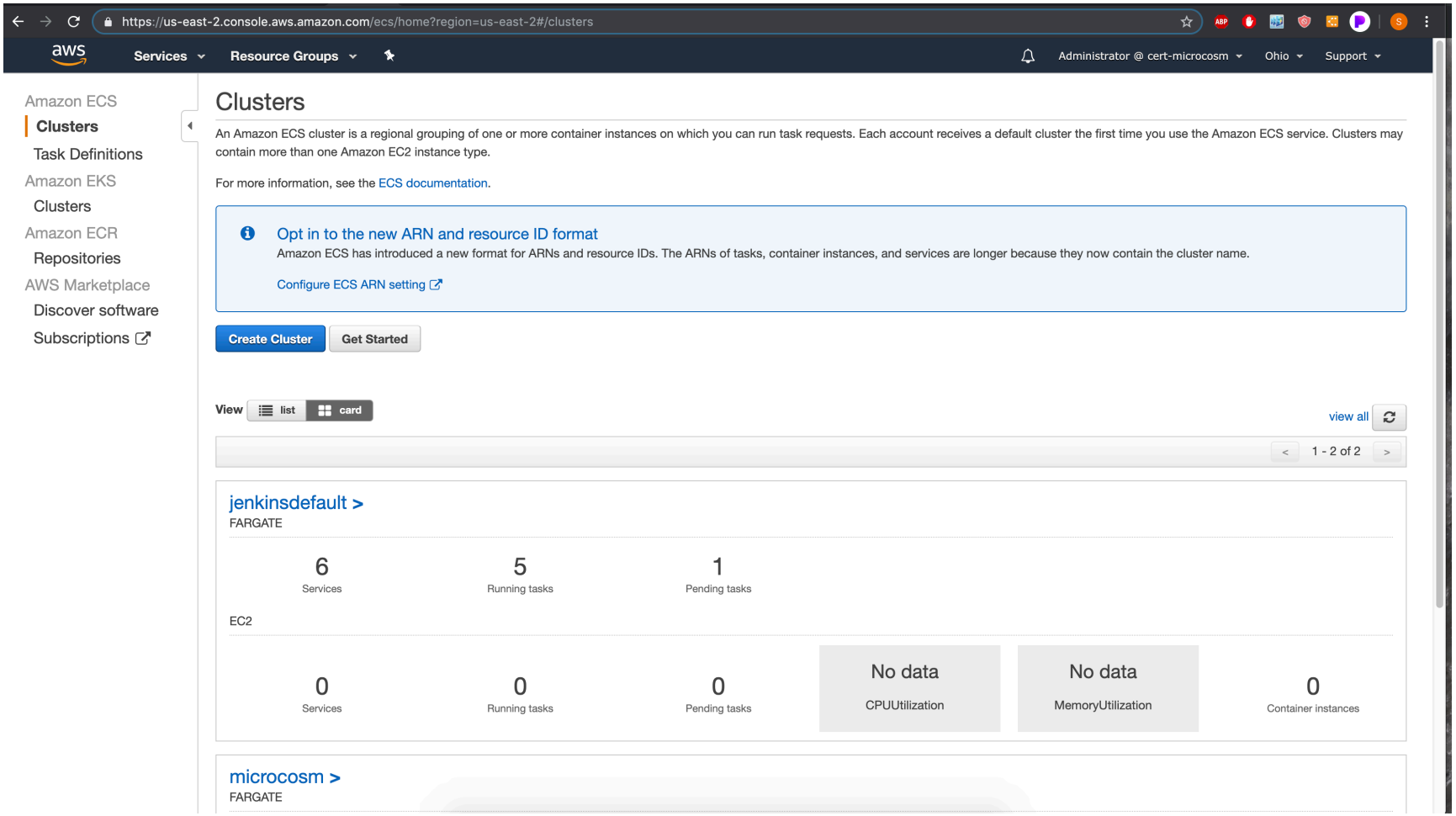


- Review the service's properties. Observe the creation of the new task.
- Once the task is present, select the task ID.
- View the task status, until the status is green and says Running
- Now you should be able to see both the private and public IP address of the container. This is important for the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

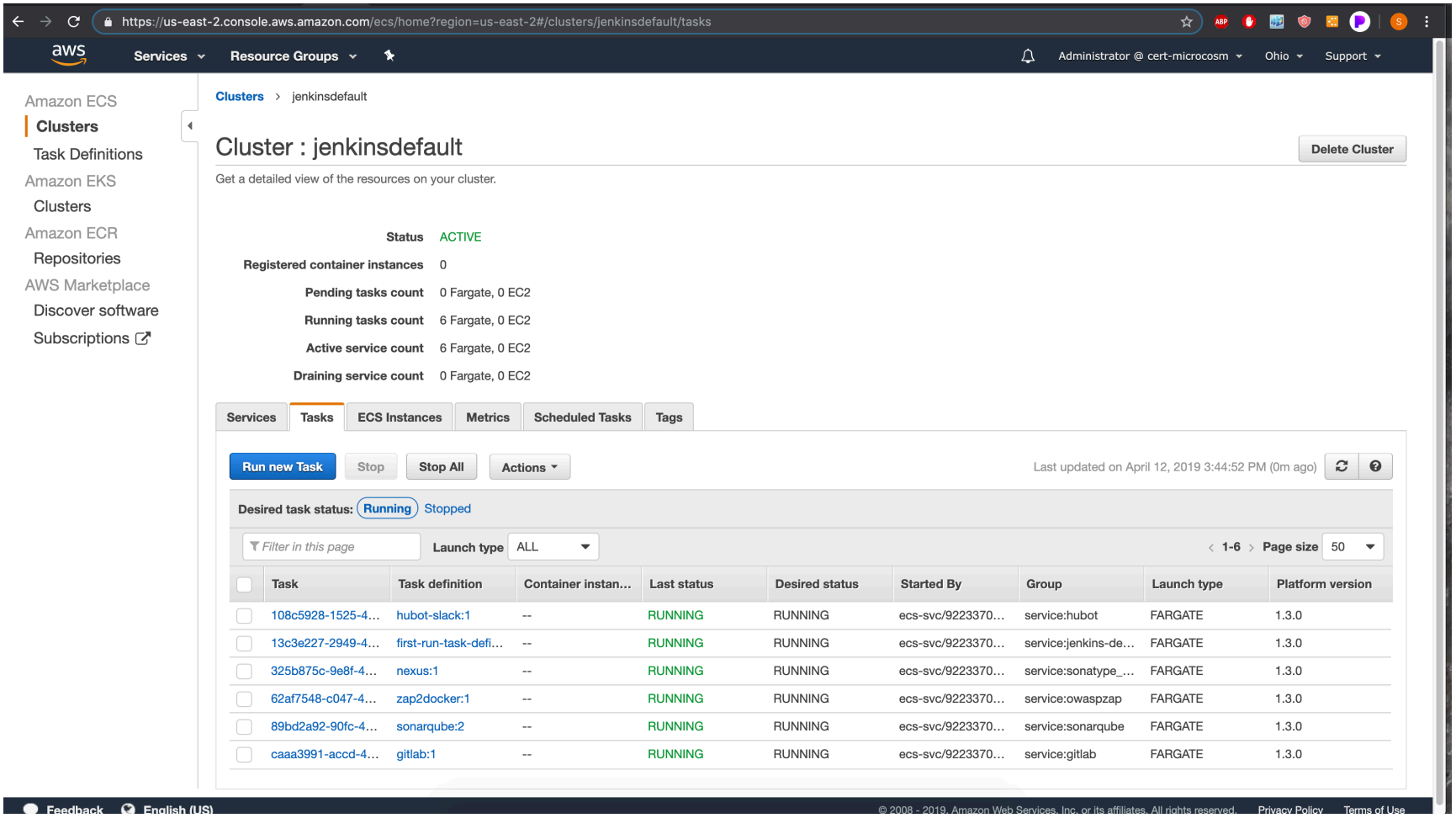


NOTE

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



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



Return to Table of Contents

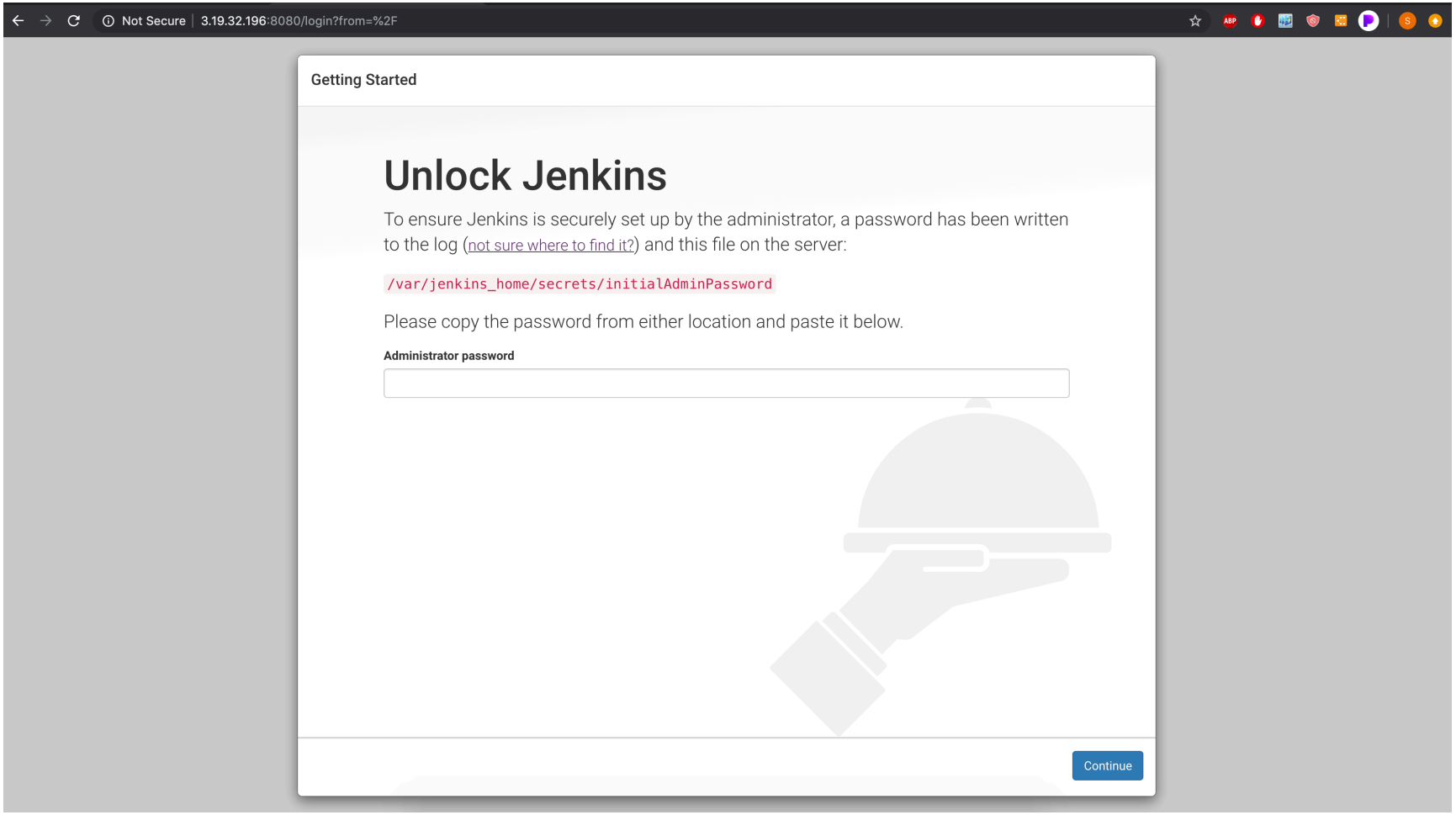
Setting up Jenkins for DevOps Pipeline (Microcosm)

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

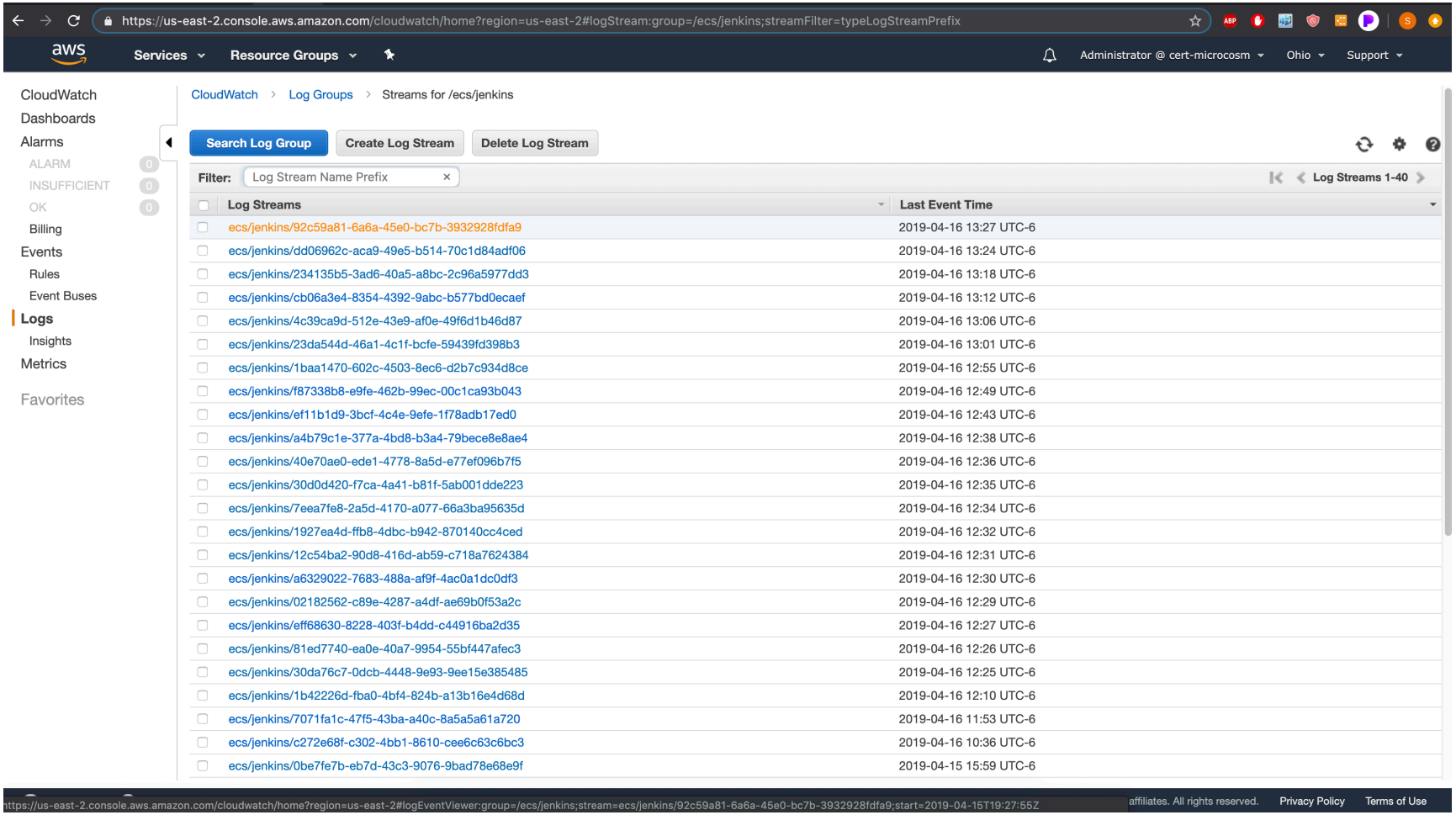
Note, that without setting up persistant volume storage (not covered here), if the task (container) fails in any way or

*is stopped or otherwise shutdown, **ALL** of the settings and data entered into that container will be lost. Therefore, be careful.*

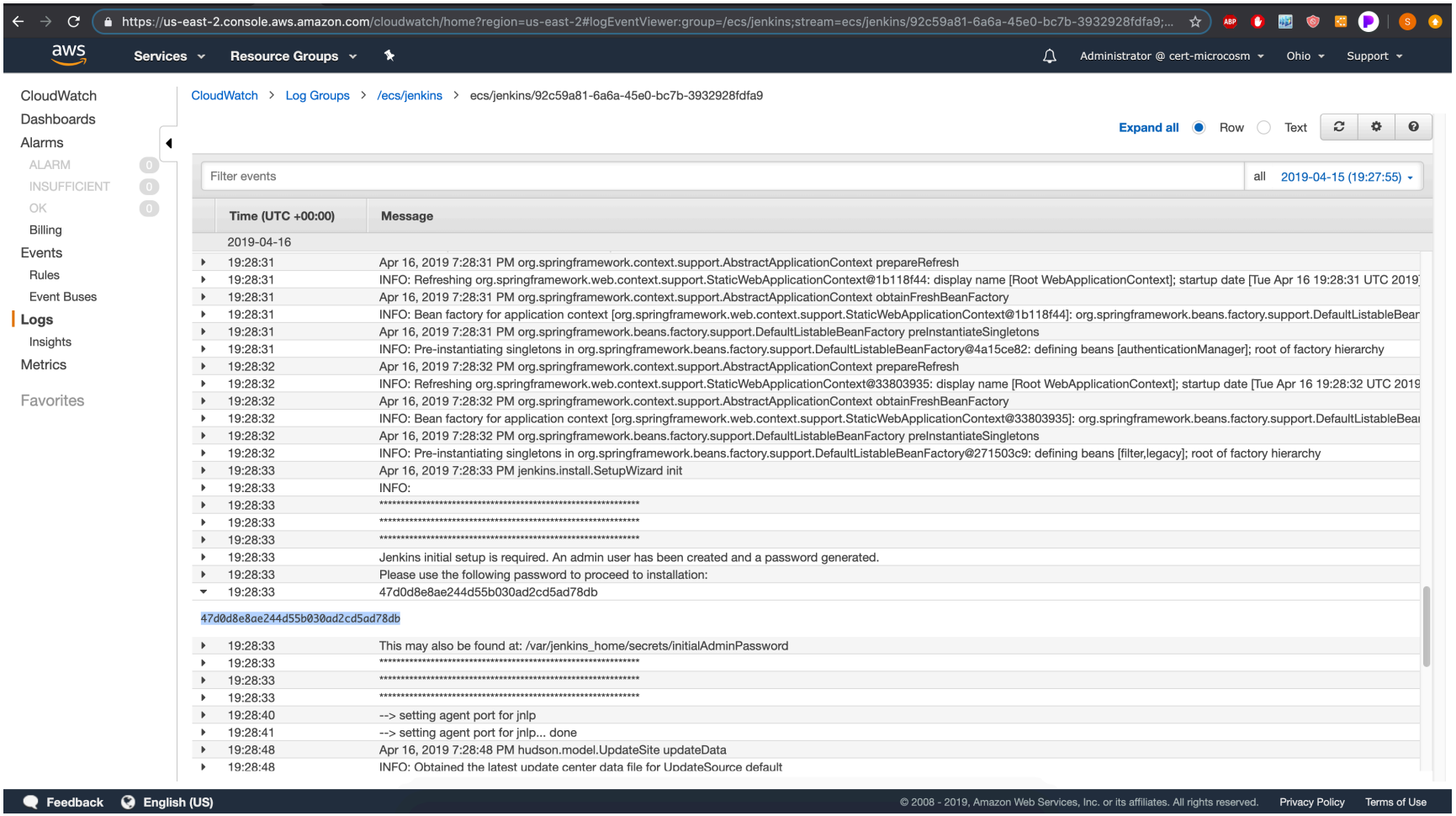
- Navigate to the IP address of the Jenkins task at port 8080 (eg 24.23.22.21:8080)
    - Remember that all task IP addresses can be found by clicking on the cluster, then the tasks tab and finally the task ID
- The first time you navigate to Jenkins, you will see the following prompt asking you to unlock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



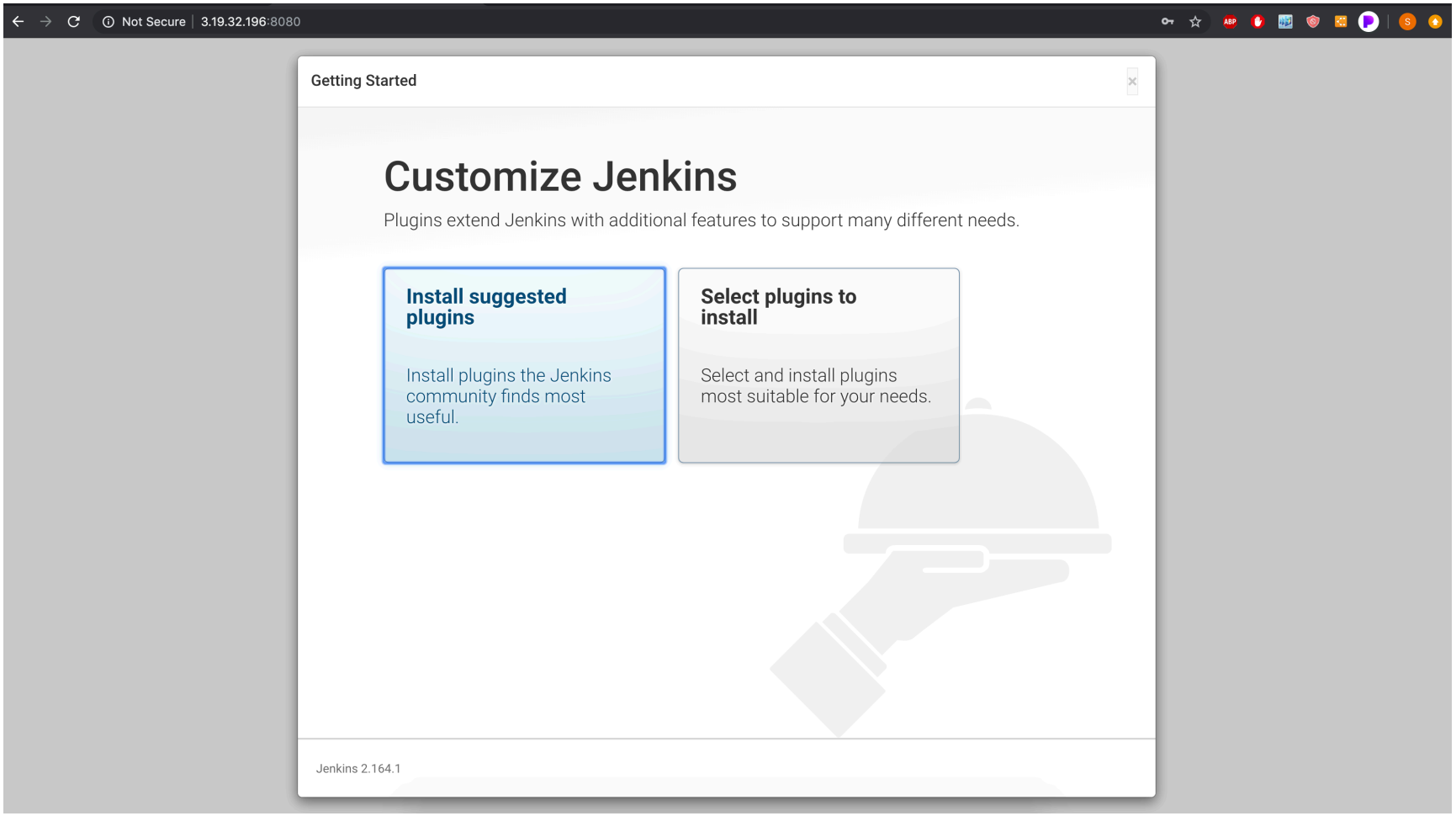
- From the services dropdown find/select CloudWatch
- Select Logs from the column on the **Left**
- Select /ecs/jenkinsTSTNAME
- Sort the Log Streams by clicking on the Last Event Time column header so that the most recent log stream date is at the top
- Select the most recent Log Stream



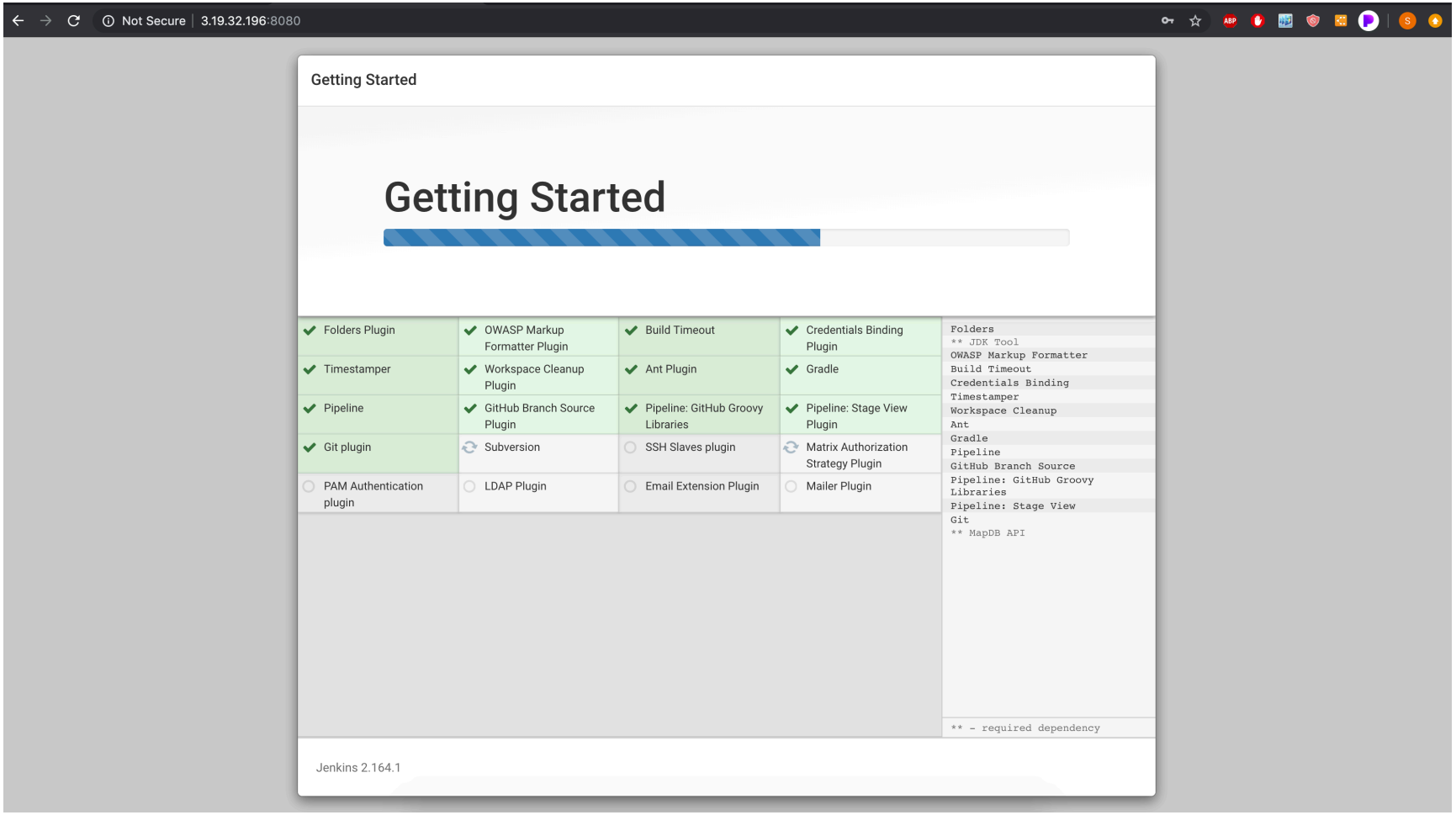
- Scroll down until you find the line "Please use the following password to proceed to installation"
- Select the line below this, which contains a string of seemingly random character
- Highlight and copy that string
- Paste that string into the Jenkins Getting Started window and press Continue



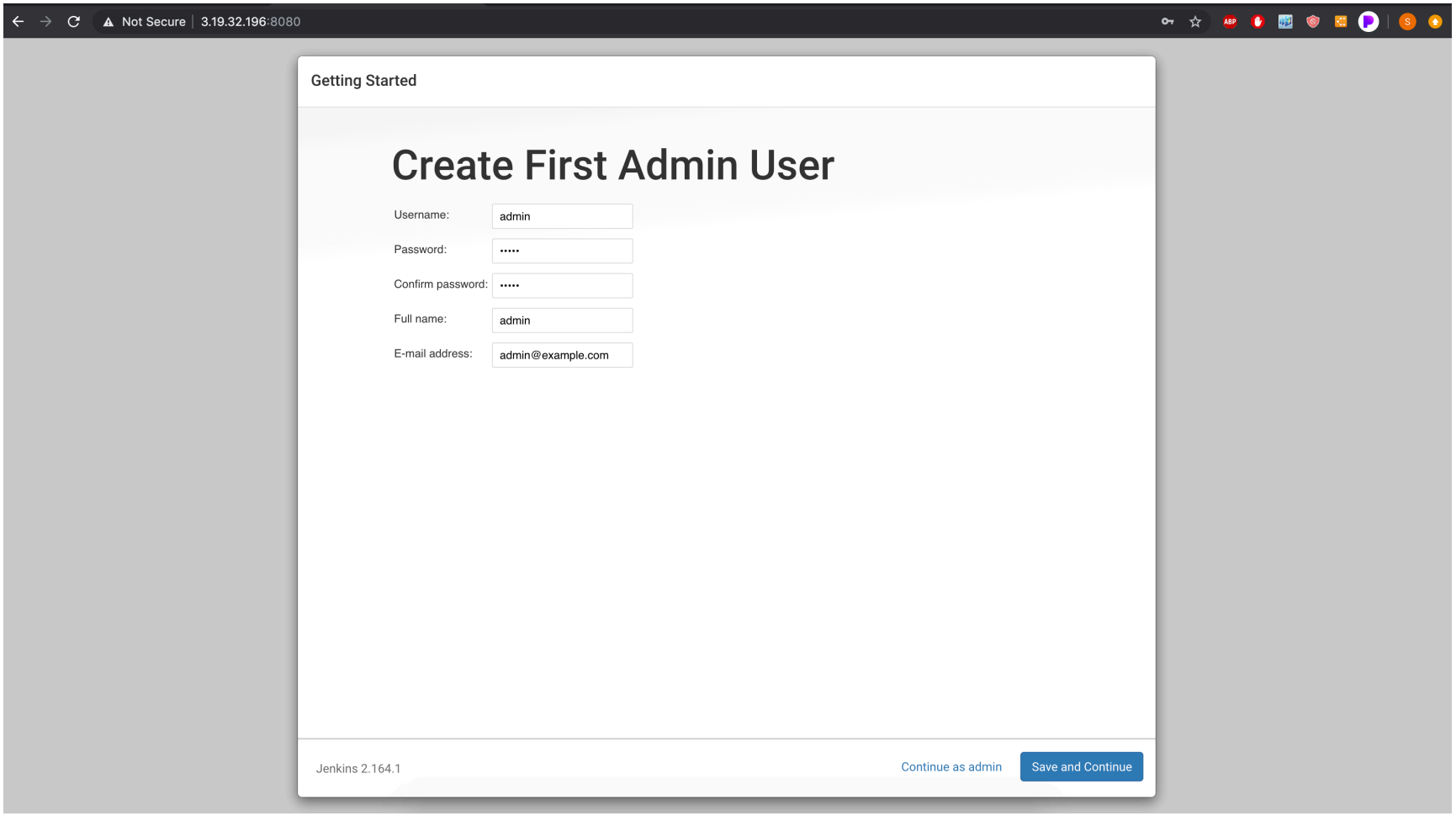
- Click Install suggested plugins



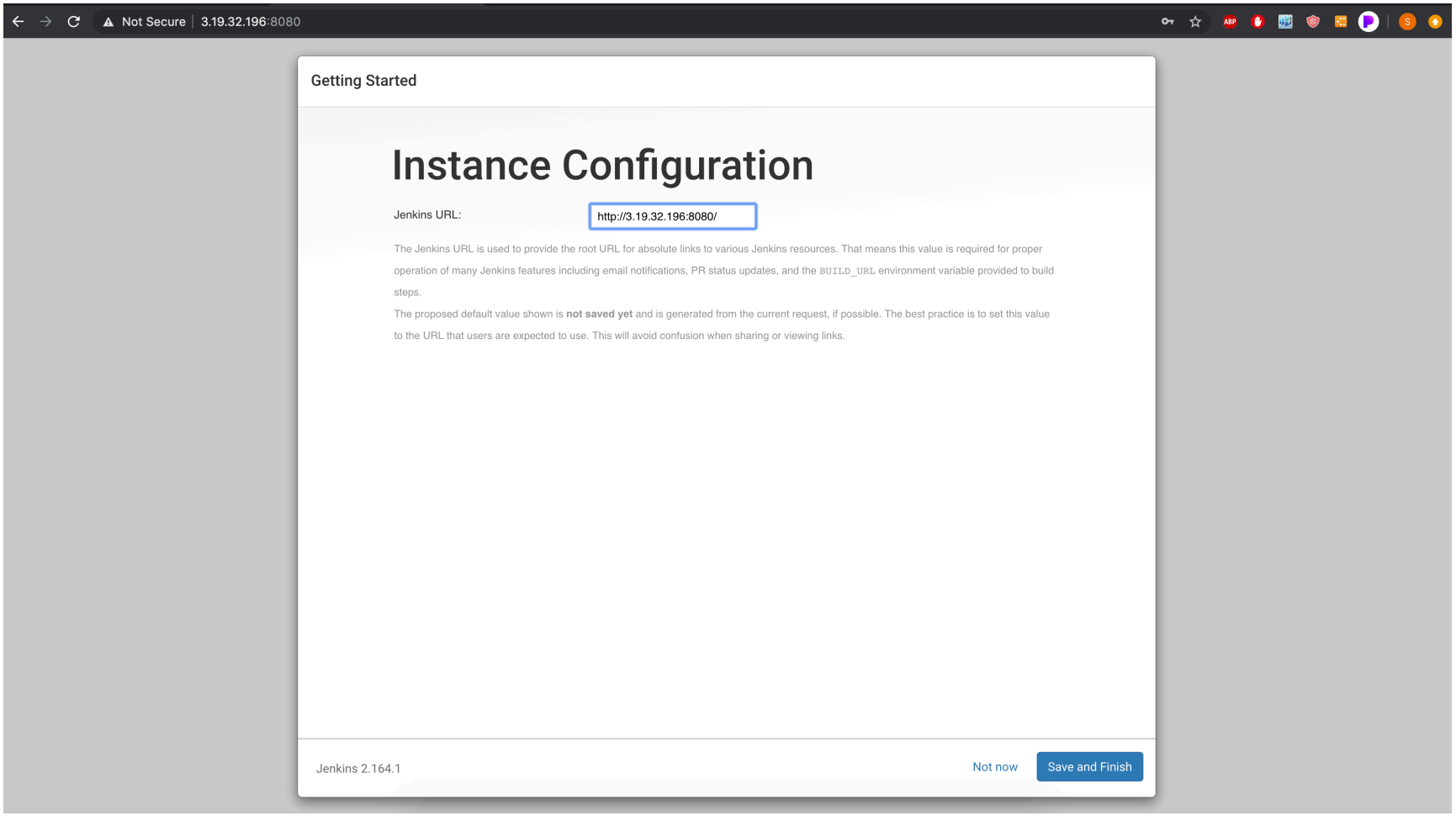
- Wait while the plugins are installed



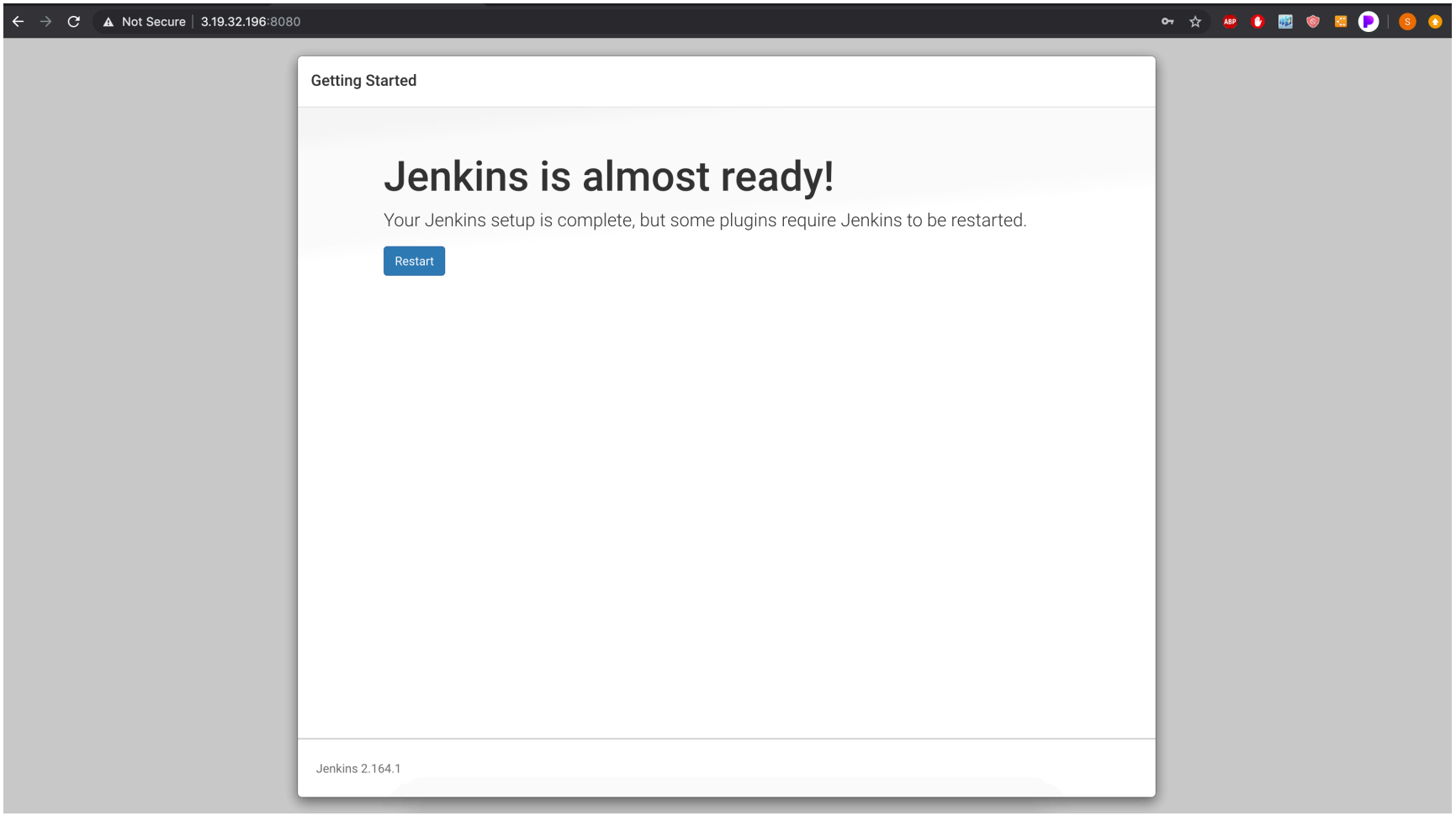
- Fill in the information for the admin user. Make sure to remember your password.
- Select Save and Continue



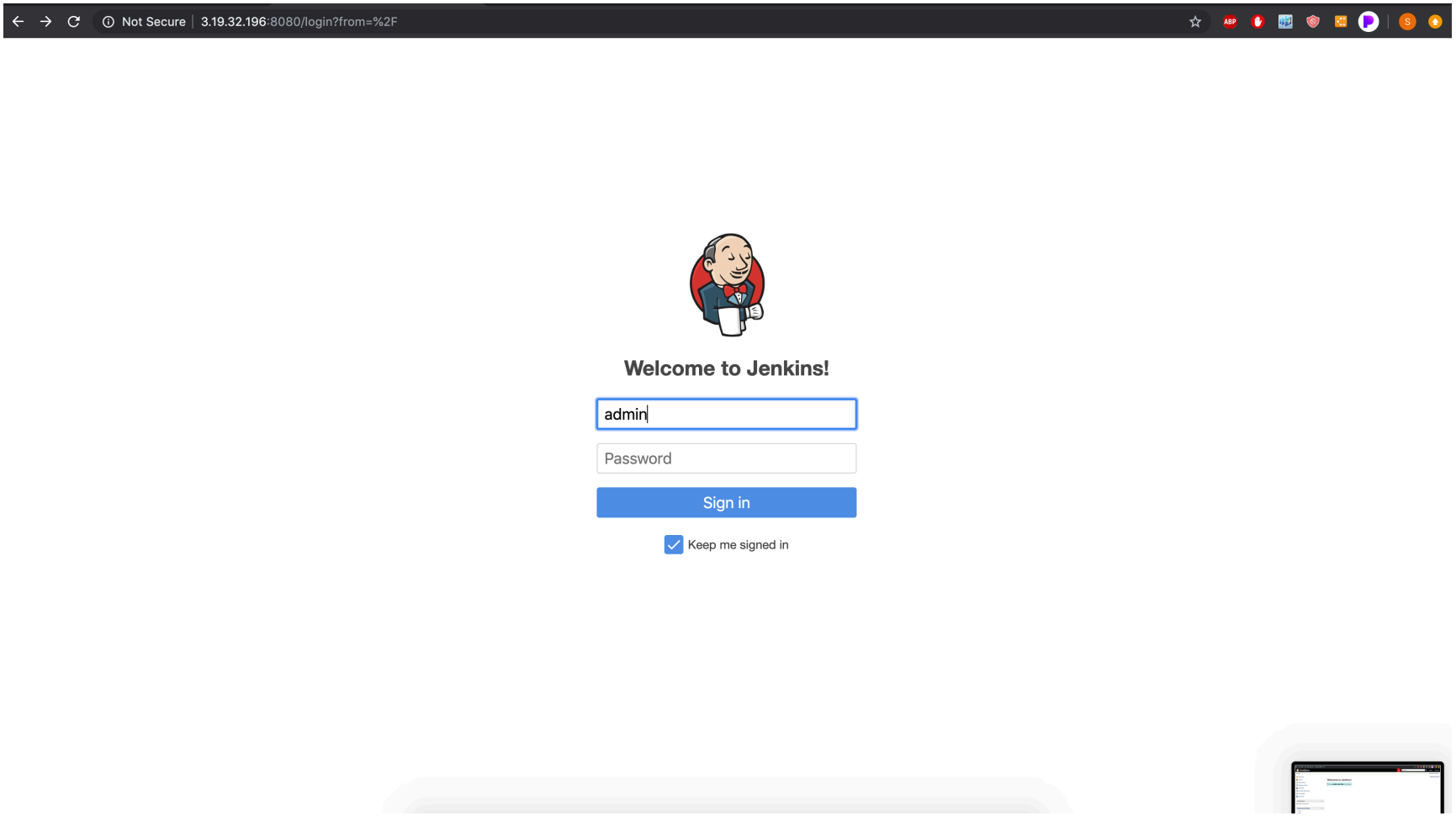
- Select Save and Finish



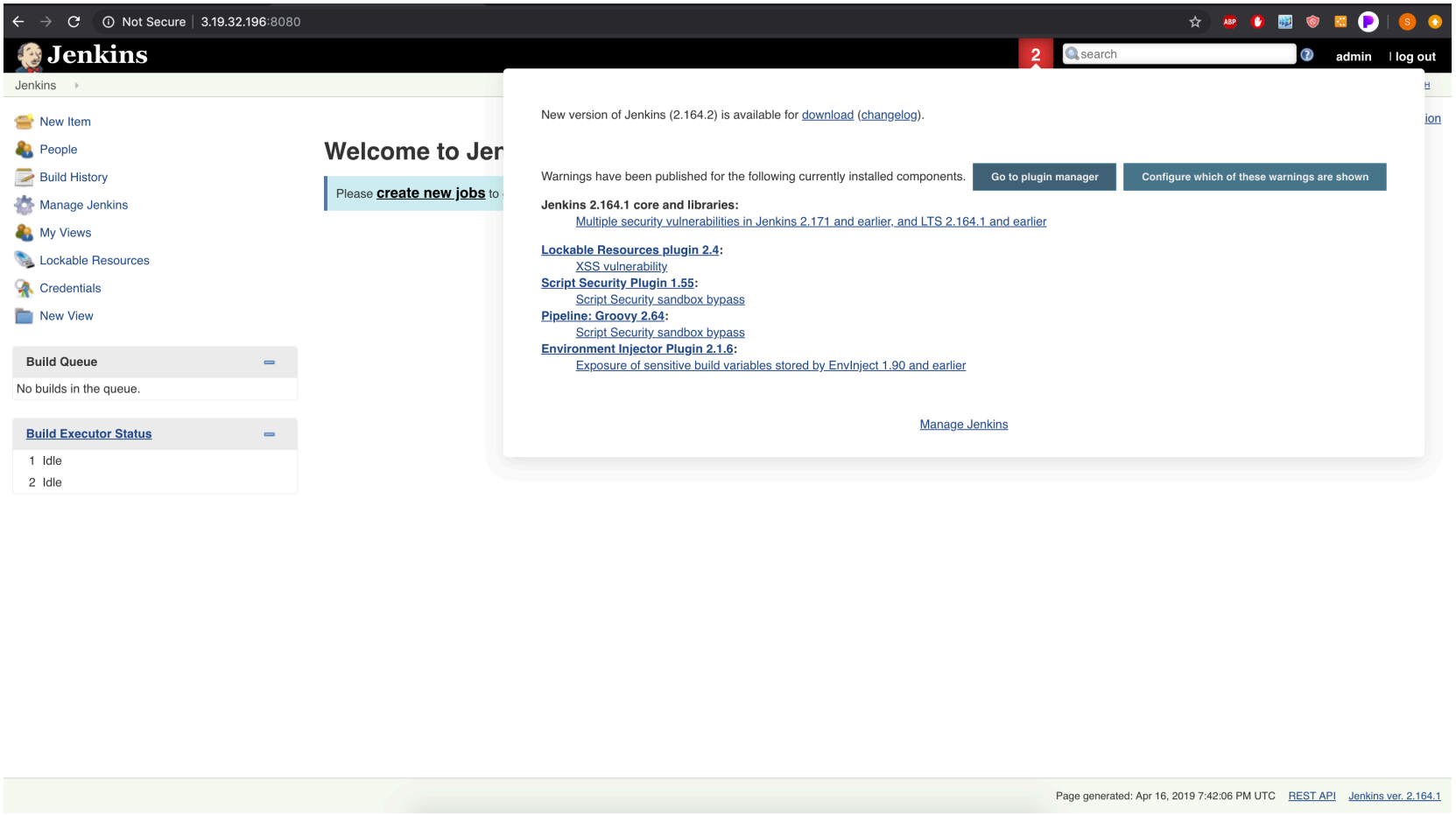
- Press Restart
- Wait (if waiting longer than a few minutes, refresh window or, open Jenkins in a new window)



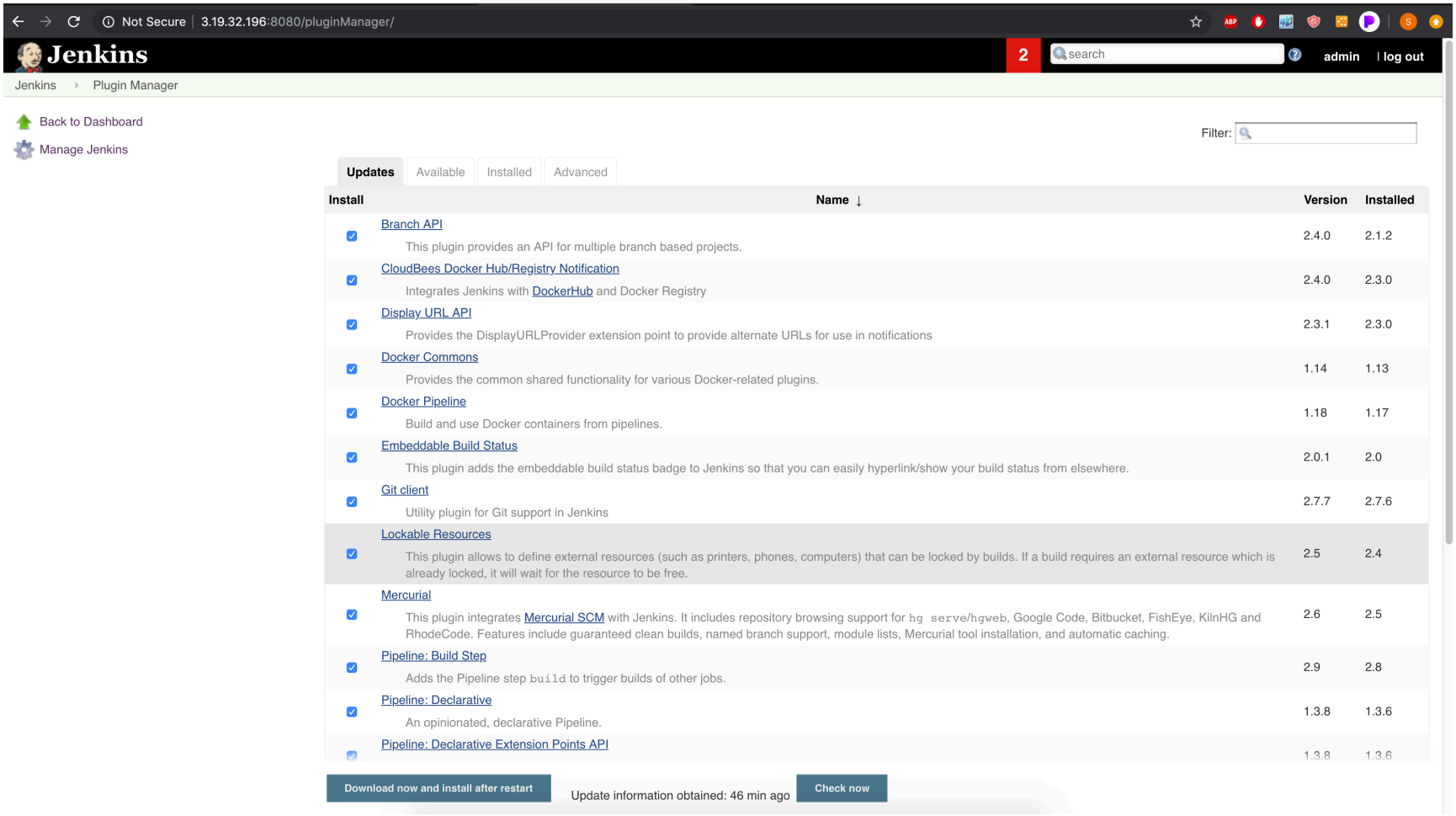
- If, after restart, you see a login prompt, use the credentials you entered when creating the admin user previously



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

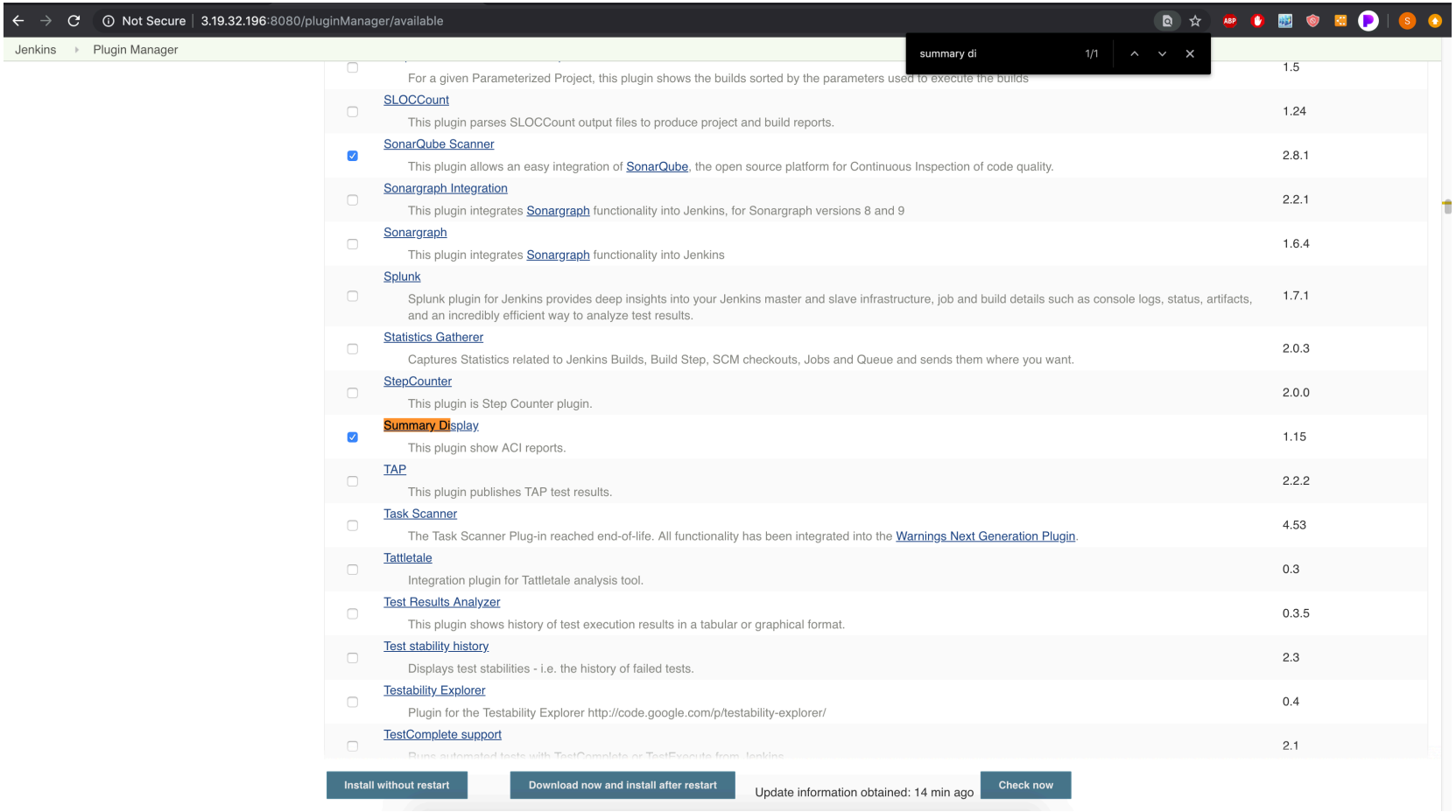


- If updates are available, scroll to the bottom of the page and click select all
- Select Download Now and install after restart
- Click "Restart Jenkins when installation is complete and no jobs are running"
- at top-left menu, click "back to Dashboard"

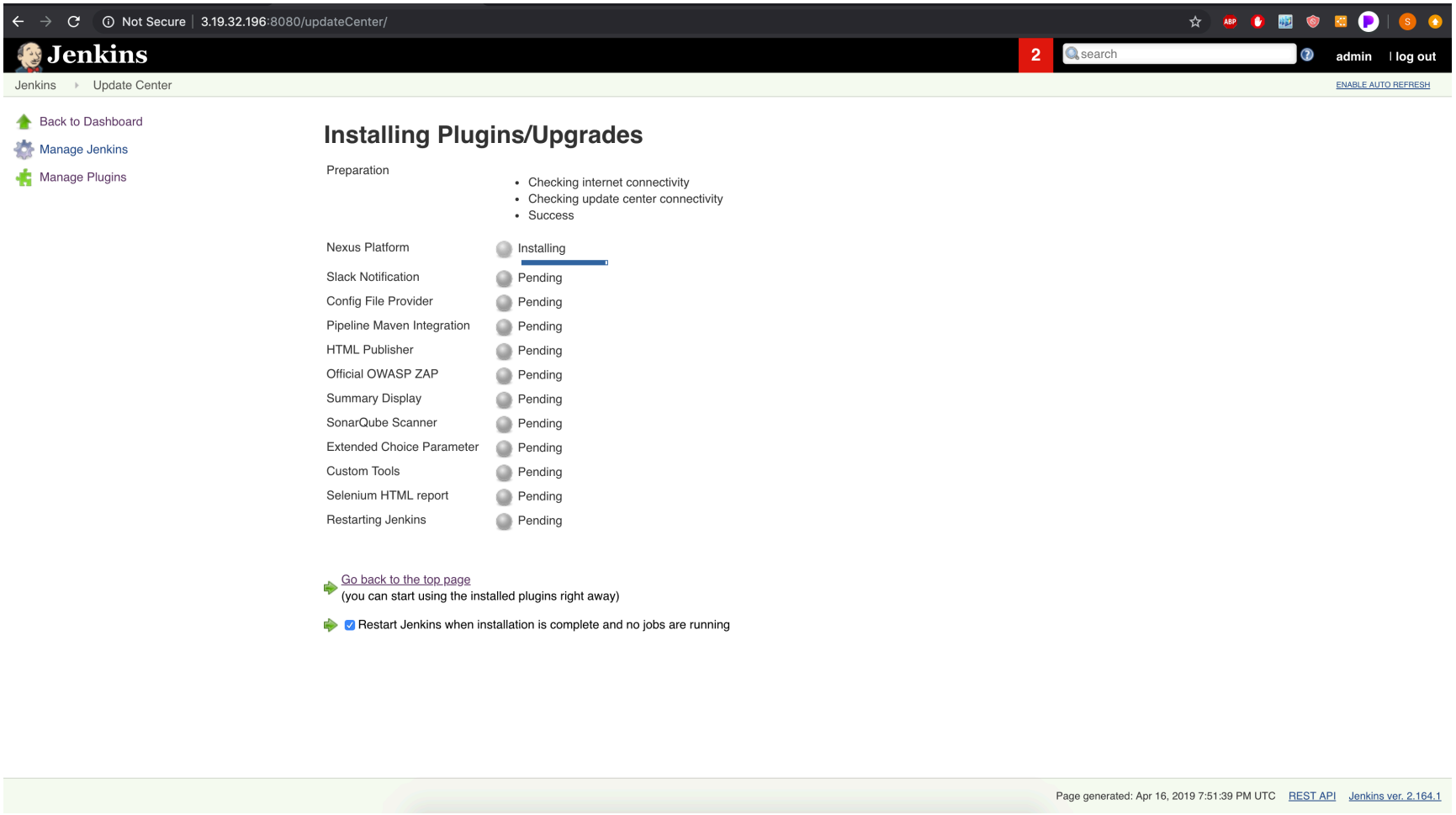


- On the Plugin Manager page, select the Available Tab
  - search & select: "Git Plugin" (May be pre-installed)
  - search & select: "Pipeline Maven Integration"
  - search & select: "SonarQube Scanner"
  - search & select: "Nexus Platform"
  - search & select: "AWS CodeDeploy Plugin for Jenkins"

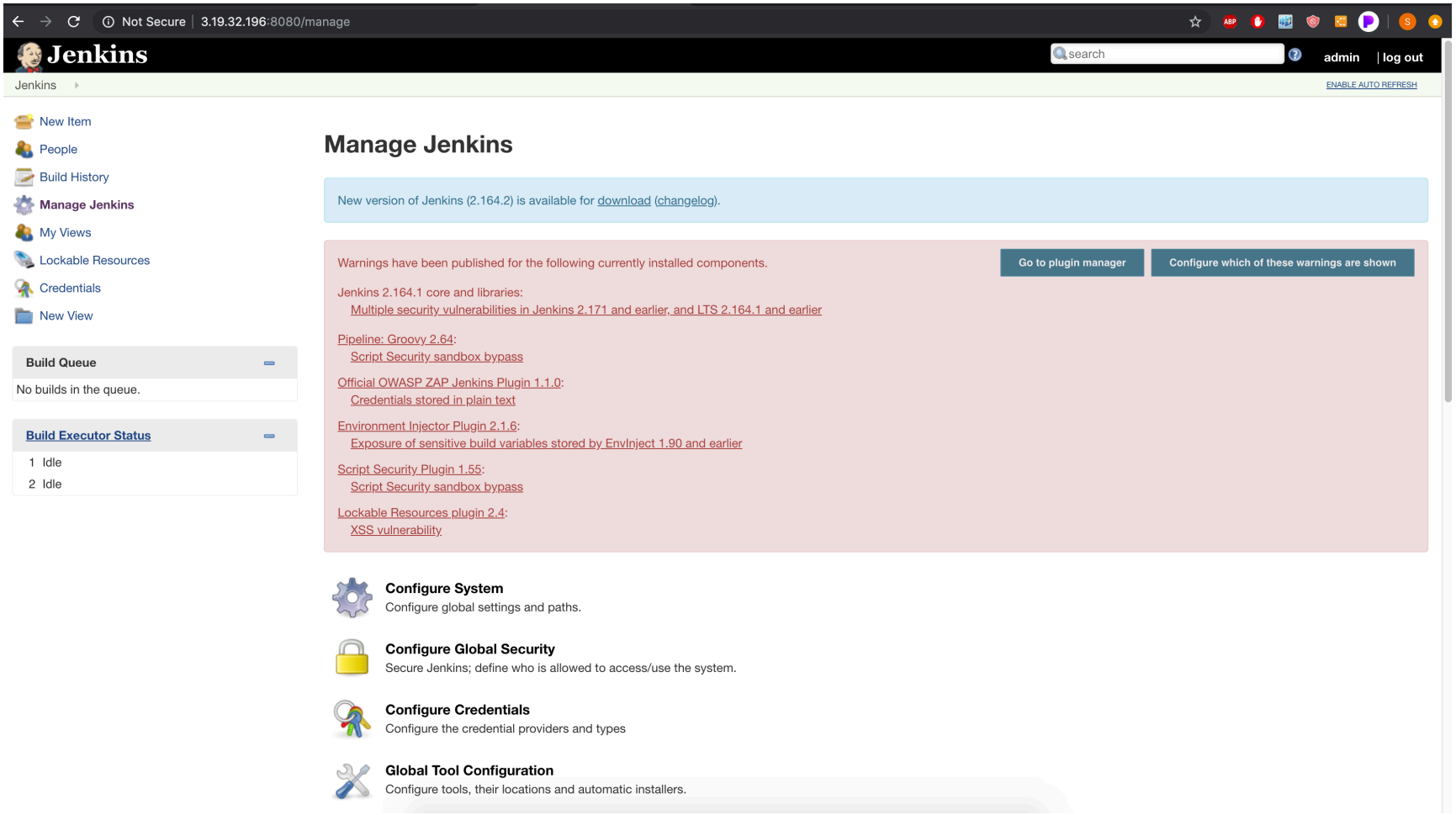
- search & select: "AWS CloudWatch Logs Publisher"
- click "install without restart" at bottom of page
- check box next to "Restart Jenkins when installation is complete and no jobs are running."
- at top-left menu, click "back to Dashboard"



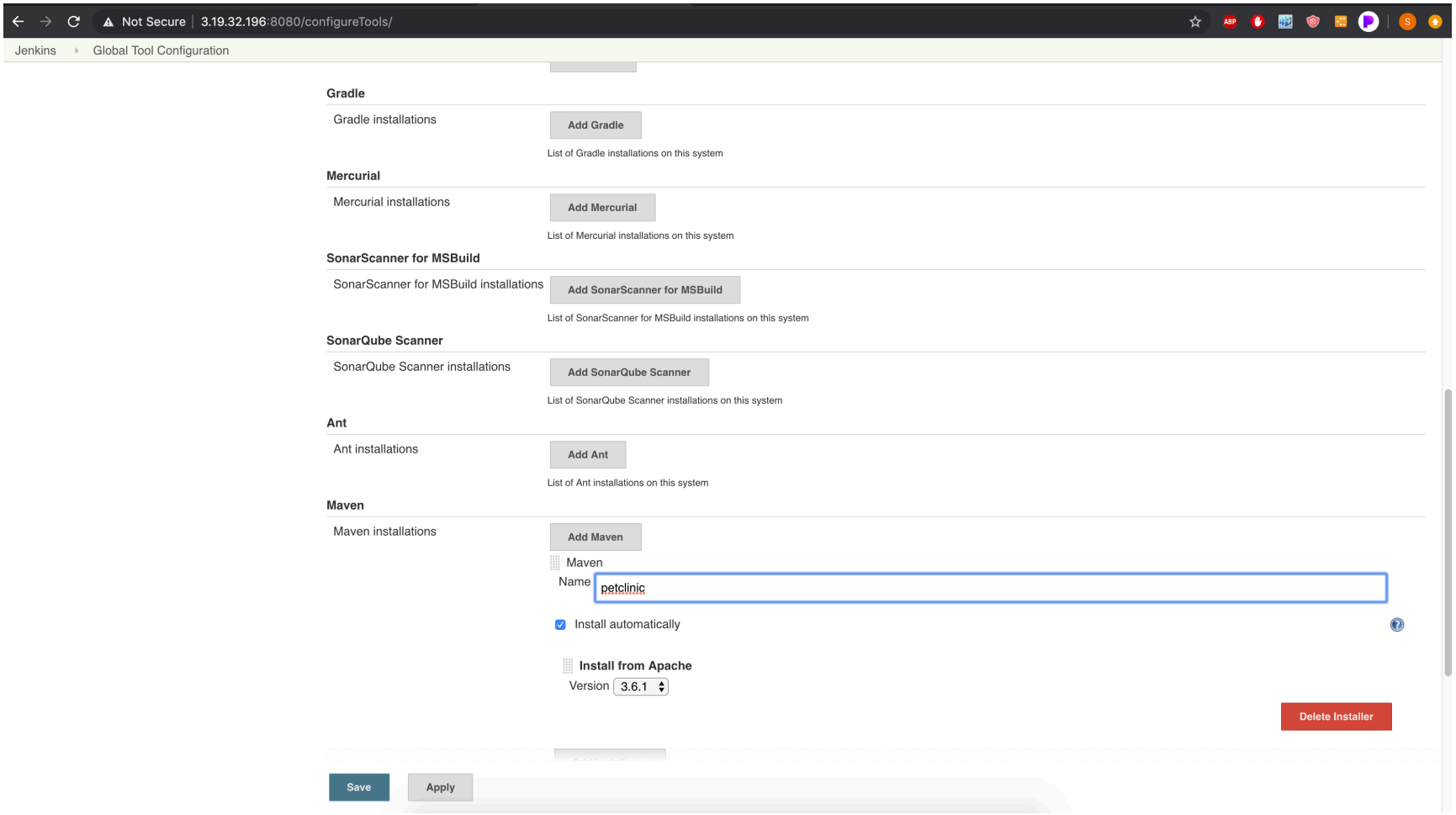
- Wait until the installation completes
- Jenkins will restart in the background and the UI may appear to be hung. Feel free to refresh the page after a few minutes



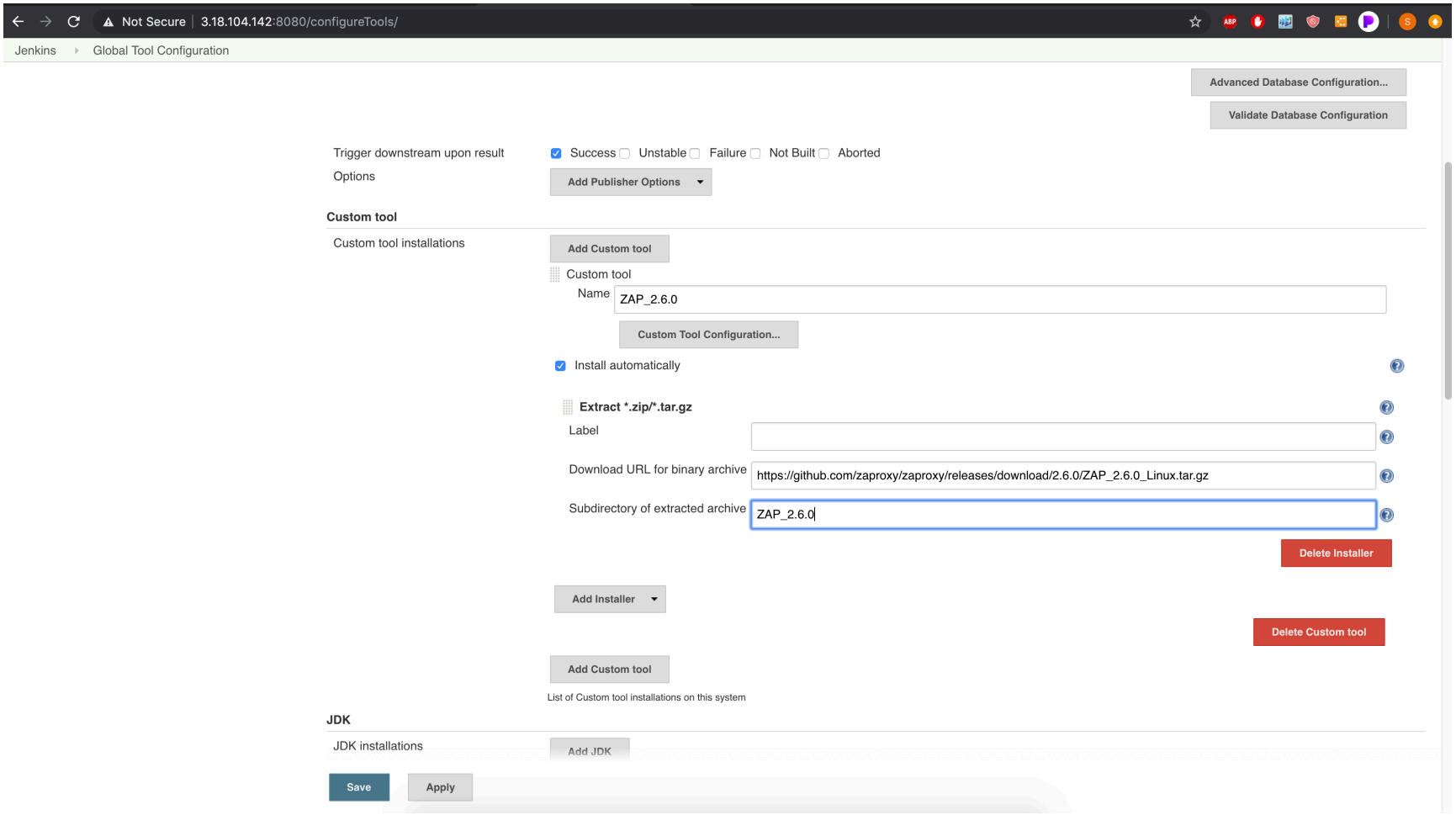
- Return to the home page and click "Manage Jenkins"
- click "Global Tool Configuration"



- 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 automaticaly is selected
- click Apply



- Click Apply
- Click Save



Slack & Hubot integration with Jenkins

Return to Table of Contents

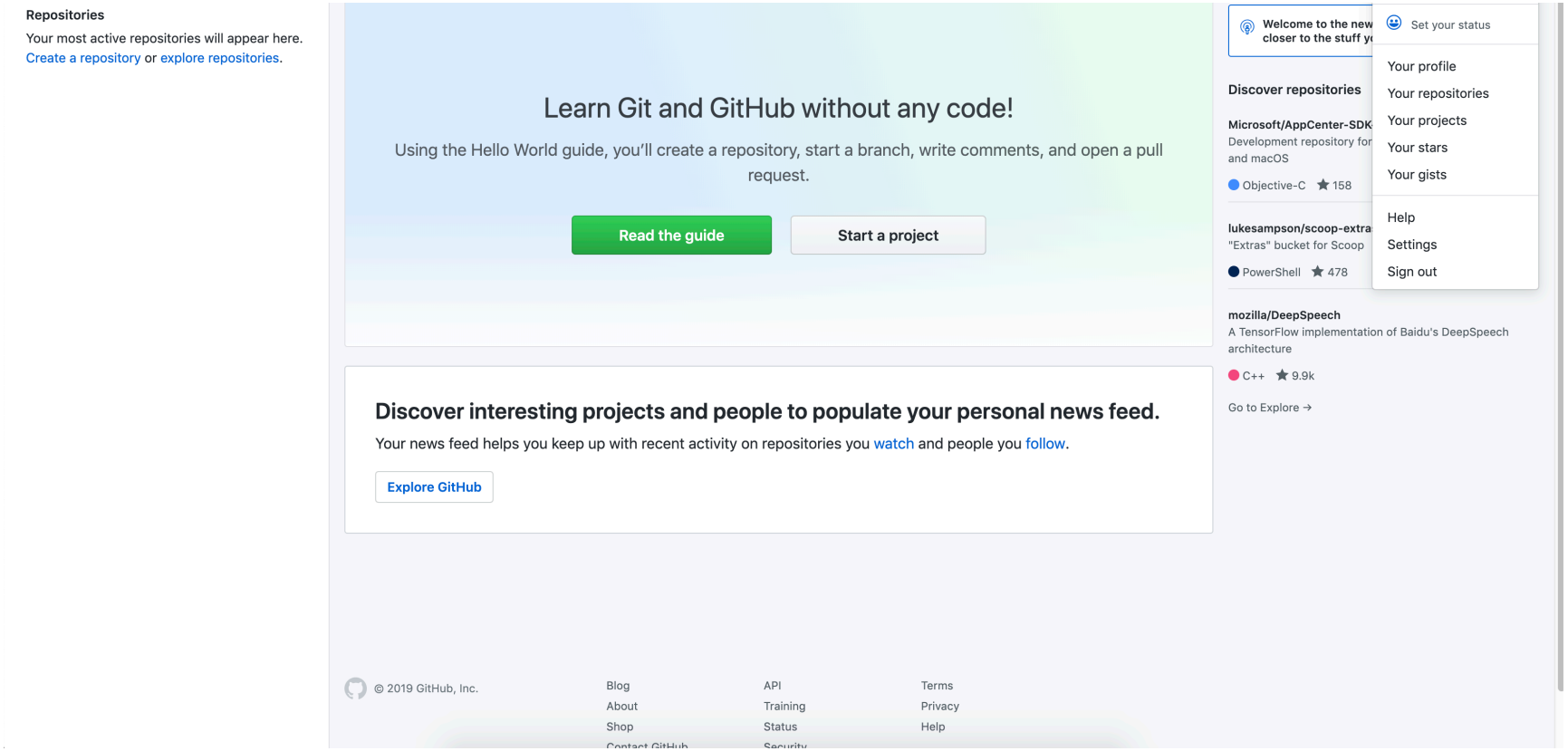
Deploying test code in DevOps Pipeline

We are going to use a publicly available, SEI created, test project hosted on Github to test our deployment pipeline.

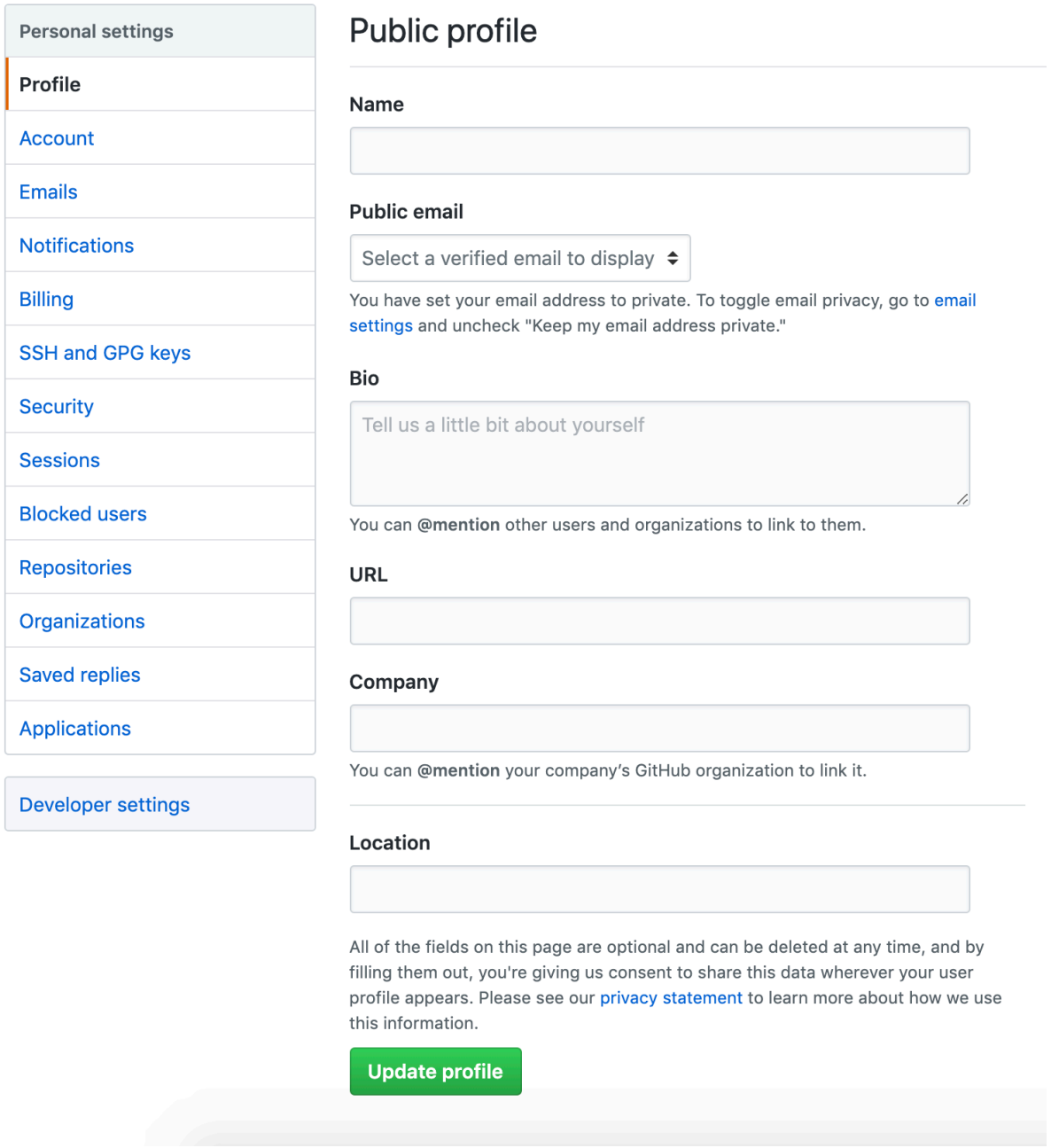
The following steps will guide you through that process.

Public Github Setup

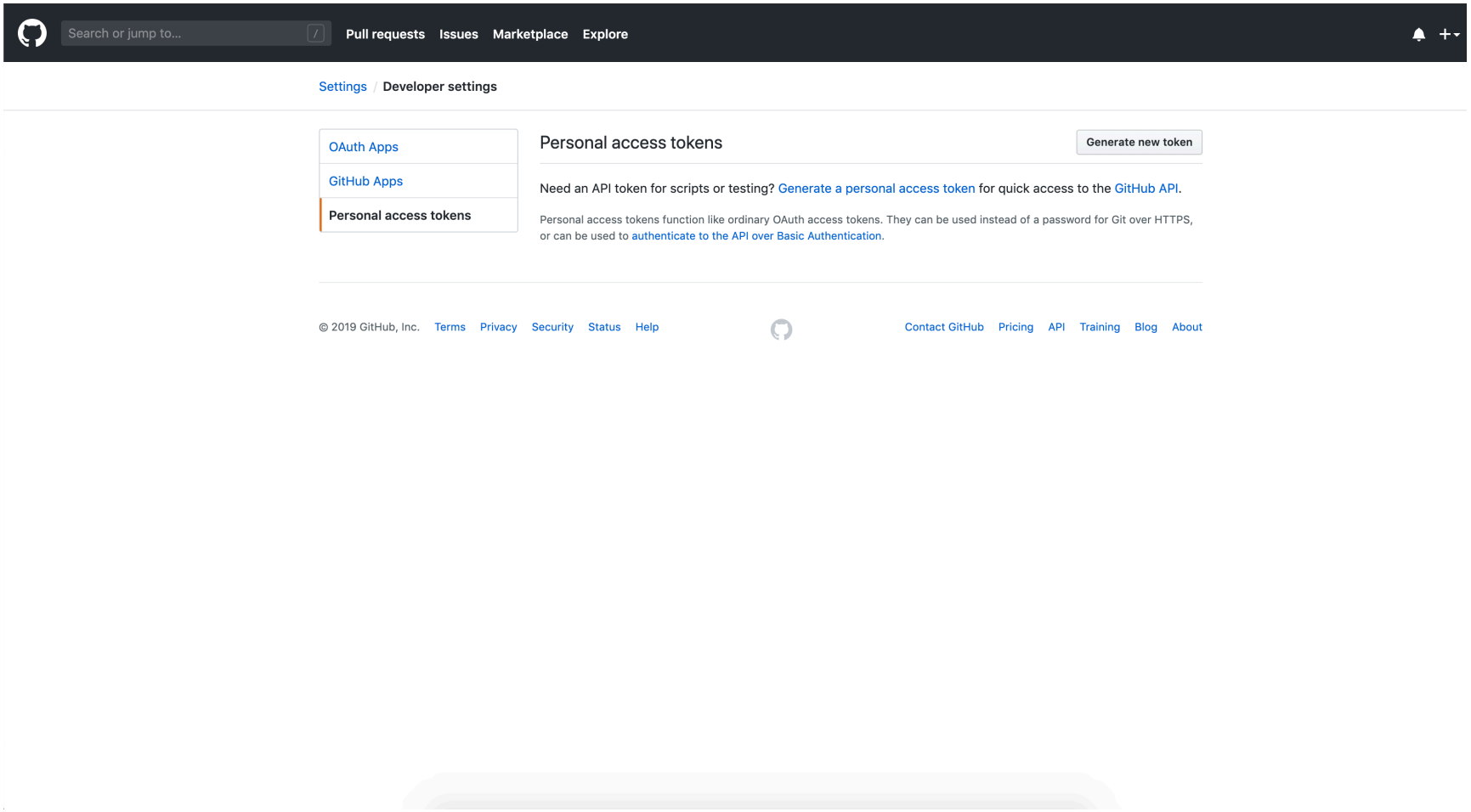
- Login to github (not your gitlab server) at https://github.com (or create an account if you don't have one already).
- After logging in, click on your user icon on the top right and select Settings



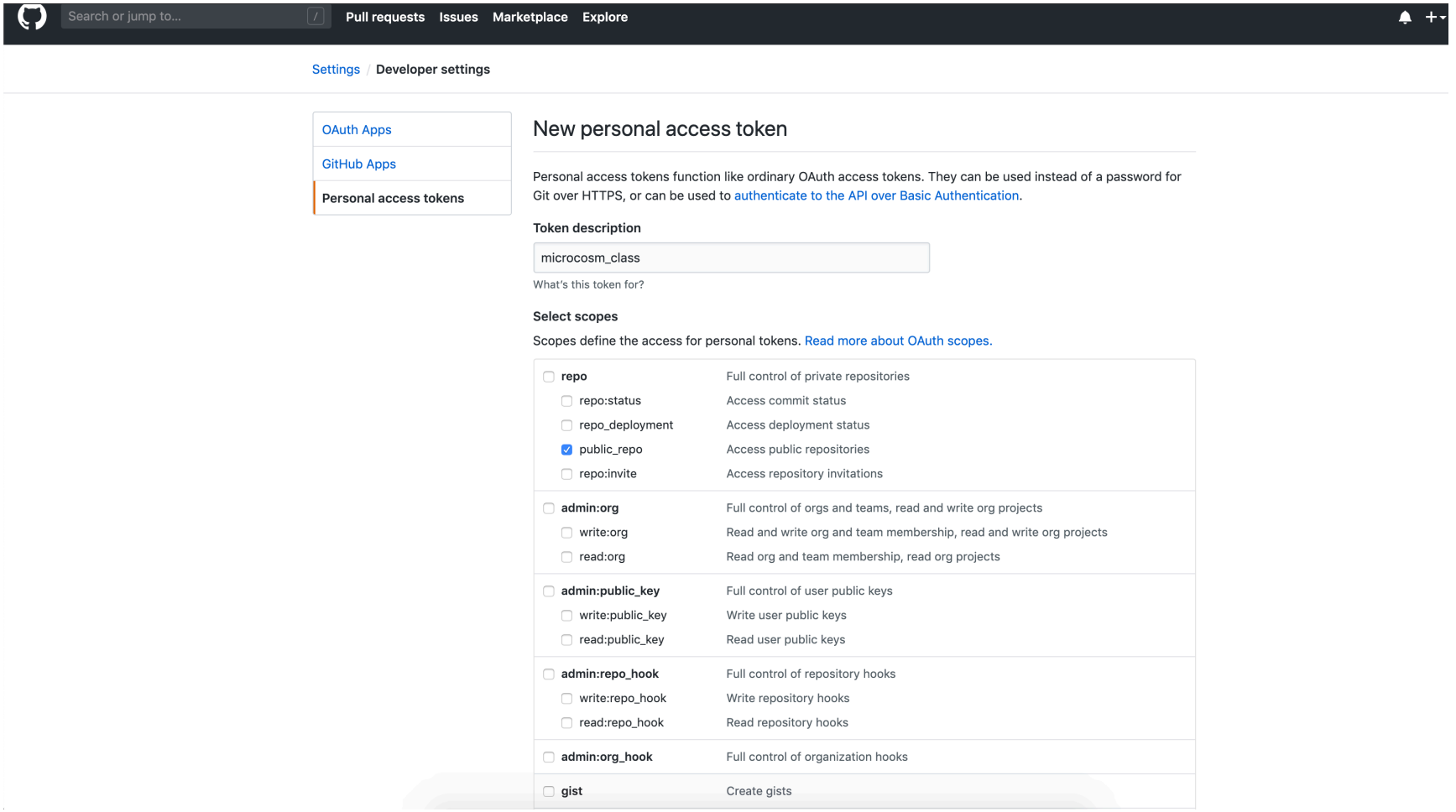
- Select Developer settings from the left menu list


Personal settings
Profile
Account
Emails
Notifications
Billing
SSH and GPG keys
Security
Sessions
Blocked users
Repositories
Organizations
Saved replies
Applications
Developer settings
Public profile
Name
Public email
Select a verified email to display
You have set your email address to private. To toggle email privacy, go to email settings and uncheck "Keep my email address private."
Bio
Tell us a little bit about yourself
You can @mention other users and organizations to link to them.
URL
Company
You can @mention your company's GitHub organization to link it.
Location
All of the fields on this page are optional and can be deleted at any time, and by filling them out, you're giving us consent to share this data wherever your user profile appears. Please see our privacy statement to learn more about how we use this information.
Update profile

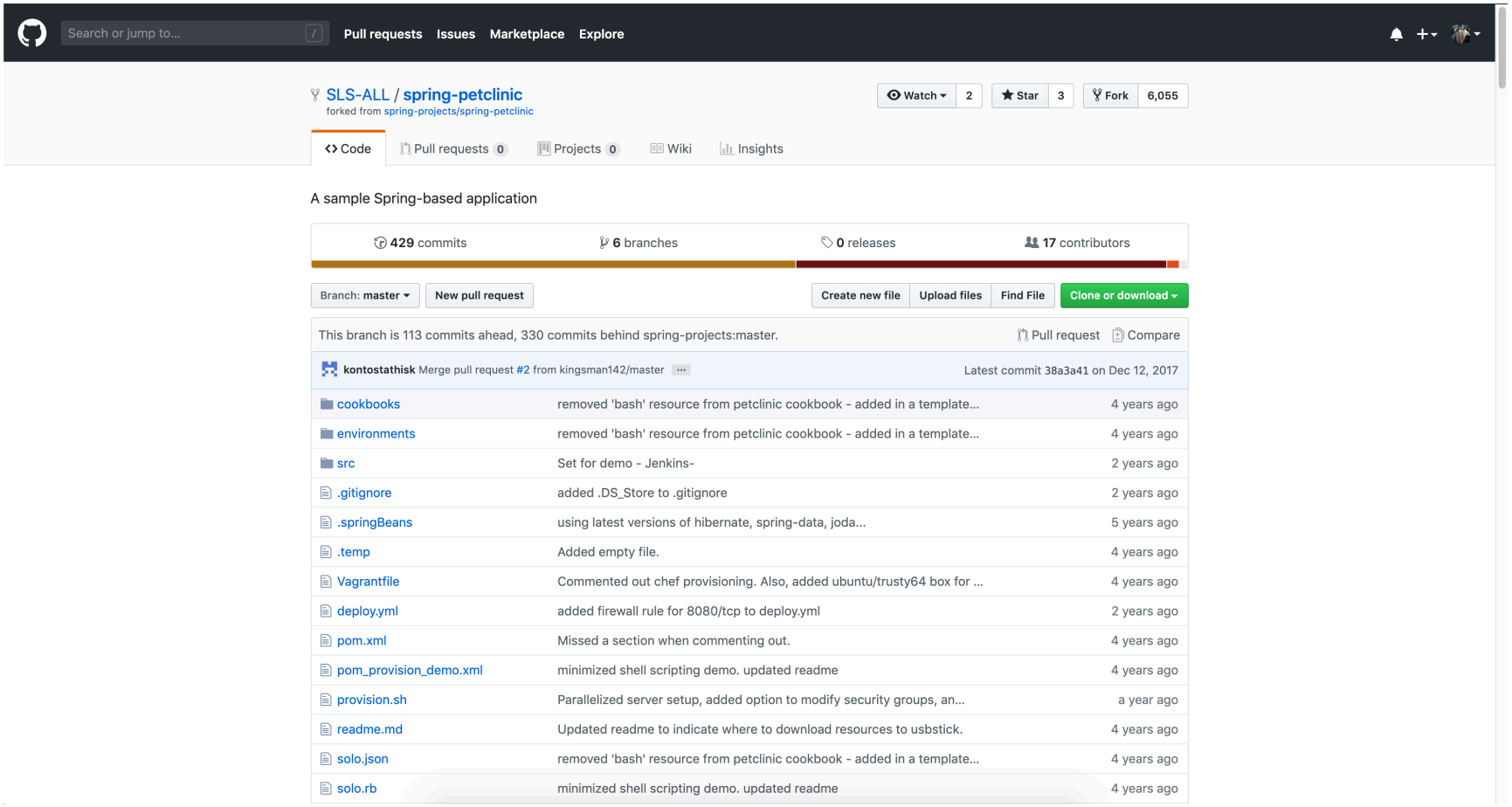
- Select Person Access Tokens
- Click Generate new token



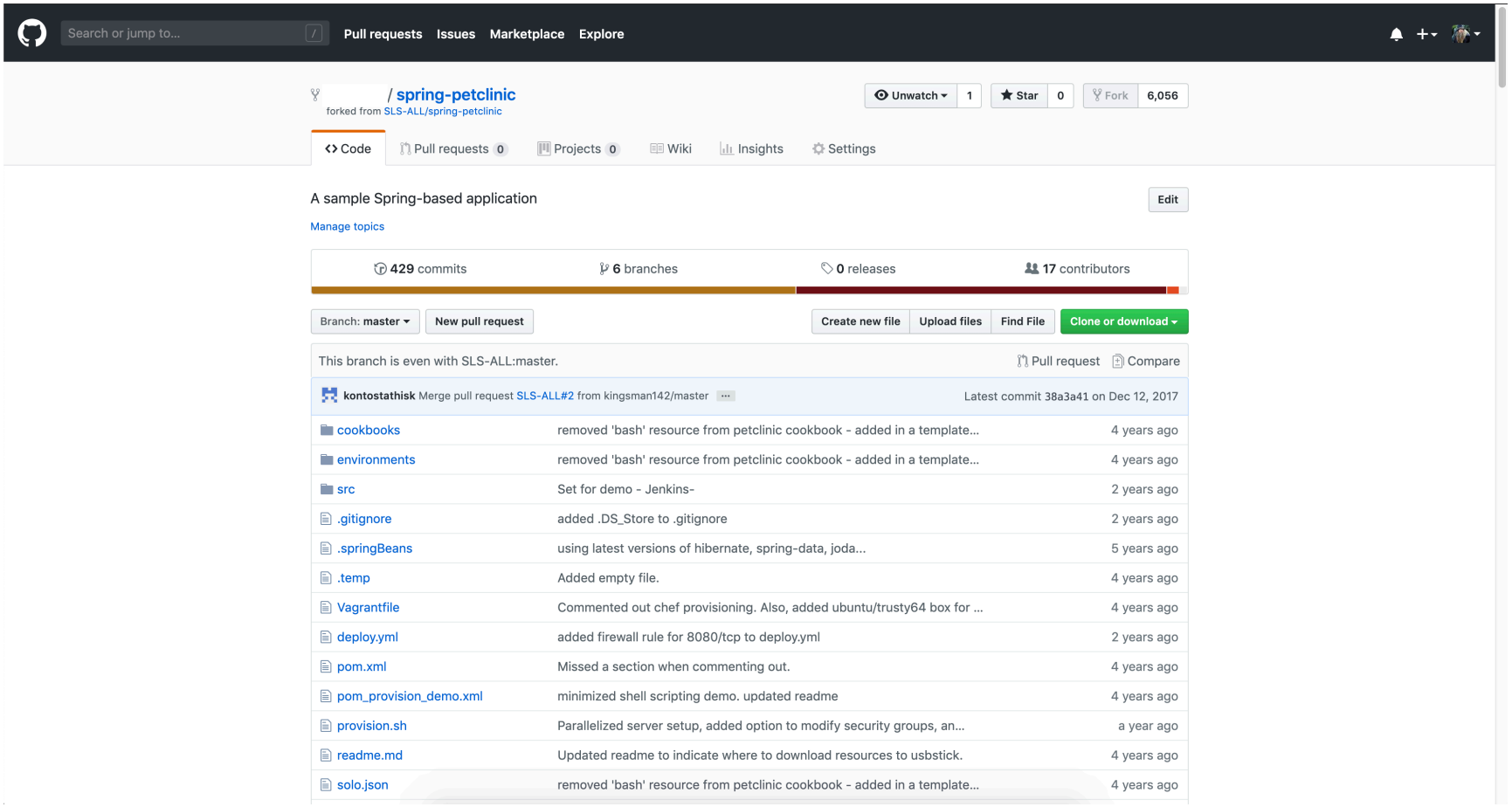
- Give your token a related description and select the public_repo checkbox
- Complete the form and **copy the generated key** to the clipboard or another location for reference in the next section



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



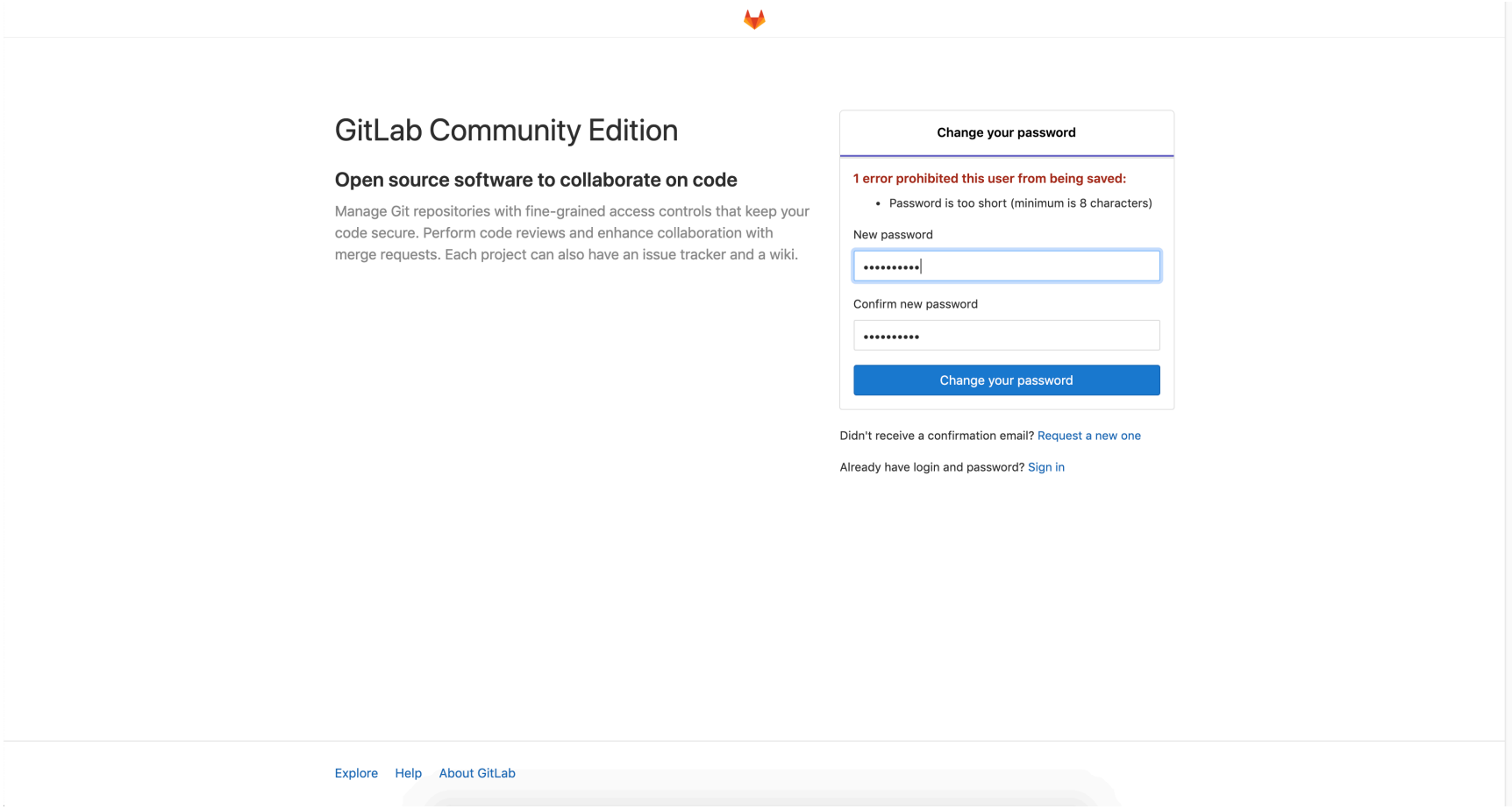
- Notice that the name at the top of the page has changed to your username and indicated the source of the fork operation



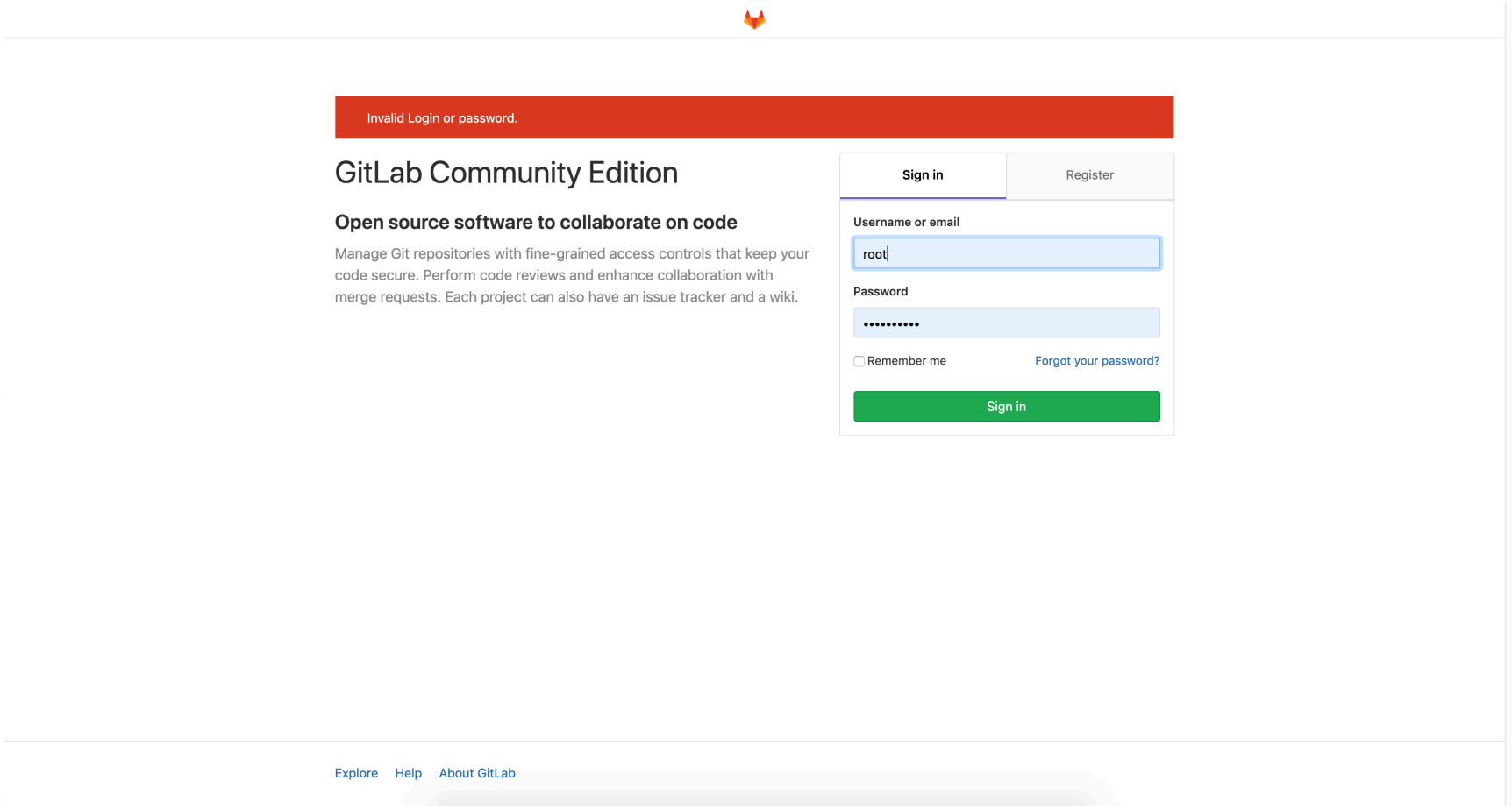
- Close Github

Gitlab Setup

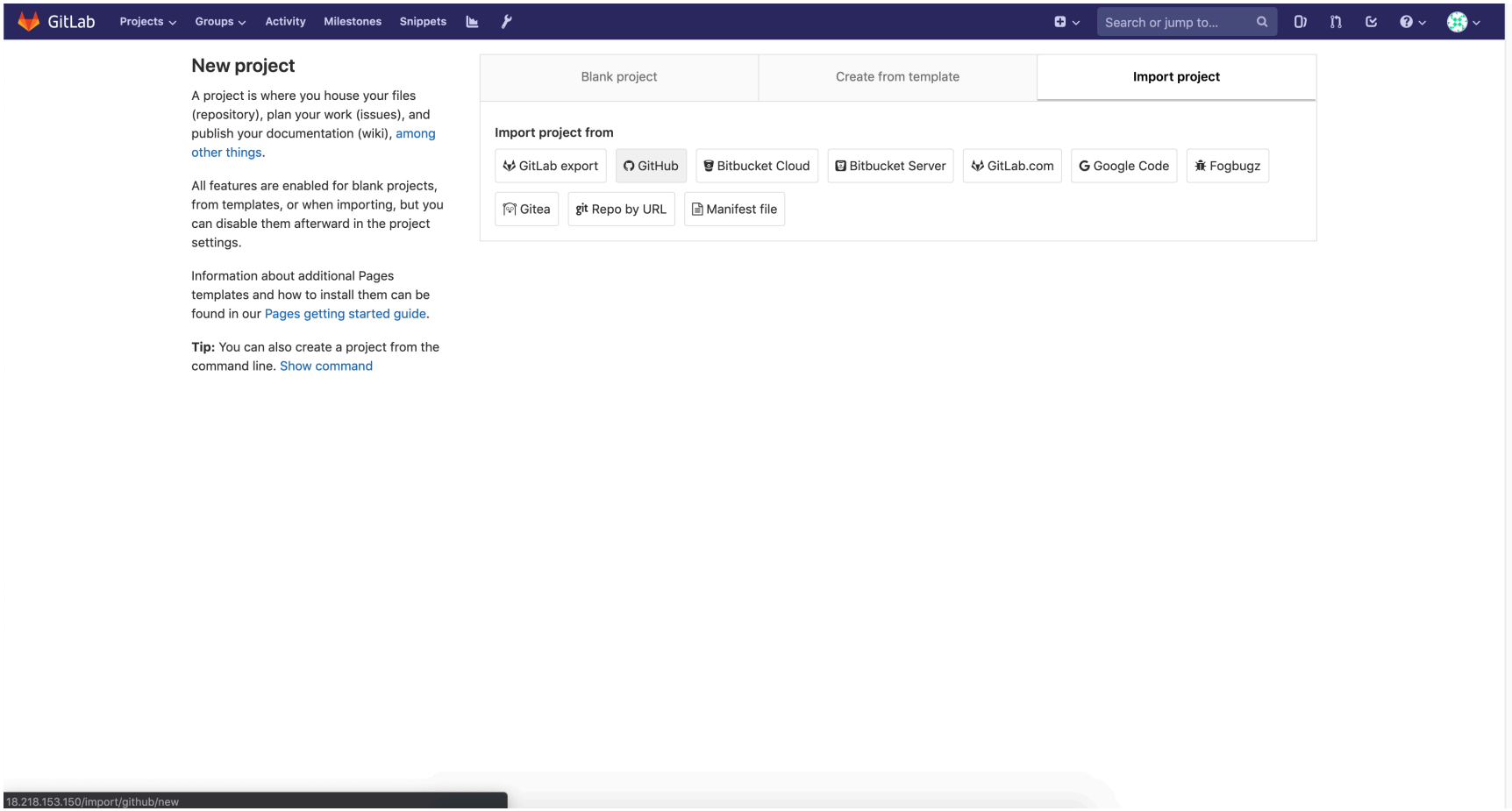
- From the Task view in AWS ECS, find the public IP of your Gitlab Task
- Change your password when prompted (at least 8 characters long)



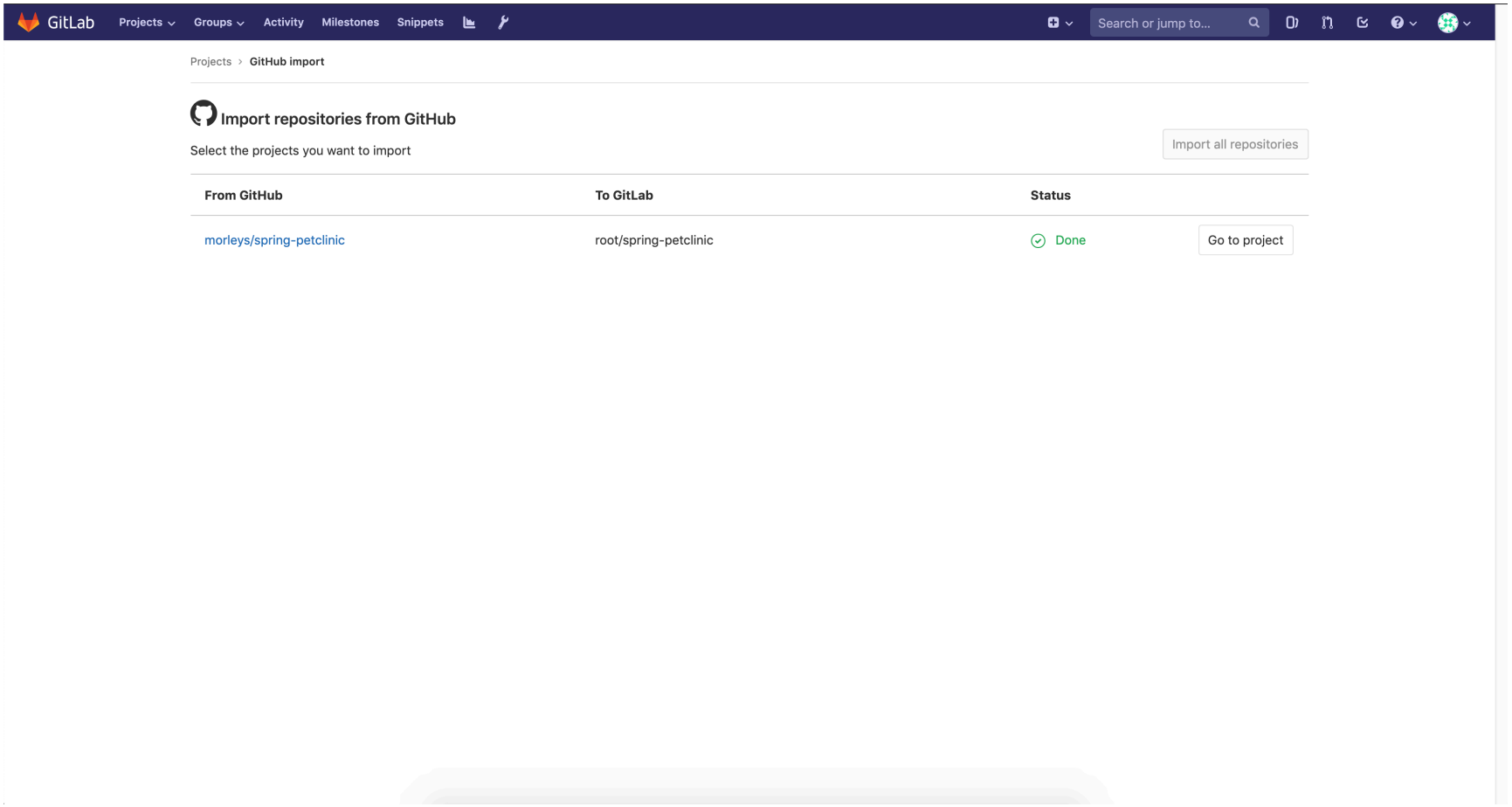
- Login with the username `root` and the password you just set



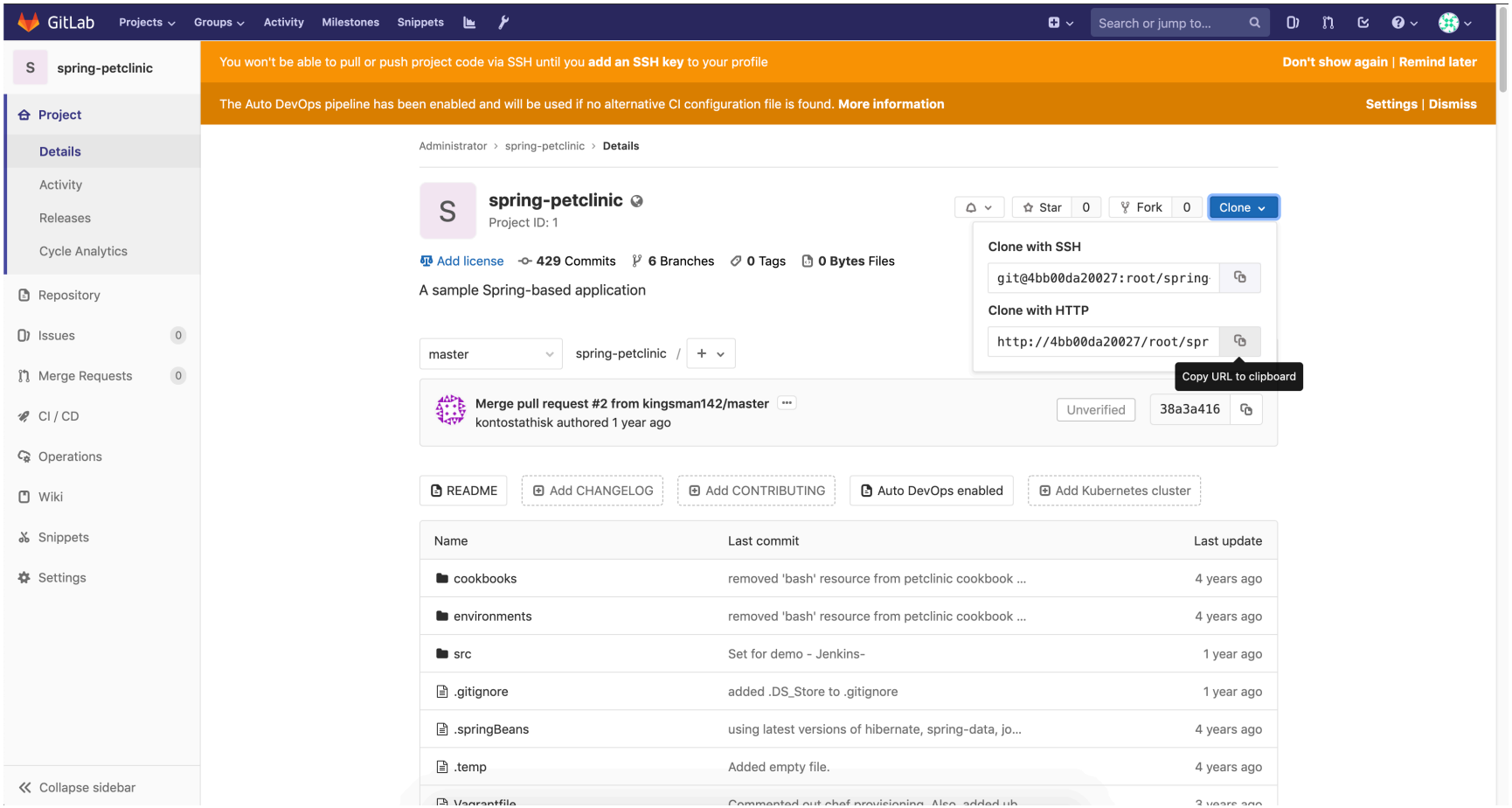
- Add `spring-petclinic` project: On GitLab dashboard, click 'new project' or 'Create a Project'
- Select the Import Project tab
- Click 'import project from github'



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



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



IMPORTANT

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

Create Maven Deployment in Jenkins

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

Lockable Resources

Credentials

New View

Build Queue

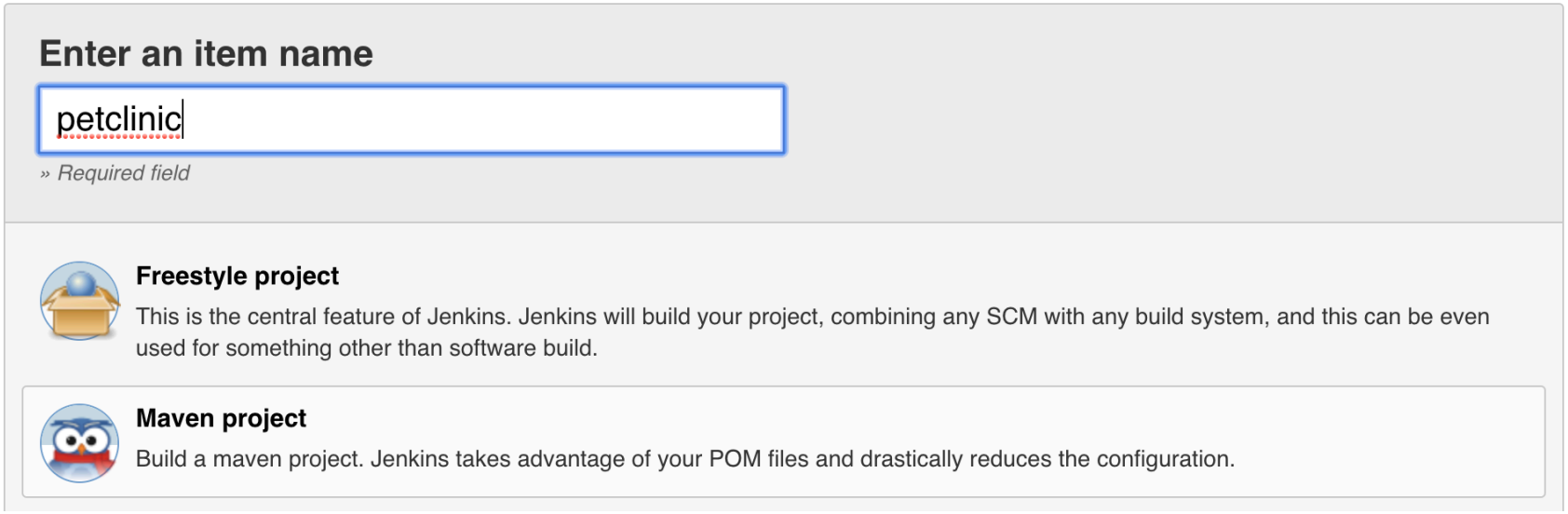
No builds in the queue.

Build Executor Status

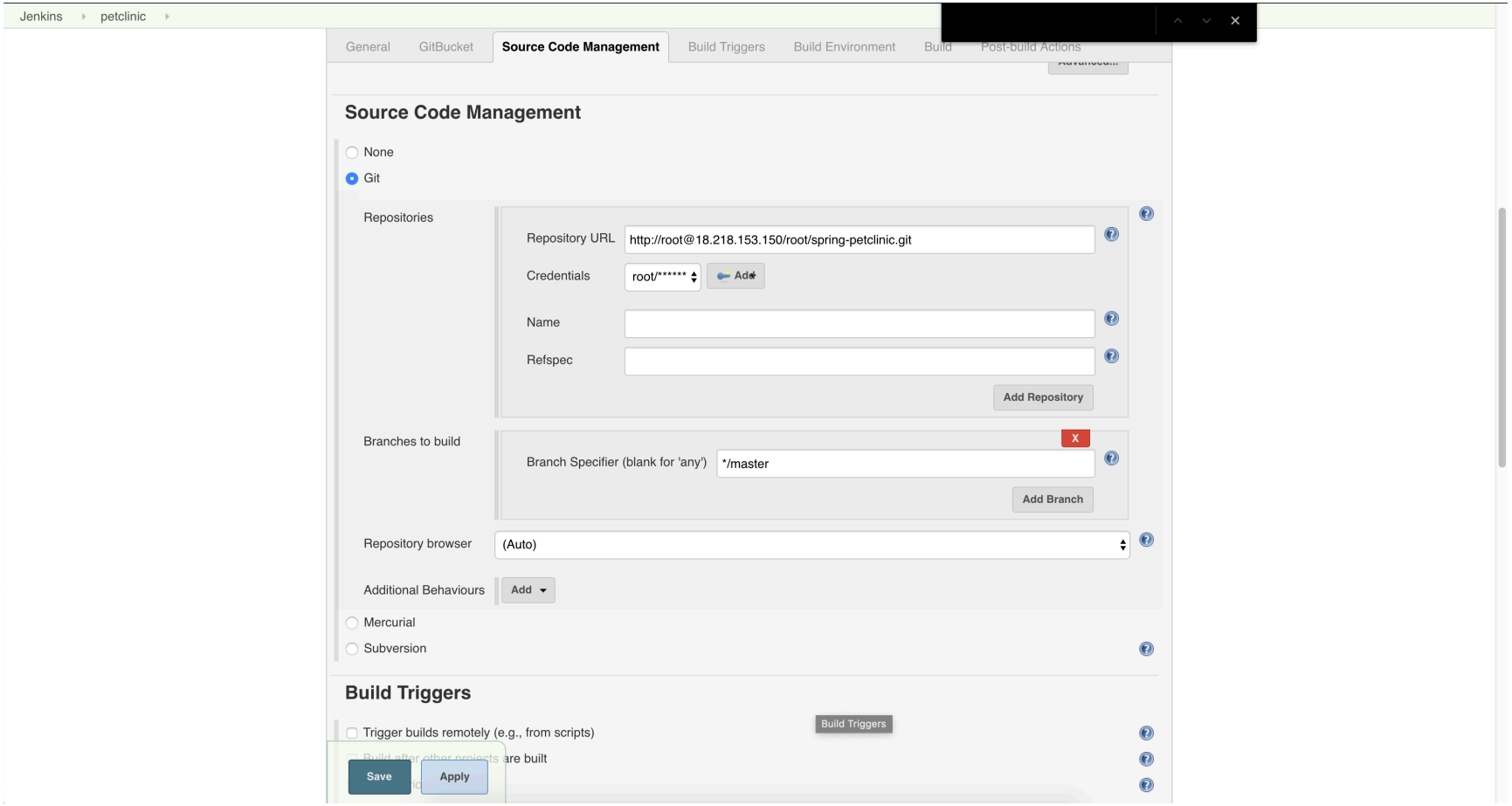
1 Idle

2 Idle

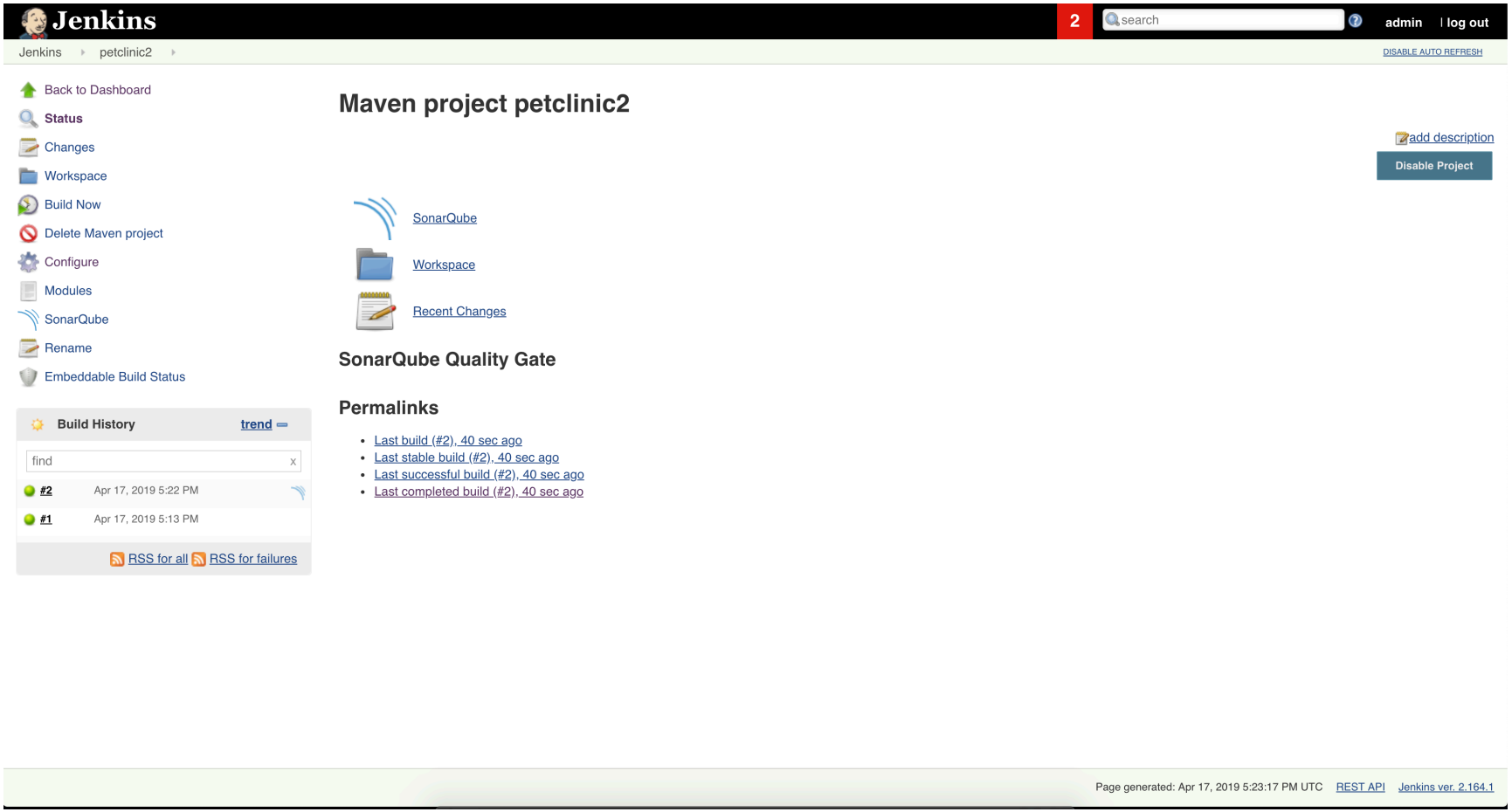
- When prompted enter the Item Name `petclinic` and select Maven project and press OK



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

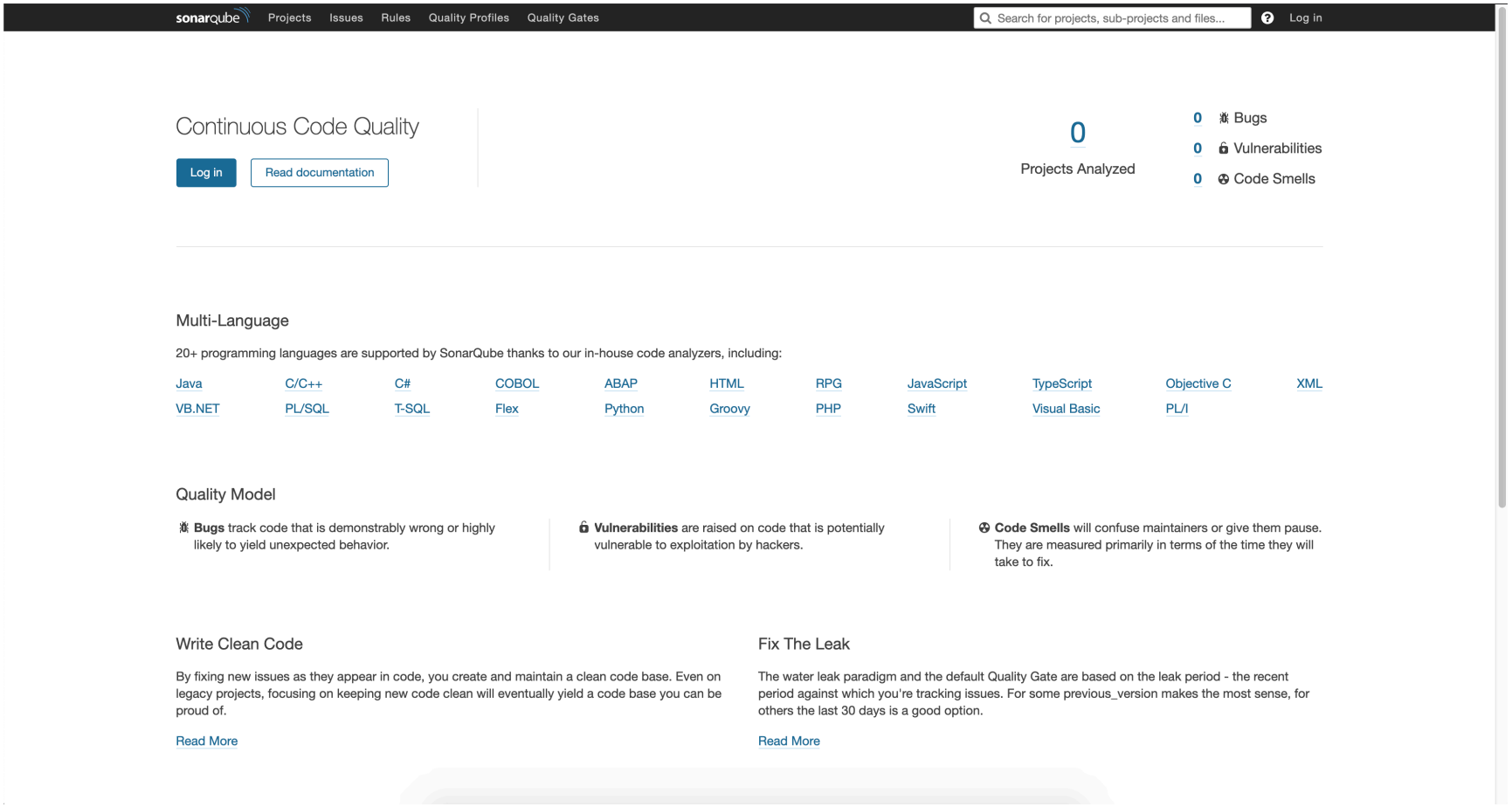


- From the project page in Jenkins select Build Now to test building

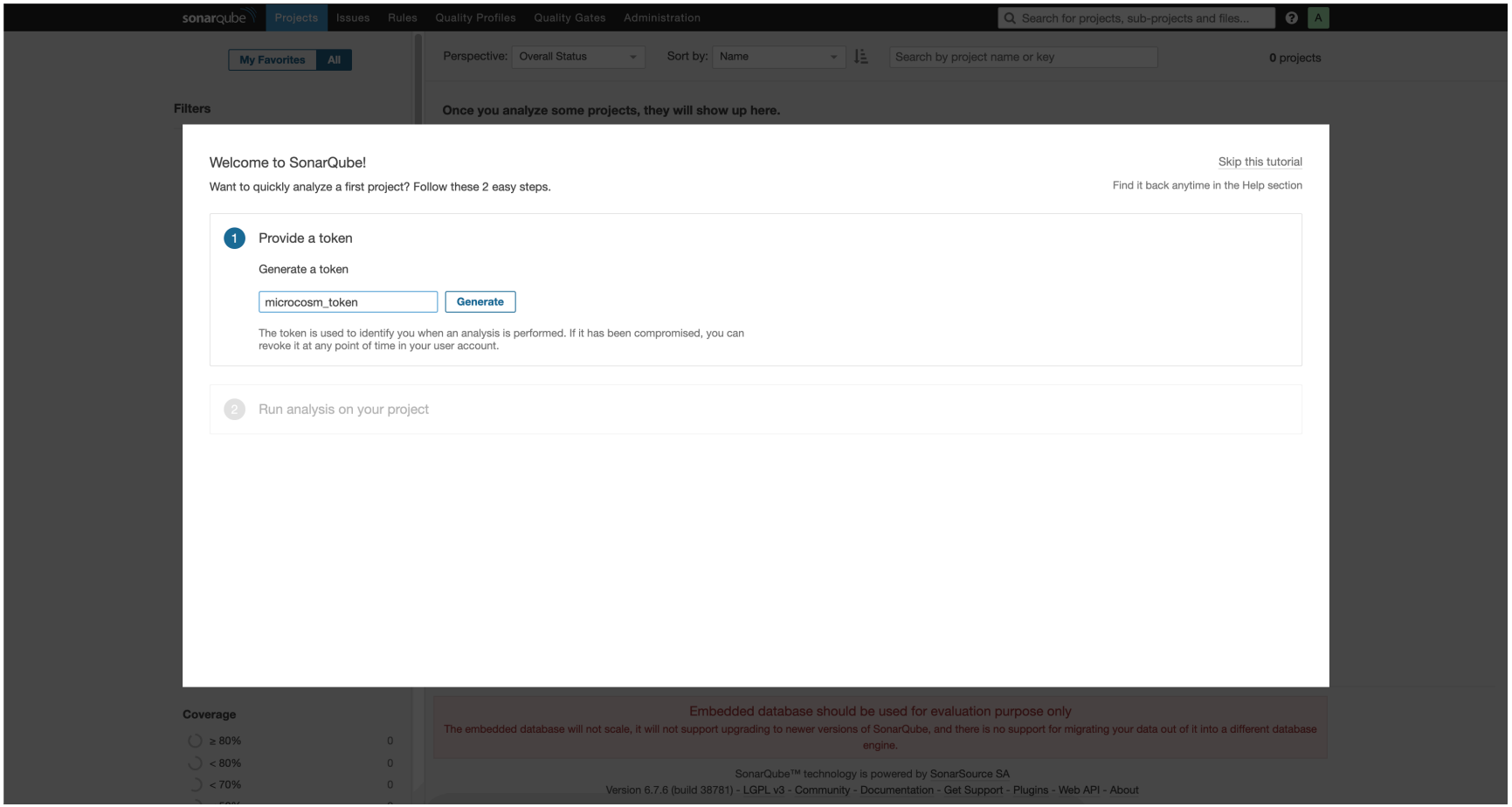


Sonarqube Setup

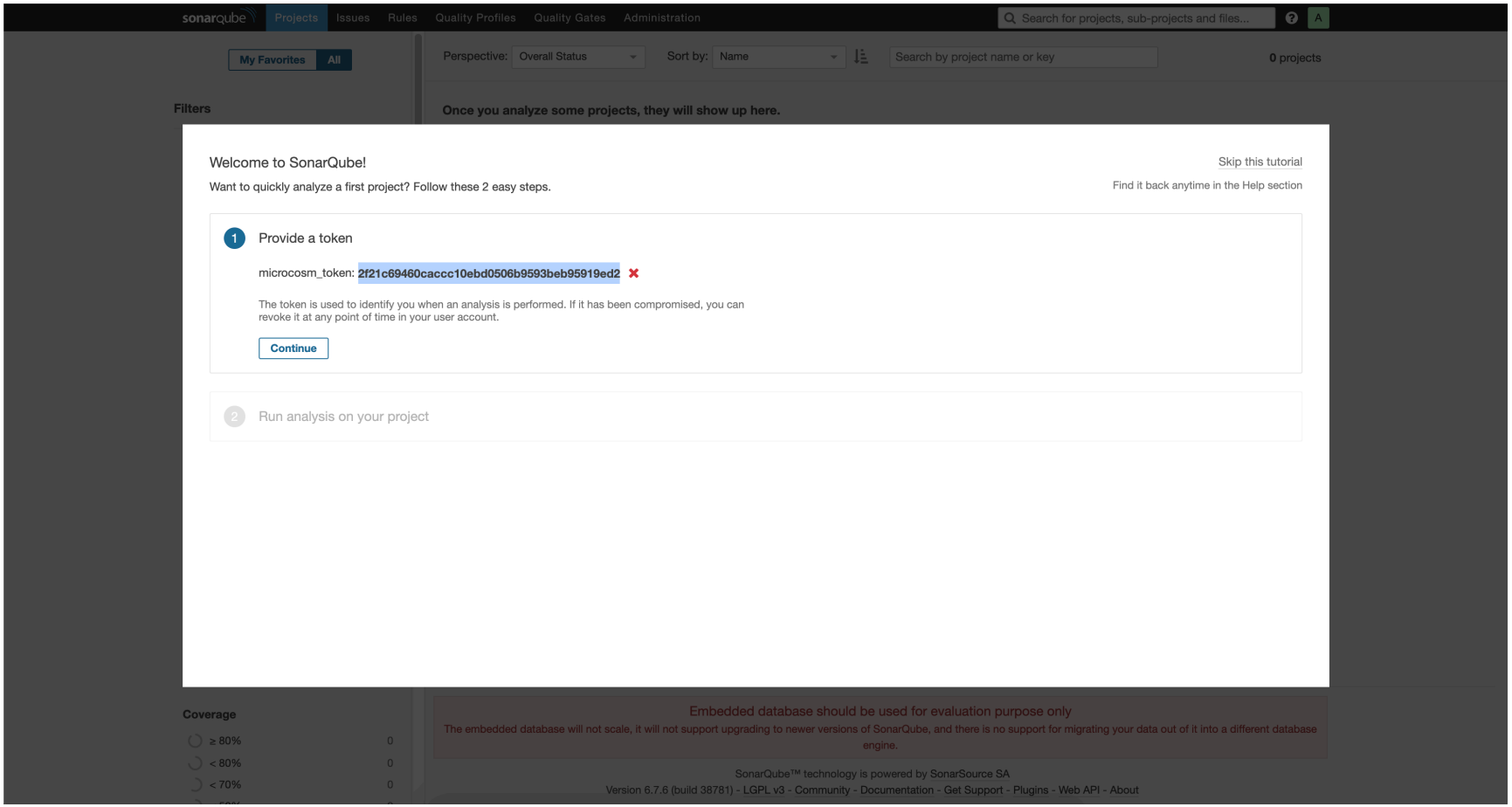
- Using the IP from the task view in AWS ECS, navigate to the sonarqube service in the browser at port 9000 (eg 44.44.44.44:9000)
- Login with the default username and password `admin:admin`



- Enter a name for you token generation

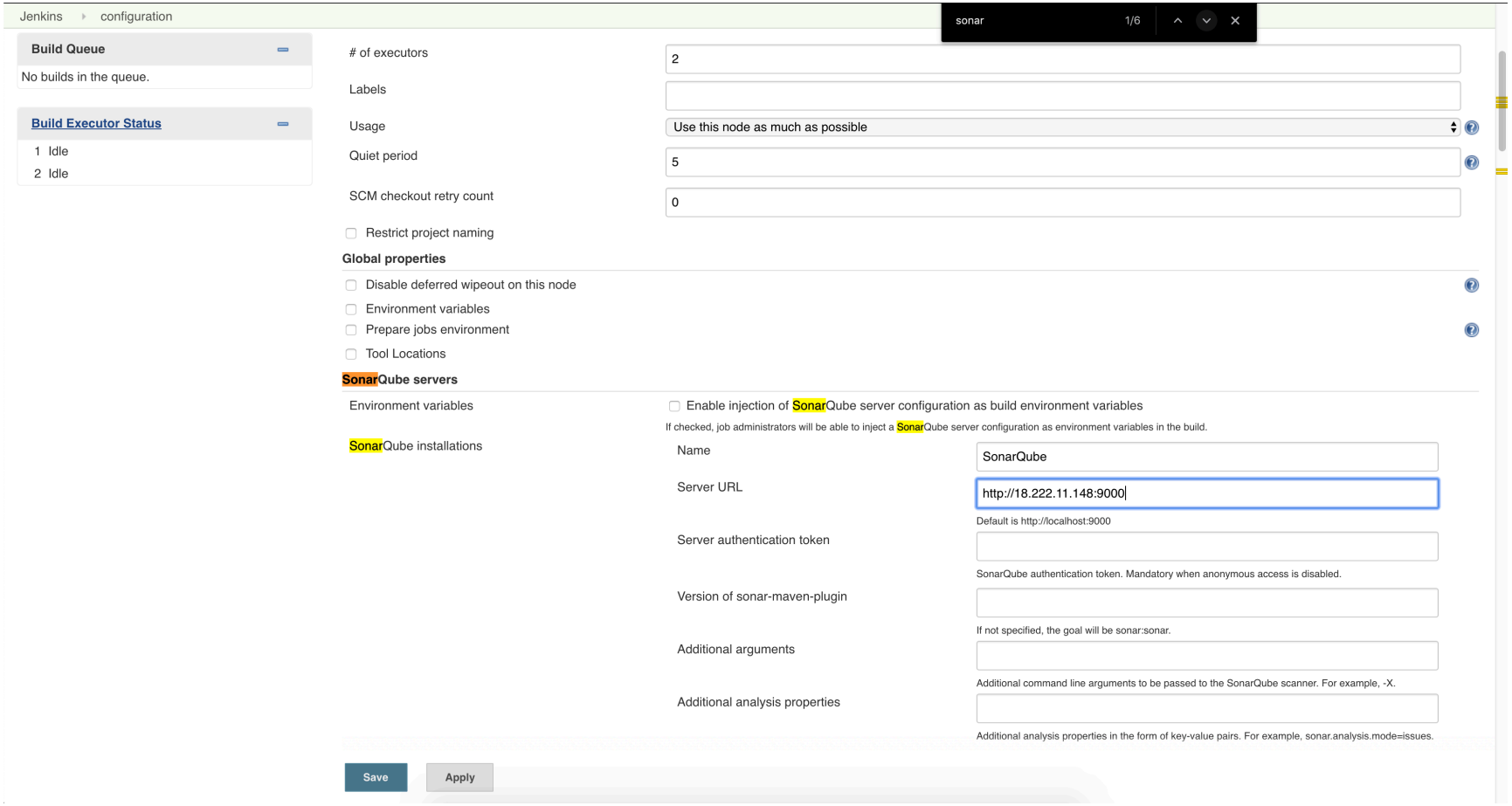


- Copy the generated token and press Continue
- Select Java, then Maven when prompted.
- Press Finish Tutorial button or Skip this Tutorial in the top right corner



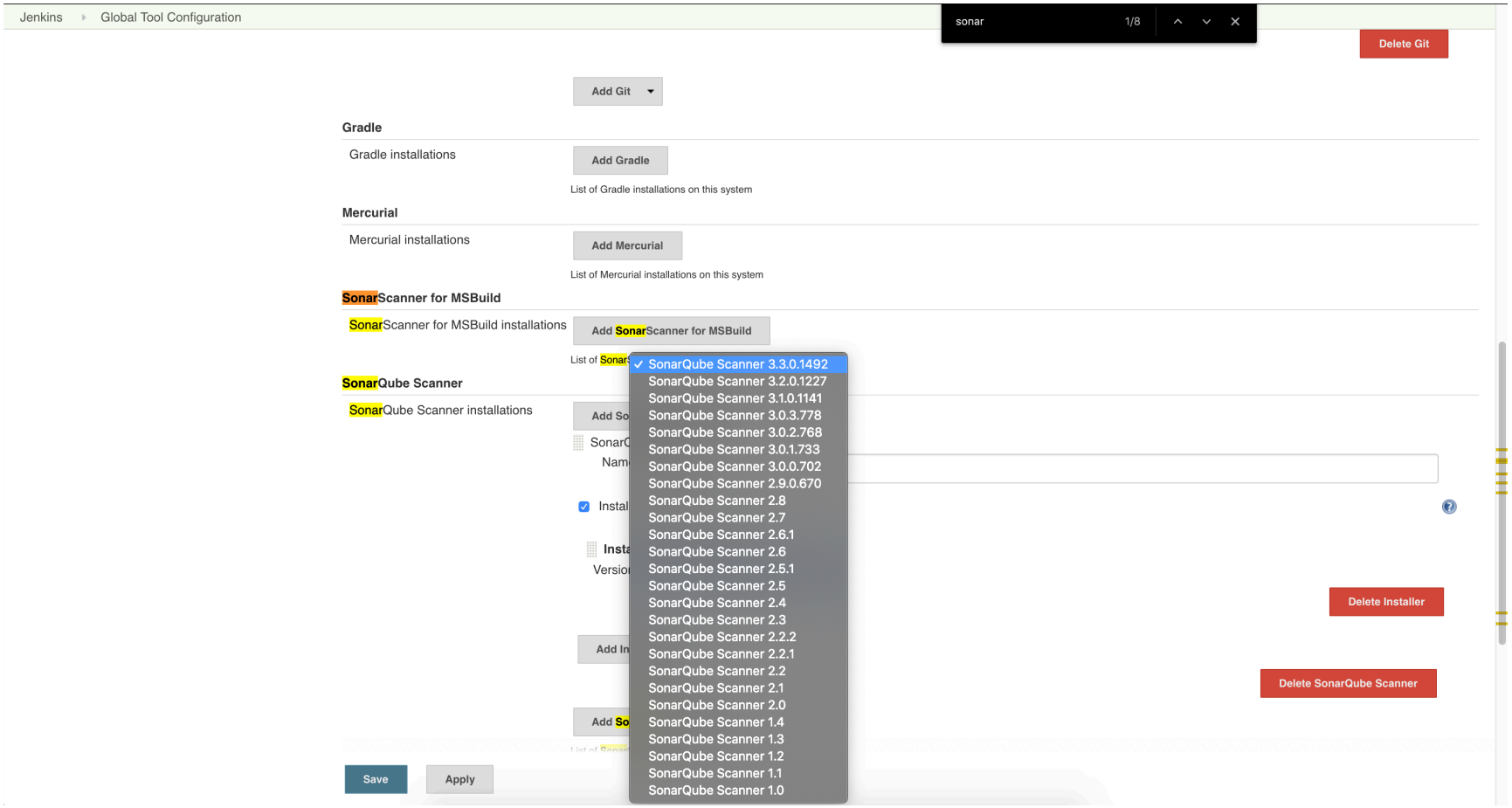
- Return to Jenkins
- From Manage Jenkins -> Under Configure System, scroll down to SonarQube Servers
- Select Add SonarQube
- Enter the name SonarQube

- Paste you server authentication token
- Enter the url inclusing port of your SonarQube server
- Press APPLY, then SAVE



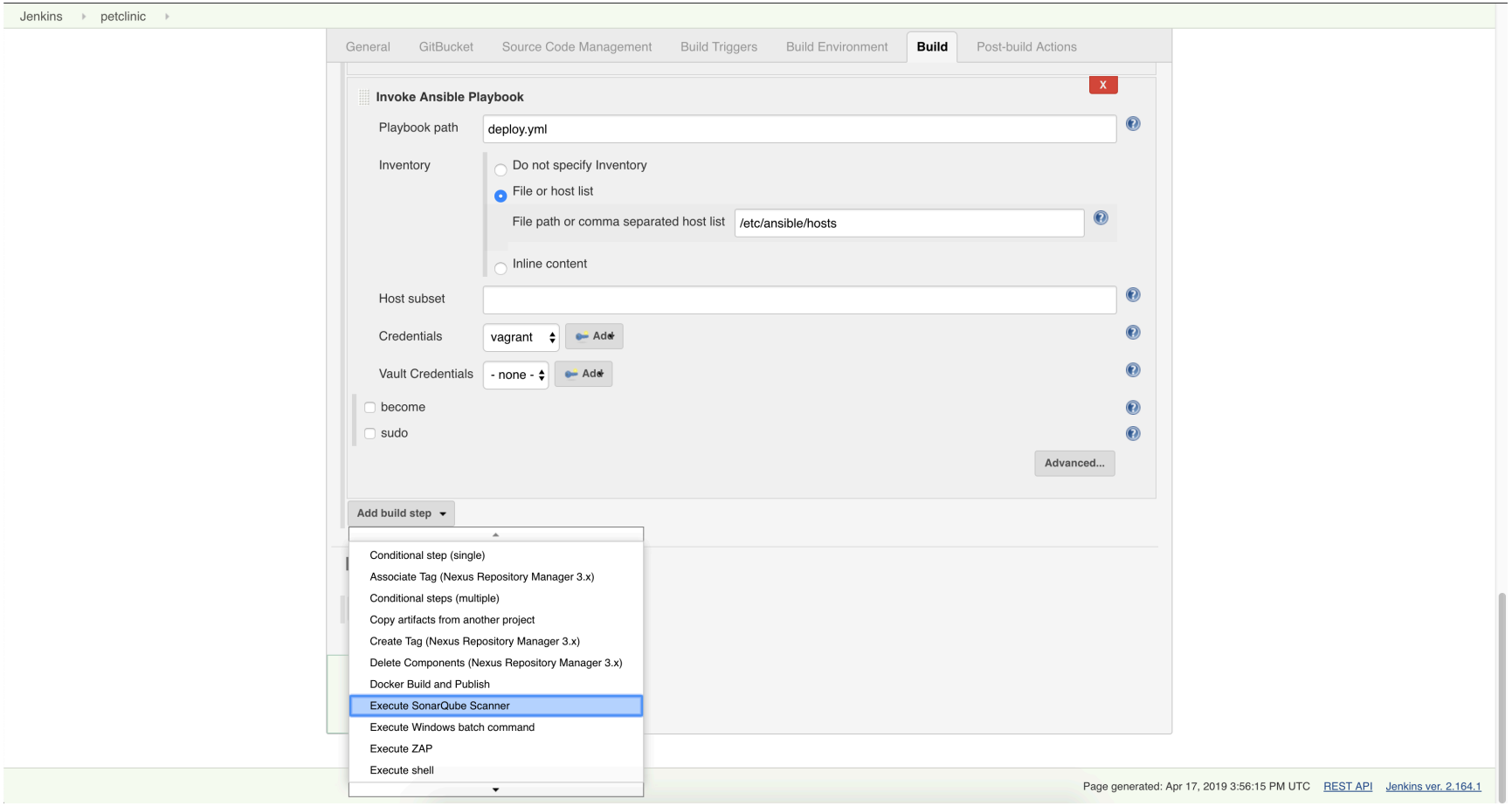
- Go to Manage Jenkins -> Global Tool Configuration
- Under SonarQube Scanner, Select Add SonarQube Scaneer

- Enter "SonarQube" in the "Name" field
- Check "Install automatically"
- Choose the most recent version of SonarQube Scanner from the version dropdown
- Click Apply and Save



- Return to the PetClinic configuration screen

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

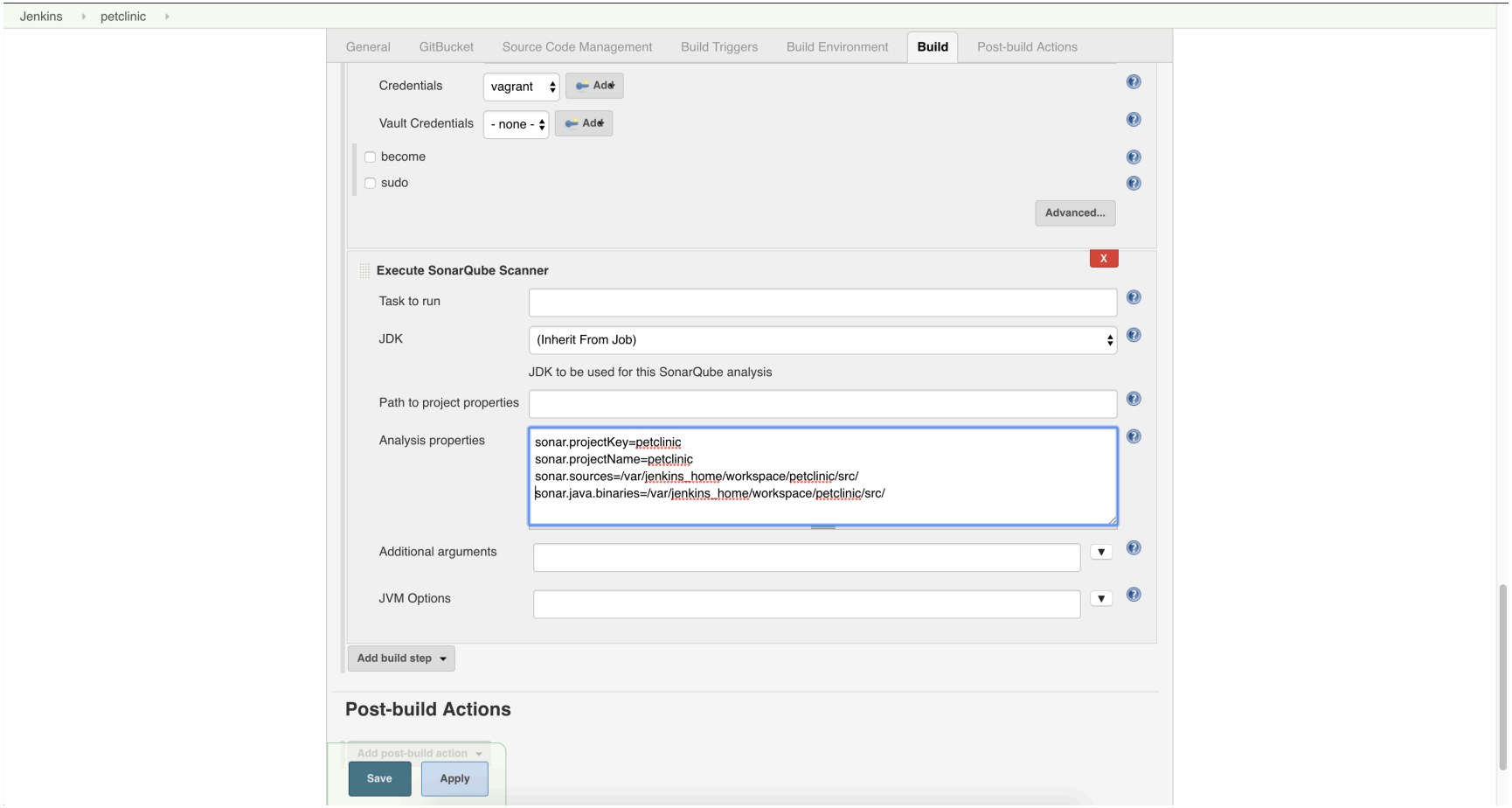


- Under "Analysis properties" enter:

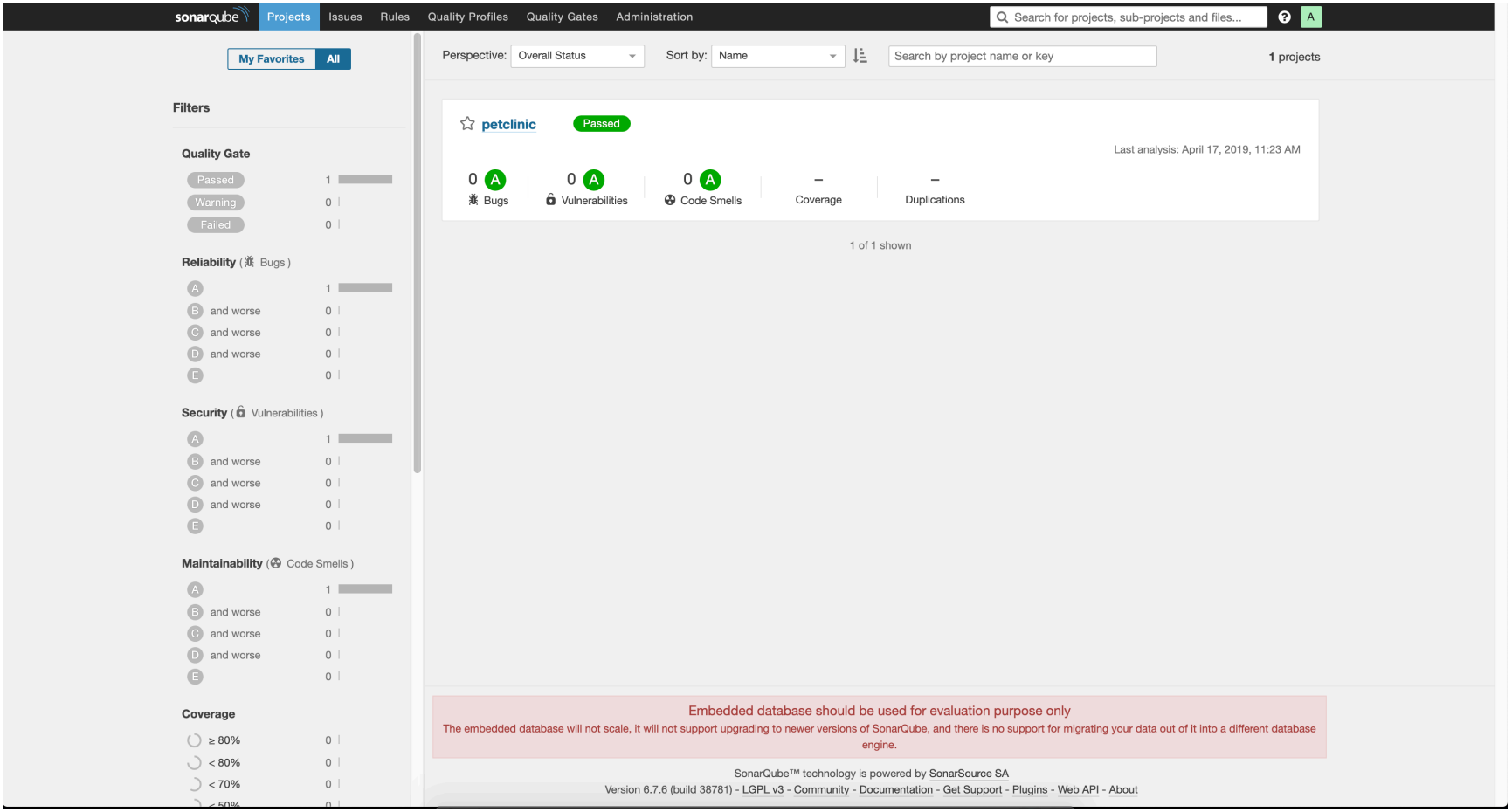
```
sonar.projectKey=petclinic
sonar.projectName=petclinic
sonar.sources=/var/jenkins_home/workspace/petclinic/src/
```

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

- Click Apply and Save



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



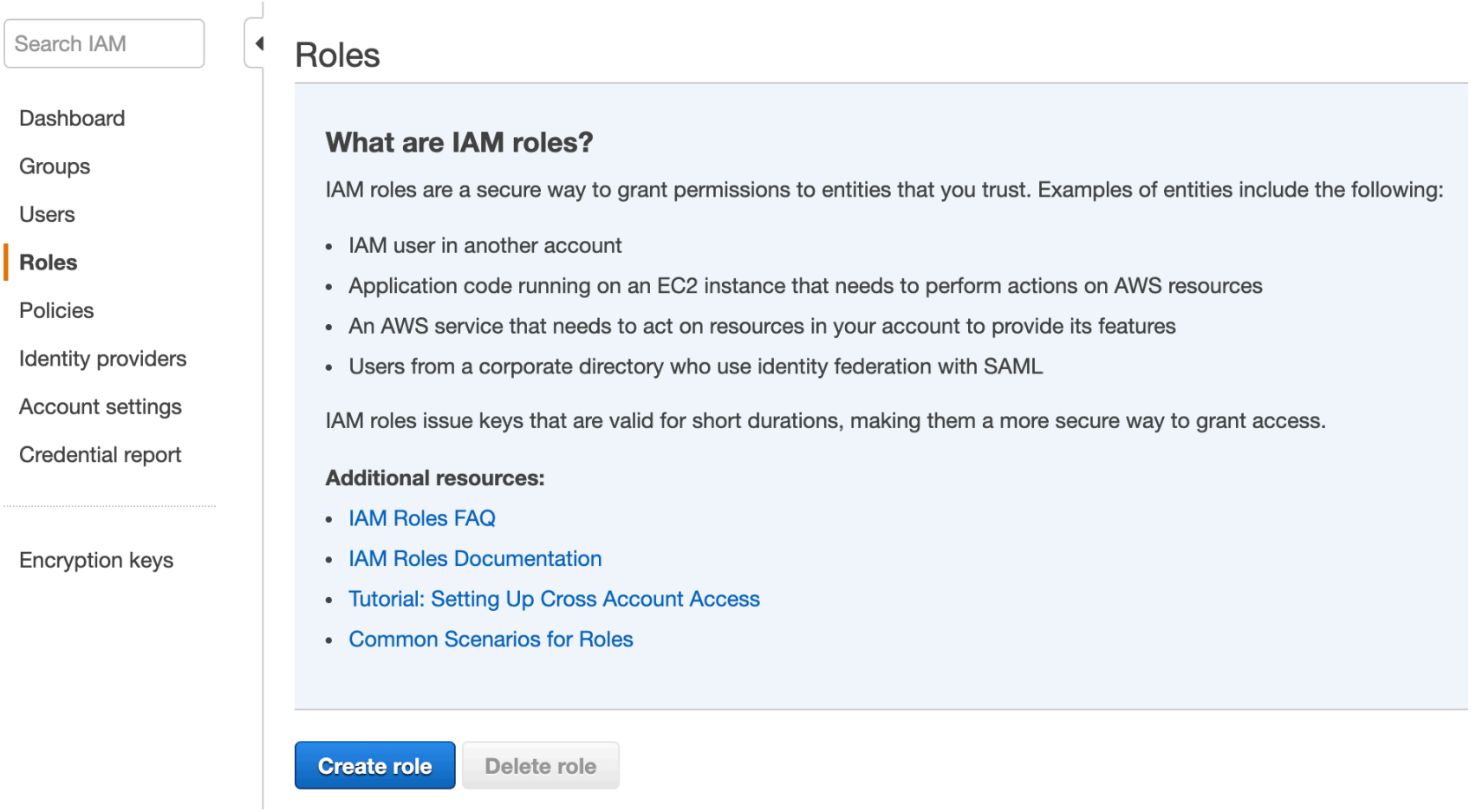
AWS EC2 Based Code Deploy

Ref. Youtube walkthrough example

Create IAM Role for EC2 Based CodeDeploy

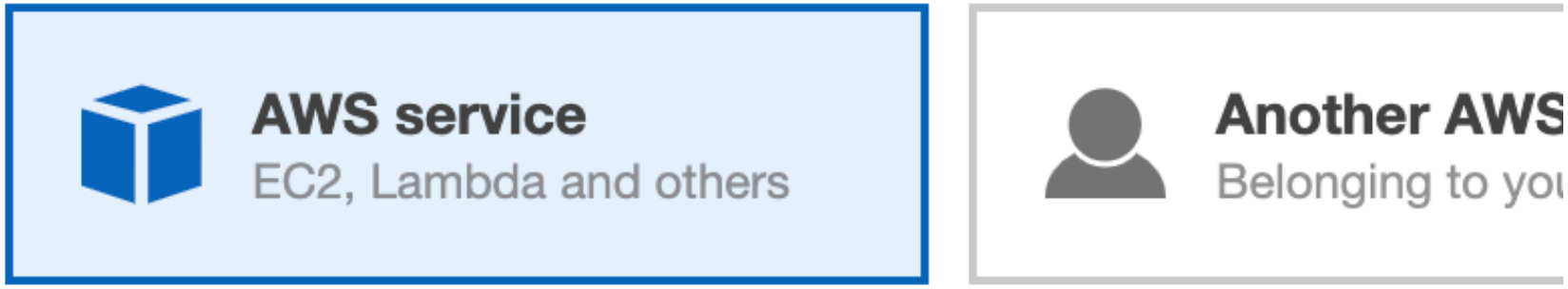
Create a role called `CodeDeployRole`

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



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

Select type of trusted entity



Allows AWS services to perform actions on your behalf. Learn

Choose the service that will use this role

EC2

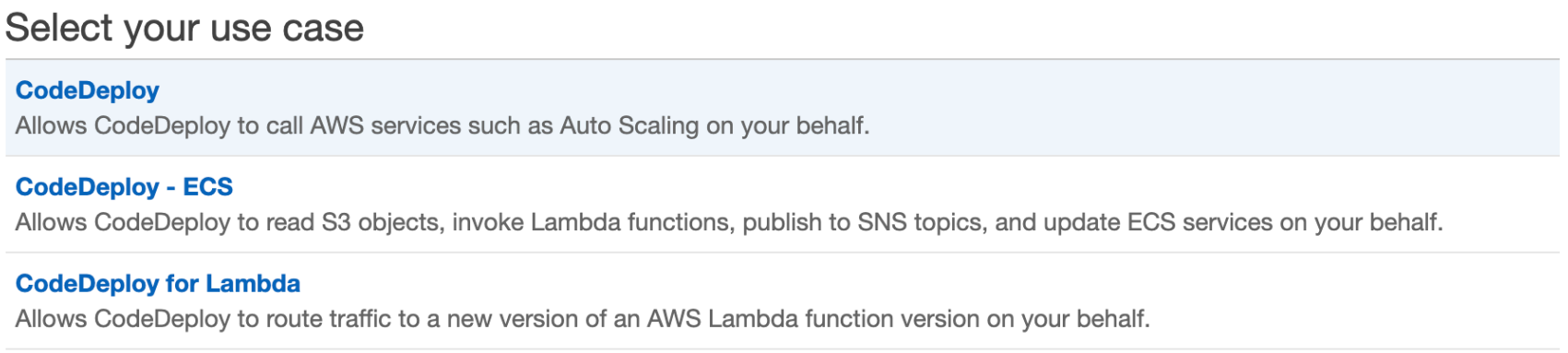
Allows EC2 instances to call AWS services on your behalf.

Lambda

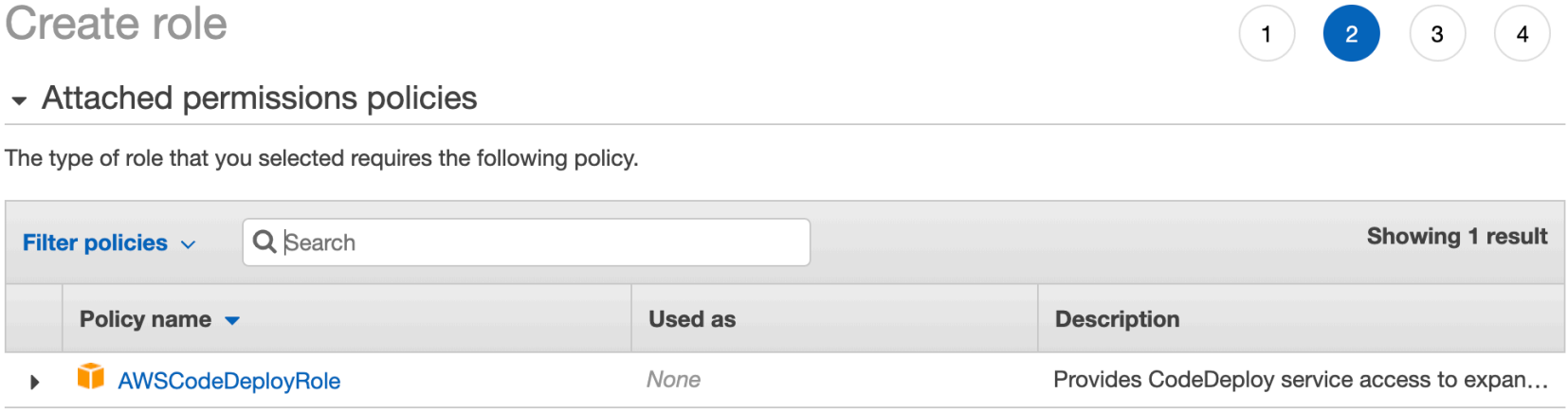
Allows Lambda functions to call AWS services on your behalf

API Gateway CodeDeploy

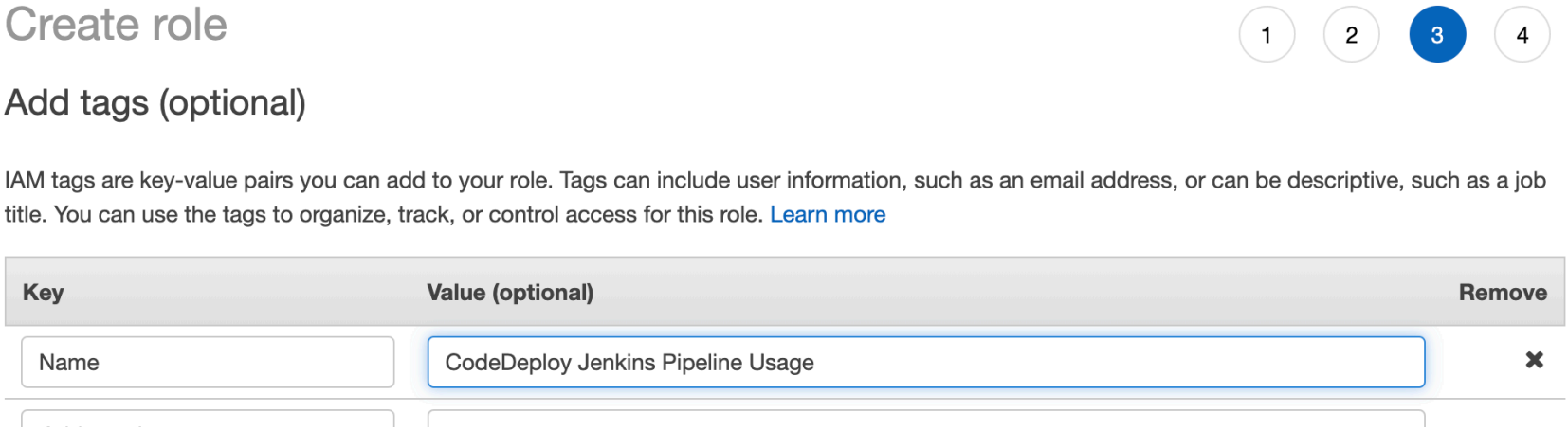
- Select CodeDeploy as your Use Case at the bottom of the page and press Next



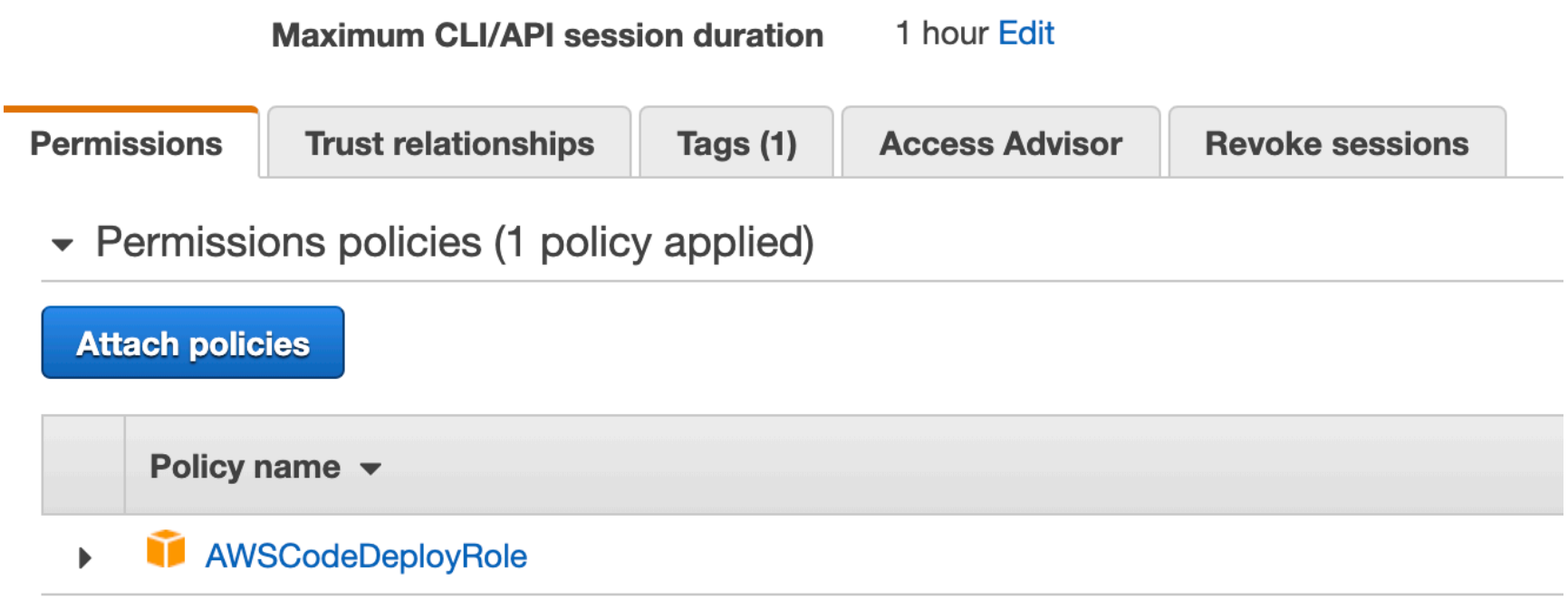
- Press Next



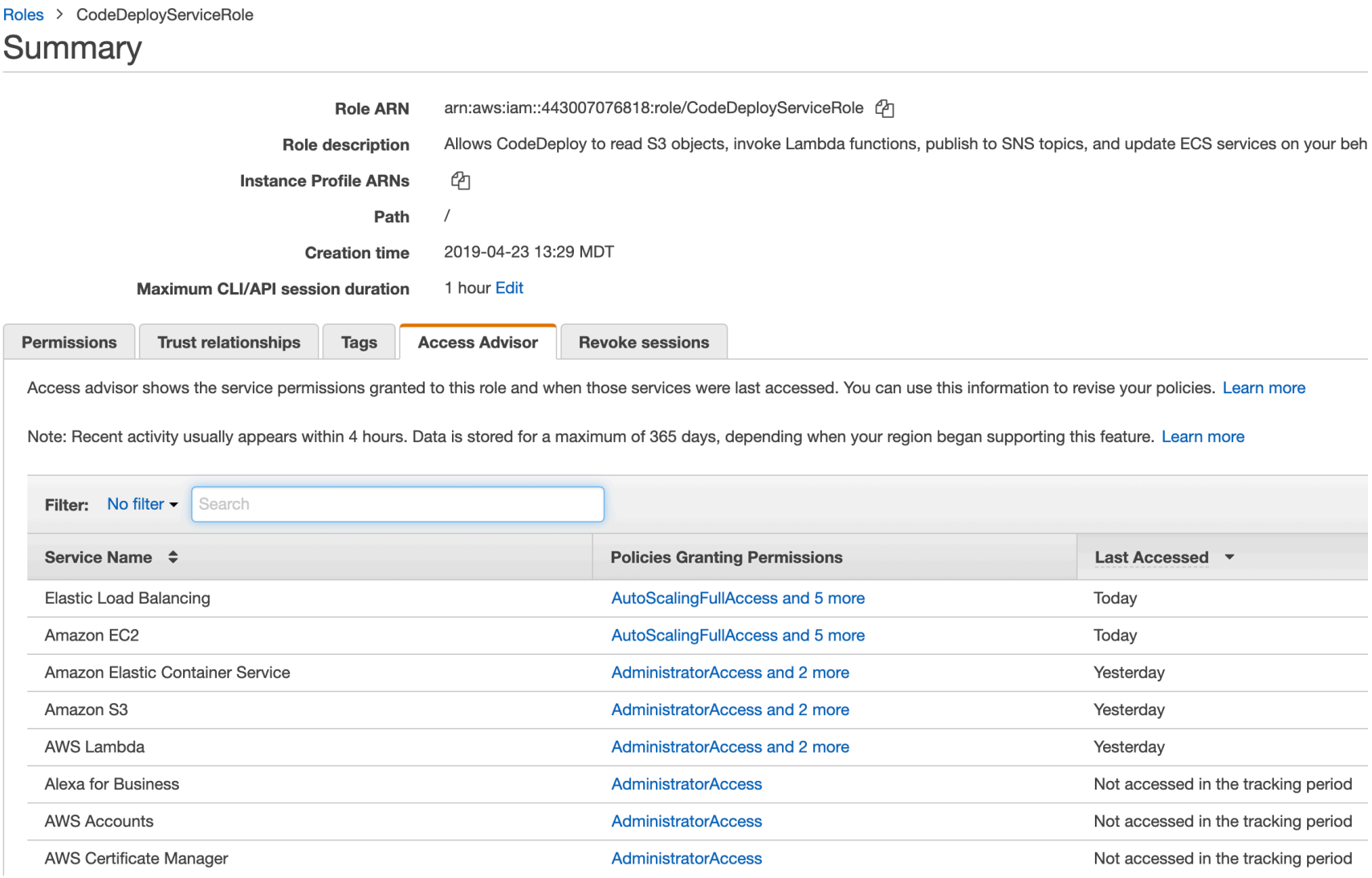
- Add Tags if desired and press Next



- 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.



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



Roles > CodeDeployServiceRole

Summary

Role ARN	arn:aws:iam::443007076818:role/CodeDeployServiceRole
Role description	Allows CodeDeploy to read S3 objects, invoke Lambda functions, publish to SNS topics, and update ECS services on your beh
Instance Profile ARNs	
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

Filter: No filter Search

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

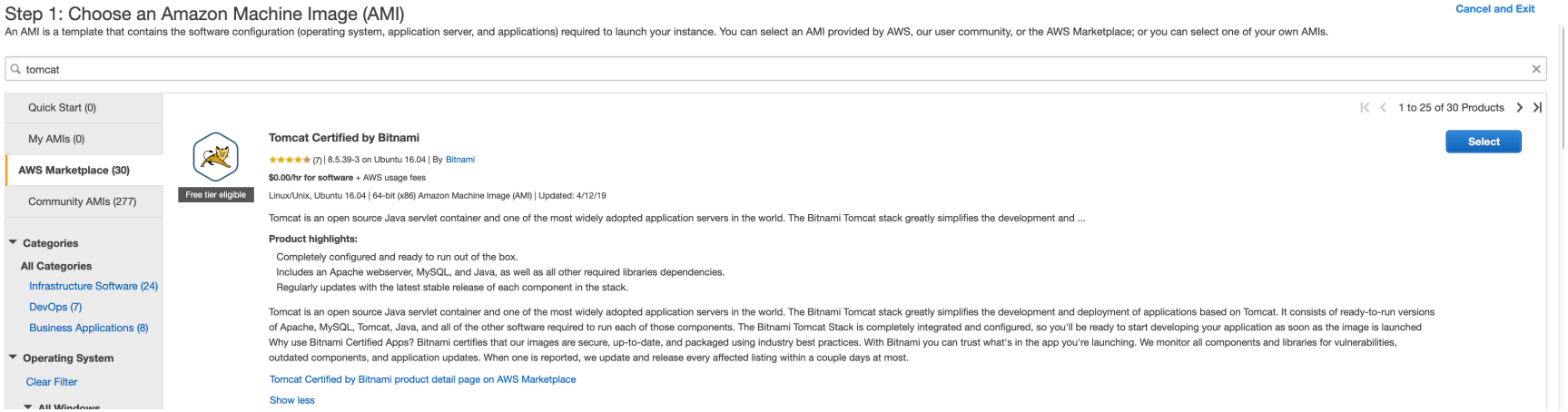
Create EC2 Artifacts

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

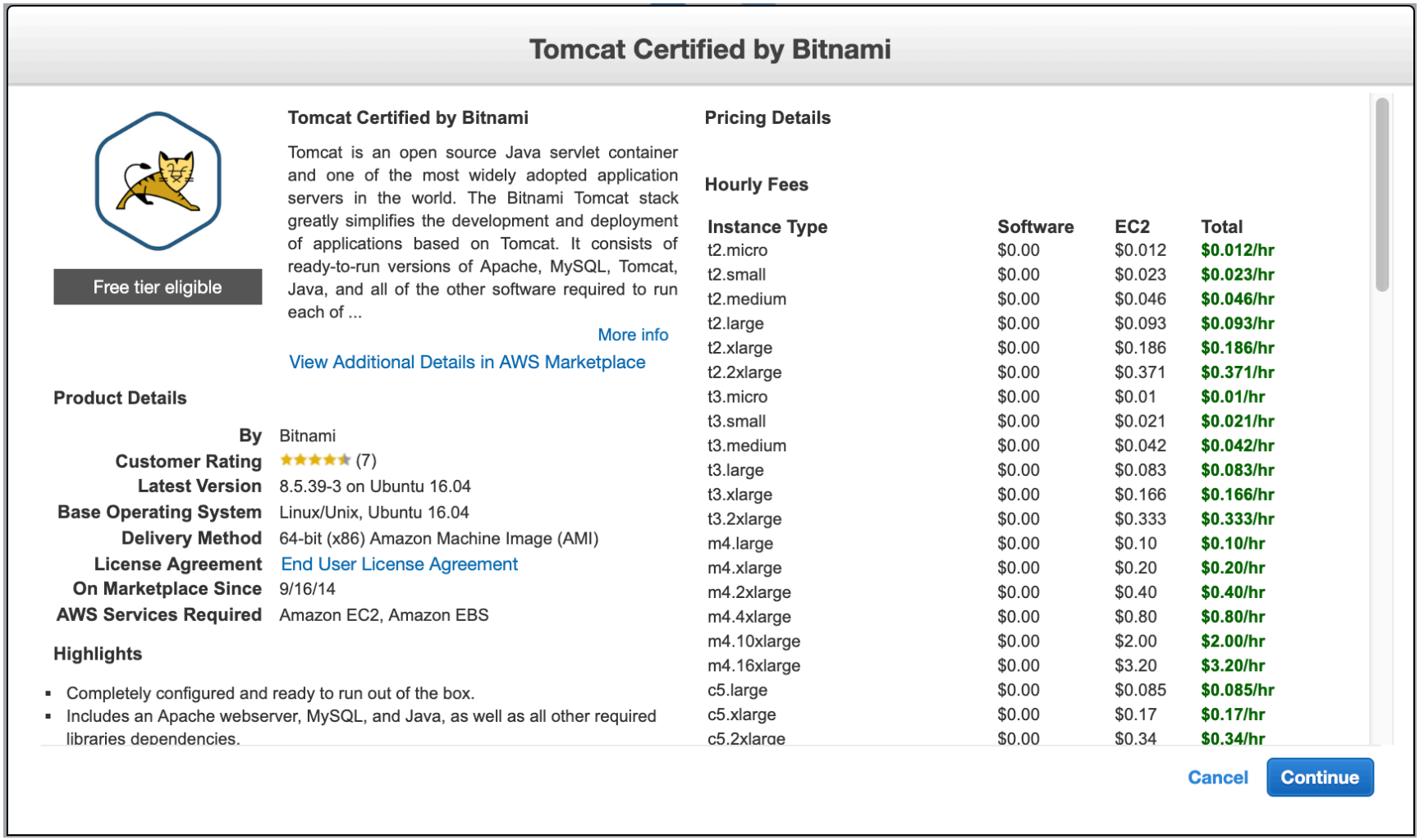
when deploying our product, we have less down time as our instances are updated in sequence.

Amazon Machine Image (AMI)

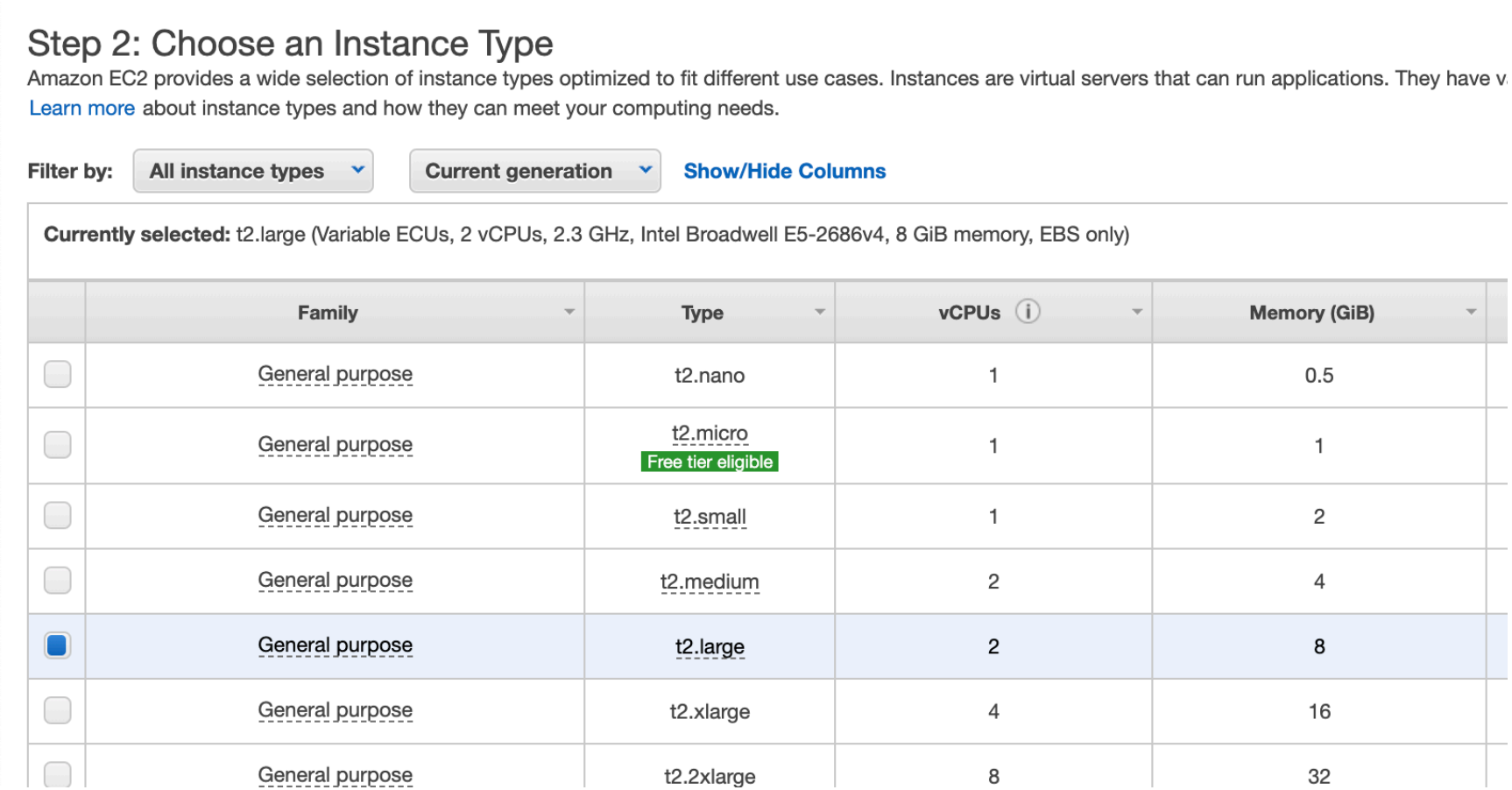
- From the EC2 Console, select Instances from the left sidebar. Then Select Launch Instance.
- Given that our application is a Java Web application, in search bar, search for Tomcat. From the AWS Marketplace select the Tomcat Certified by Bitnami image.



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



- 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

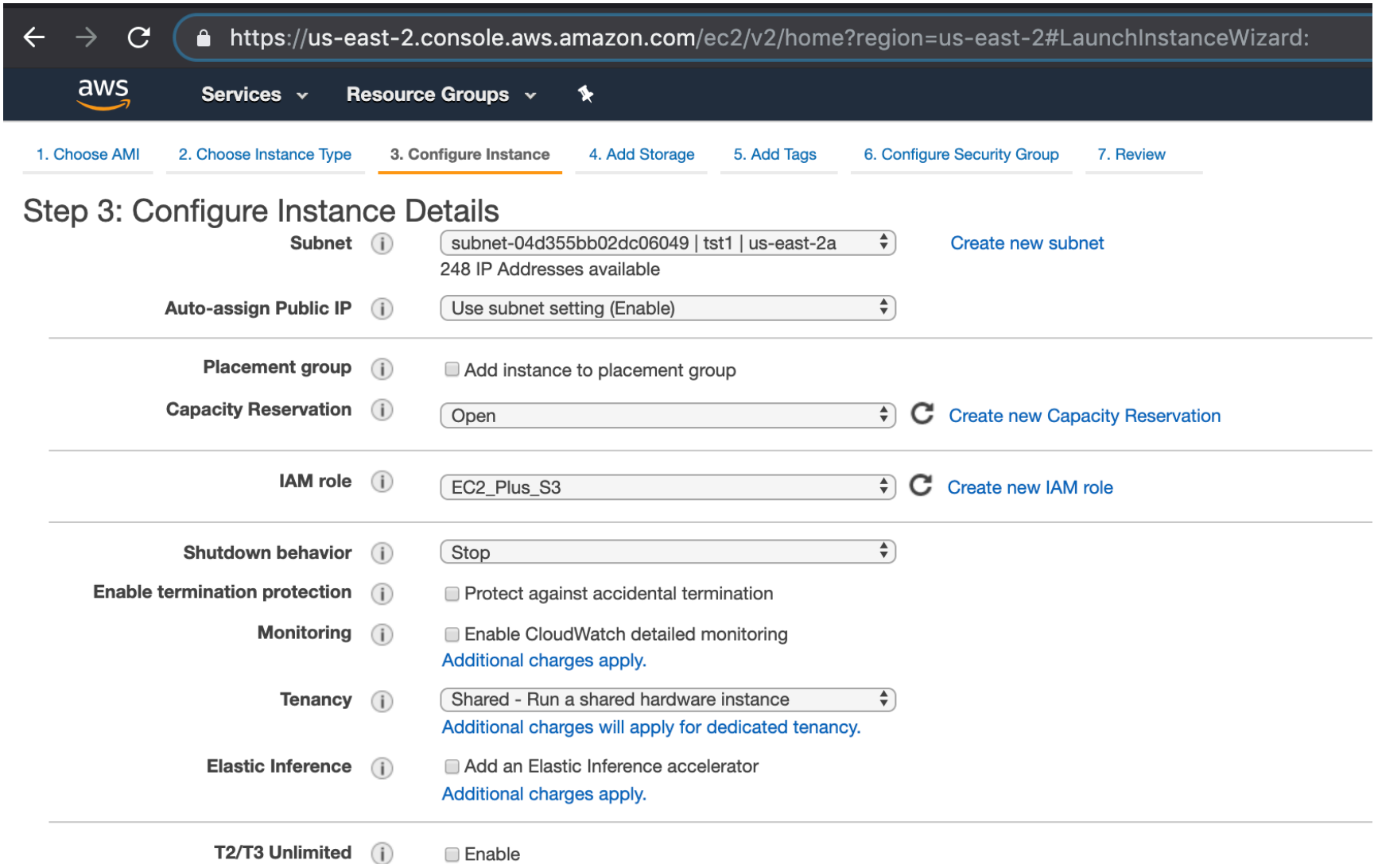
Currently selected: t2.large (Variable ECUs, 2 vCPUs, 2.3 GHz, Intel Broadwell E5-2686v4, 8 GiB memory, EBS only)

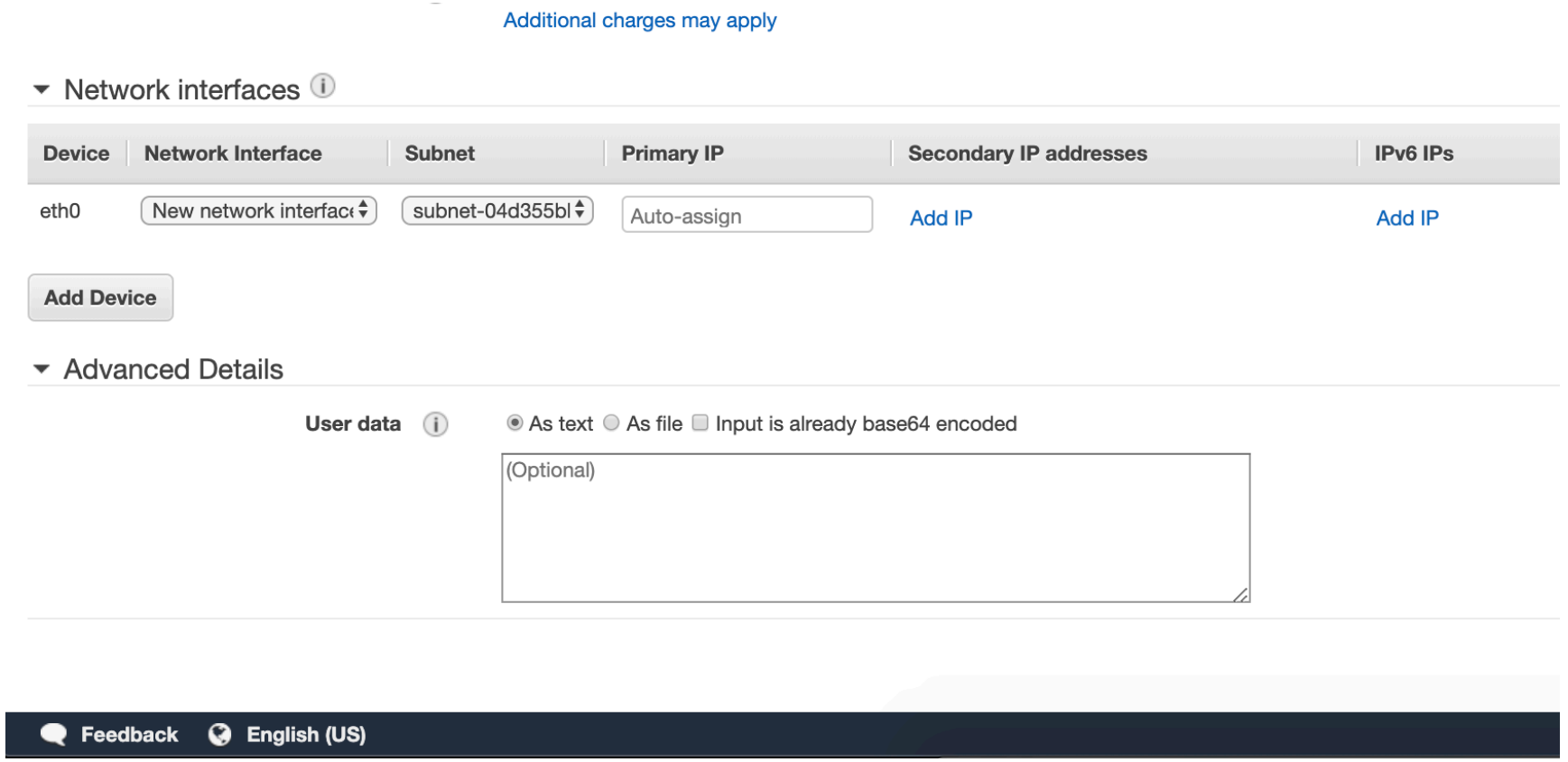
	Family	Type	vCPUs	Memory (GiB)
☐	General purpose	t2.nano	1	0.5
☐	General purpose	t2.micro Free tier eligible	1	1
☐	General purpose	t2.small	1	2
☐	General purpose	t2.medium	2	4
☑	General purpose	t2.large	2	8
☐	General purpose	t2.xlarge	4	16
☐	General purpose	t2.2xlarge	8	32

- In Step 3, do the following
  - Number of Instance: 2
  - Network: Select your Student VPC
  - Subnet: Select either subnet
  - Auto-Assign public IP: Select Enable (or leave the default if it is Enable)
  - **Under Advanced Detail**, copy and paste the contents of the `addCodeDeploytoEC2Image.sh` file in the optional text box. Make sure the As Text radio button is selected. The purpose of this script is to add the

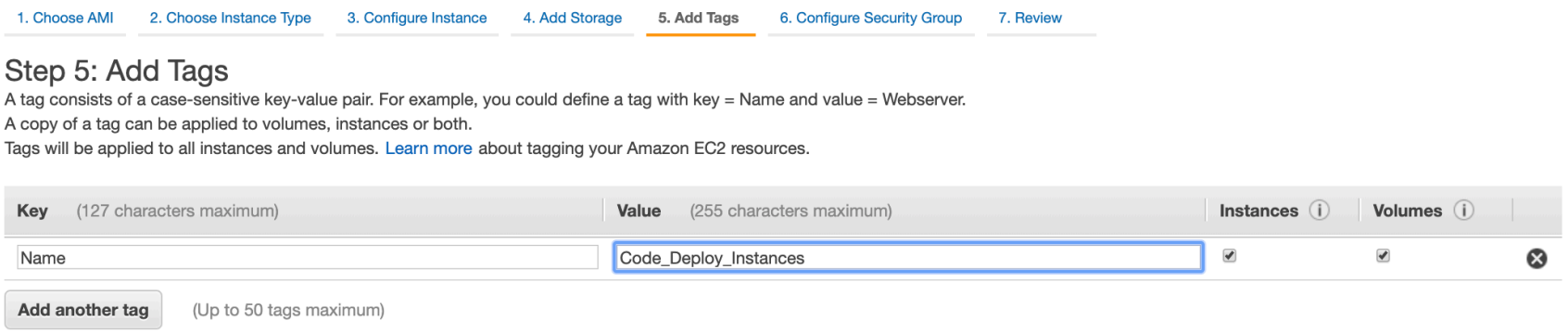
libraries and services necessary to the Ubuntu based VM to be able to deploy to the machine in an automated fashion. Note that the script uses the AWS EAST-2 region (Ohio), change this portion of the S3 url for the region you are using. Additionally, this script installs the services necessary to forward logs to CloudWatch to be used for monitoring and debugging deployments, and program runs. ref.

- Press Next

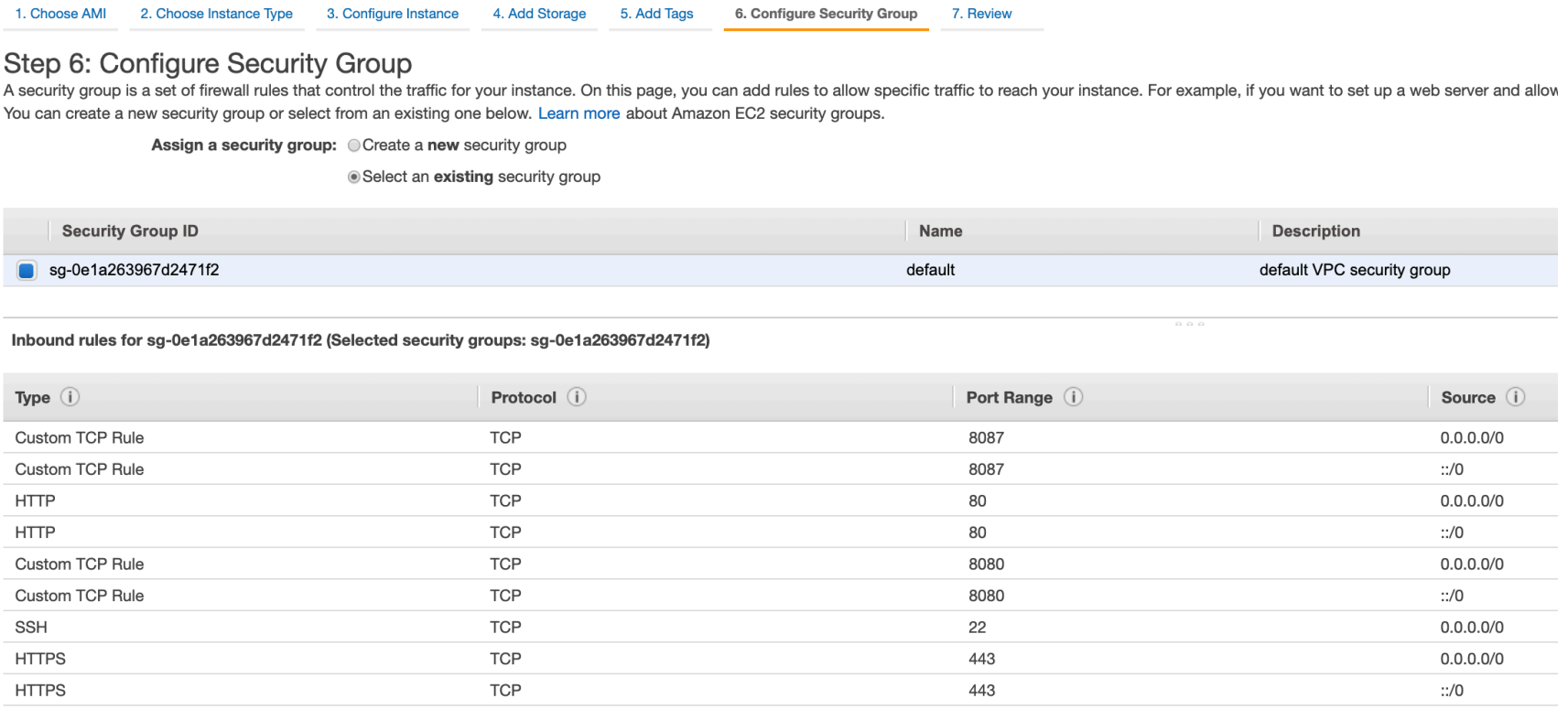




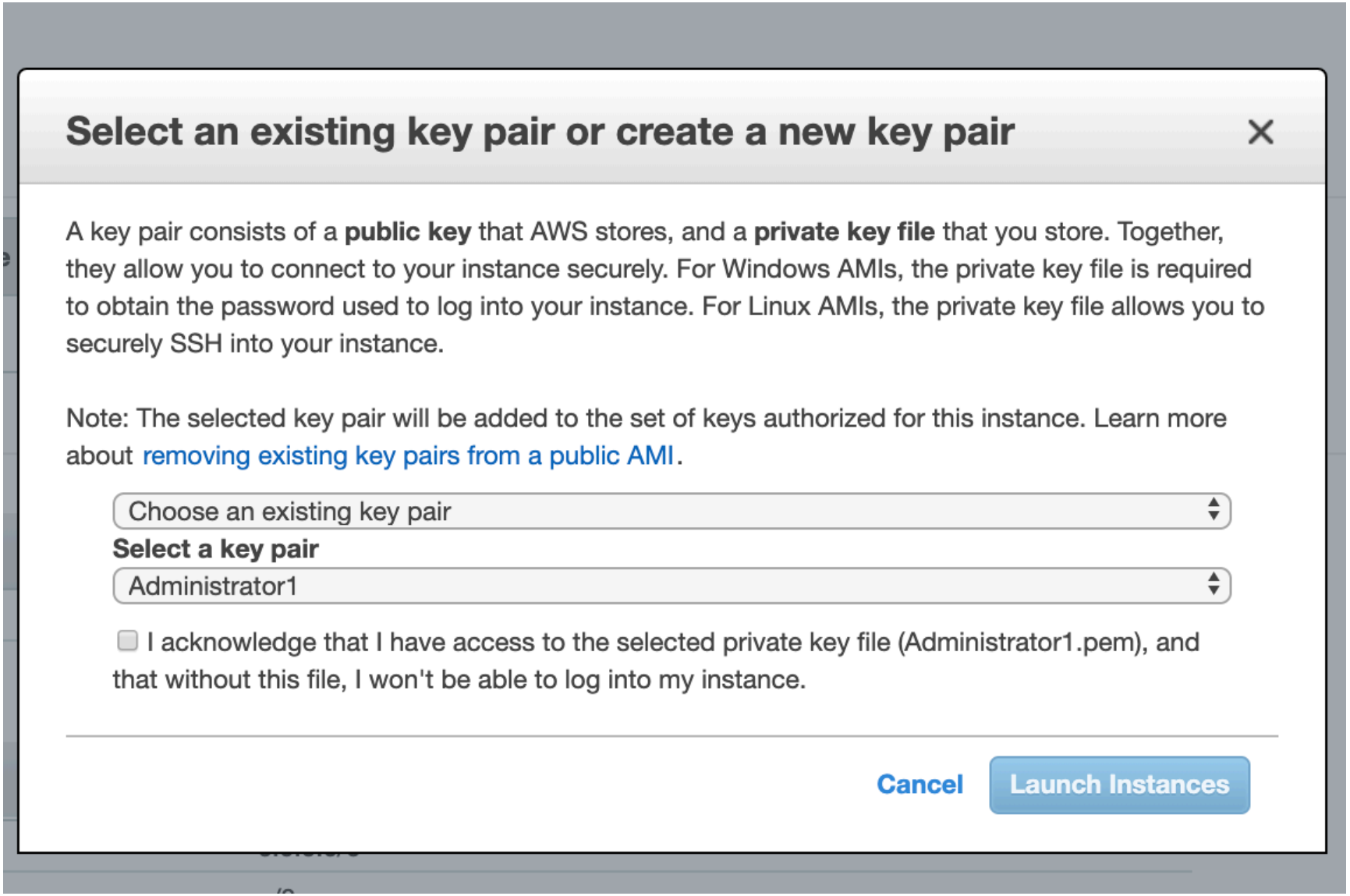
- Press Next on Step 4, using the default storage.
- In Step 5, add a tag with the Key = Name, and Value = Code_Deploy_Instances



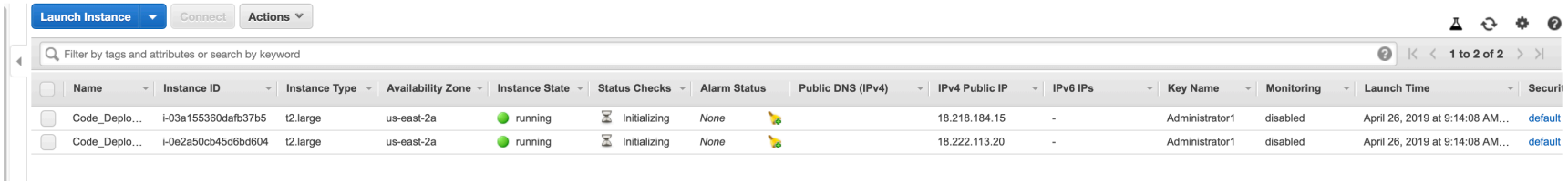
- Step 6, Select an Existing Security Group, then select your student Security group.
- Select Review and Launch.



- 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

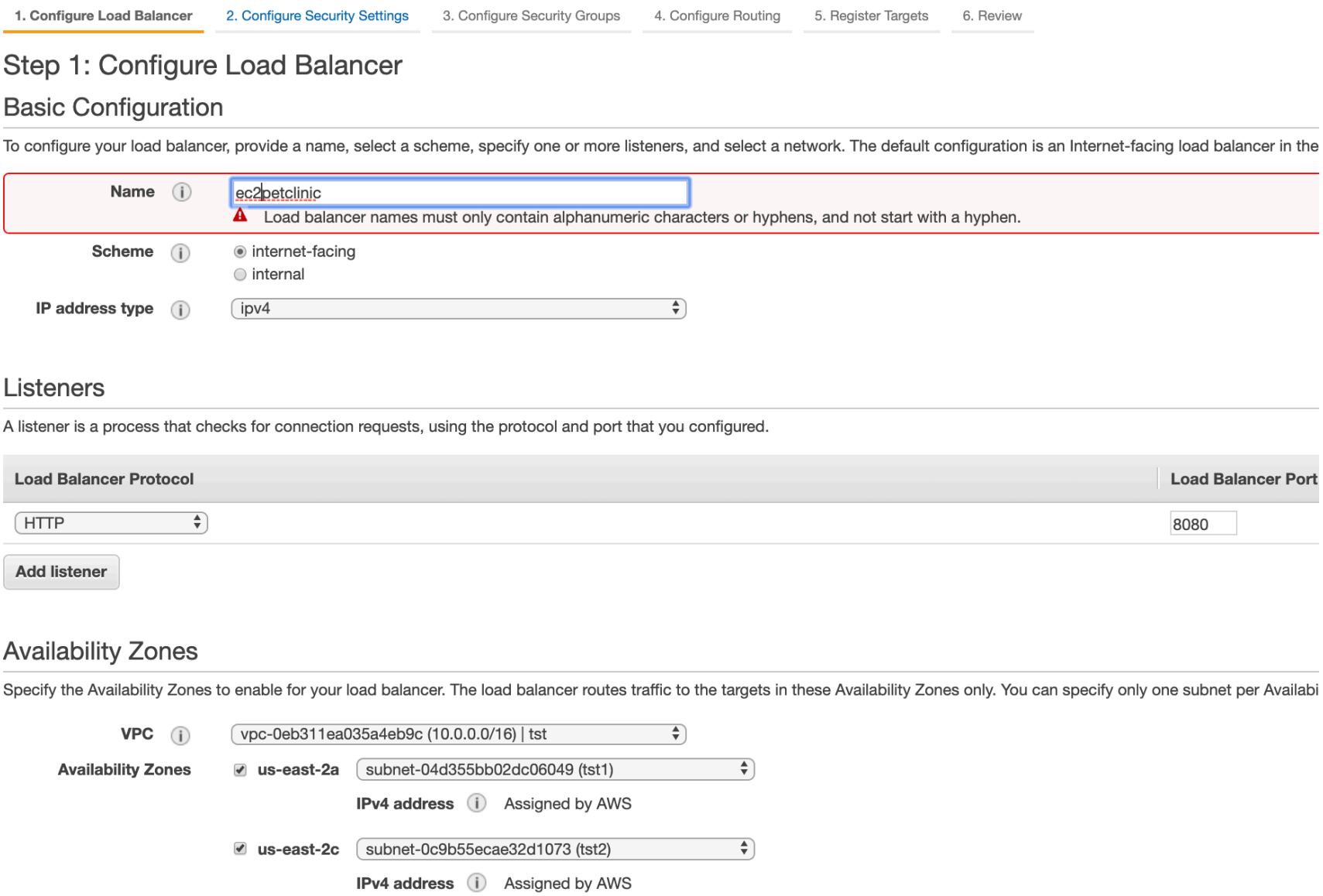


- Review the intances created and make note of the IP addresses

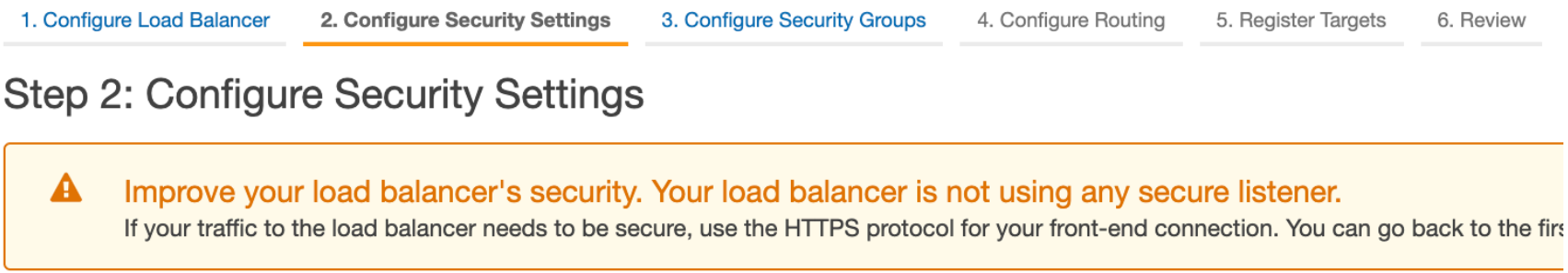


Load Balancer

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

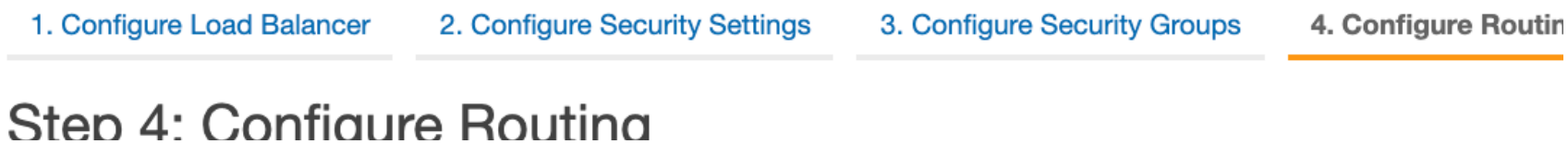

1. Configure Load Balancer
2. Configure Security Settings
3. Configure Security Groups
4. Configure Routing
5. Register Targets
6. Review
Step 1: Configure Load Balancer
Basic Configuration
To configure your load balancer, provide a name, select a scheme, specify one or more listeners, and select a network. The default configuration is an Internet-facing load balancer in the
Name
ec2petclinic
Load balancer names must only contain alphanumeric characters or hyphens, and not start with a hyphen.
Scheme
internet-facing
internal
IP address type
ipv4
Listeners
A listener is a process that checks for connection requests, using the protocol and port that you configured.
Load Balancer Protocol
Load Balancer Port
HTTP
8080
Add listener
Availability Zones
Specify the Availability Zones to enable for your load balancer. The load balancer routes traffic to the targets in these Availability Zones only. You can specify only one subnet per Availabi
VPC
vpc-0eb311ea035a4eb9c (10.0.0.0/16) | tst
Availability Zones
us-east-2a
subnet-04d355bb02dc06049 (tst1)
IPv4 address
Assigned by AWS
us-east-2c
subnet-0c9b55ecae32d1073 (tst2)
IPv4 address
Assigned by AWS
Tags

- Press Next for step 2



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

- Step 4, Create a new Target Group
  - Name: create a memorable name (e.g. StudentCodeDeployTG8080)
  - Port: 80
  - Path: /index.html
- Press Next



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

Target group

Target group (i) New target group

Name (i) TGcodedeployec2petclinic

⚠ TargetGroup name cannot contain characters that are not letters, c

Target type

- (•) Instance
- ( ) IP
- ( ) Lambda function

Protocol (i) HTTP

Port (i) 8080

Health checks

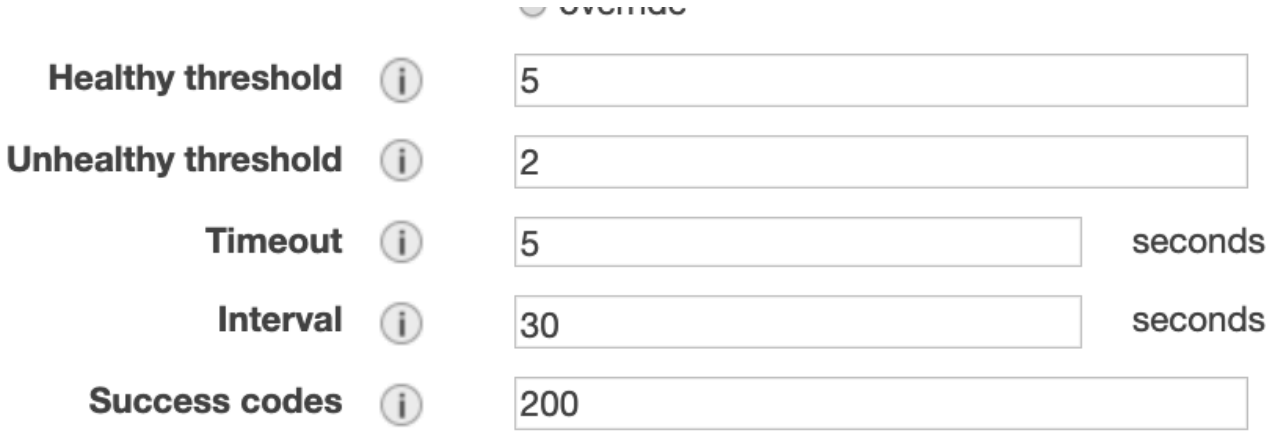
Protocol (i) HTTP

Path (i) /petclinic

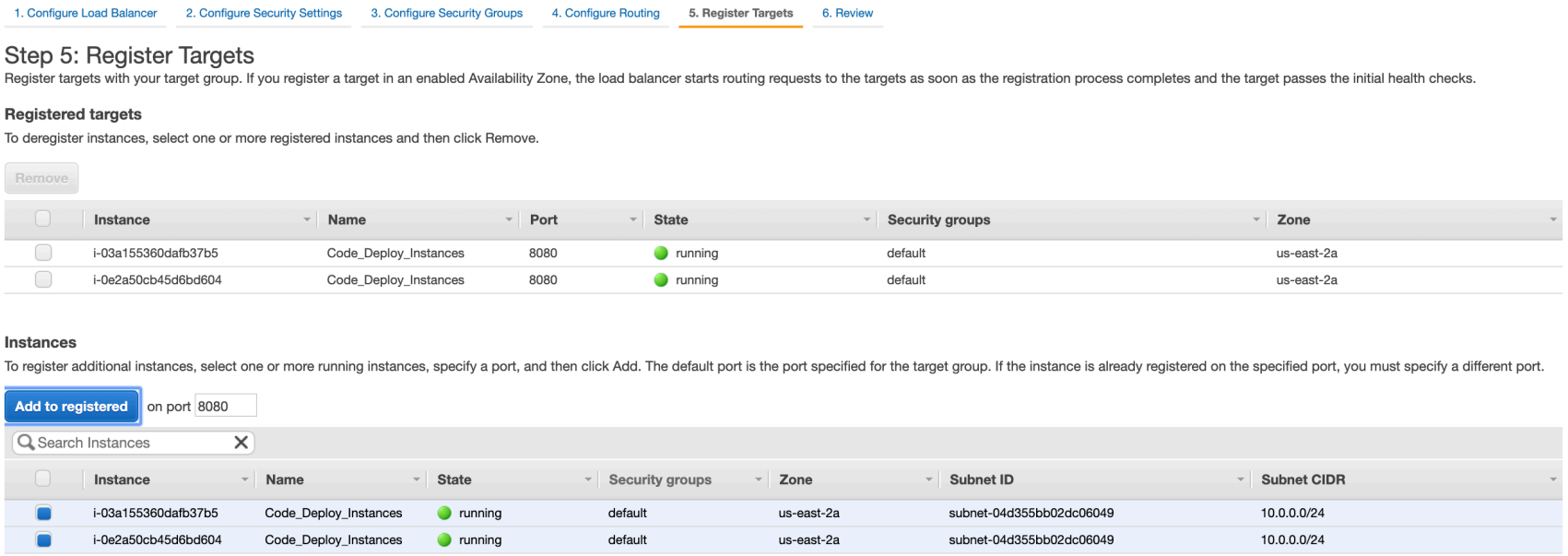
▼ Advanced health check settings

Port (i)

- (•) traffic port
- ( ) override



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



- Step 6, Review and Press Create

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

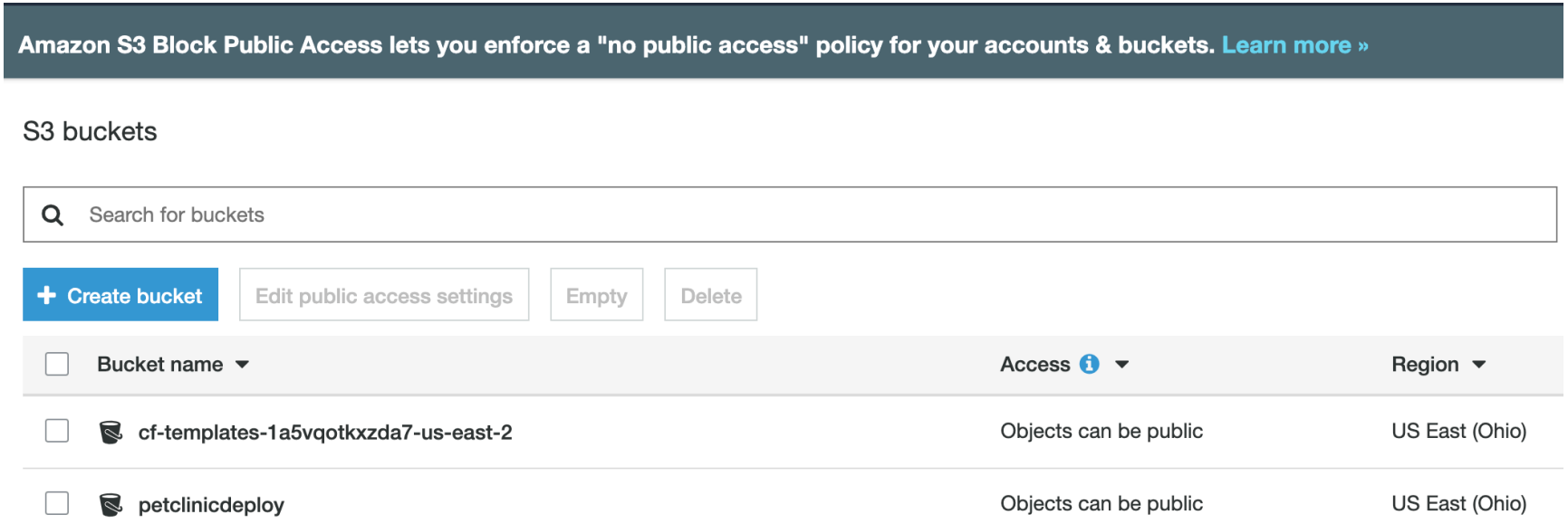
Target group	New target group
Target group name	TGcodedeployec2petclinic
Port	8080
Target type	instance
Protocol	HTTP
Health check protocol	HTTP
Path	/petclinic
Health check port	traffic port
Healthy threshold	5
Unhealthy threshold	2
Timeout	5
Interval	30
Success codes	200

Targets

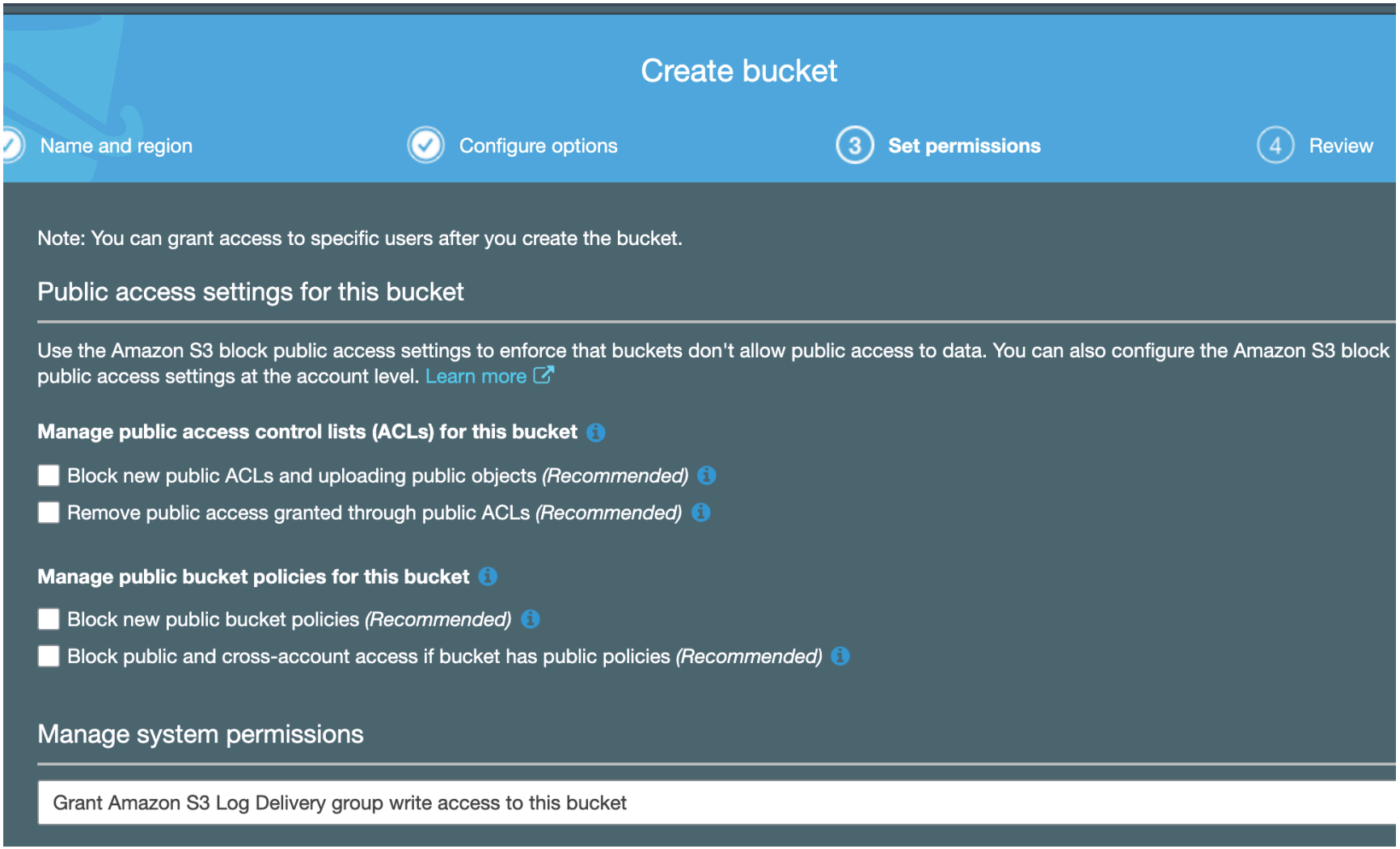
Instances	i-03a155360dafb37b5 (Code_Deploy_Instances):8080, i-0e2a50cb45d6bd604 (Code_Deploy_Instances):8080

Create S3 Bucket

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



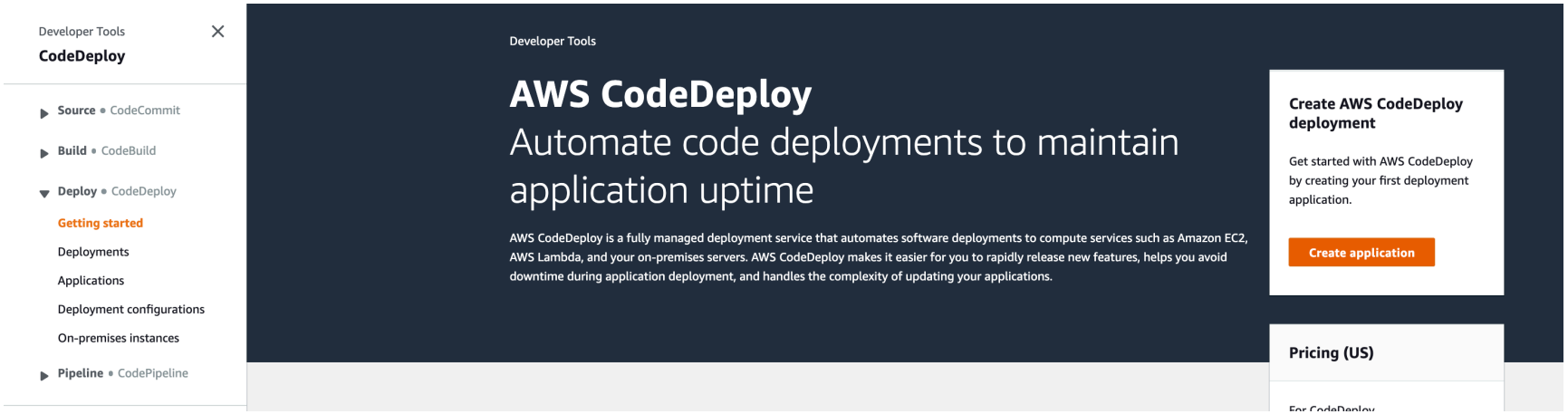
- Choose a name for your bucket and your region (we are using EAST-2 (OHIO)) and proceed through the wizard (Remember the bucket name as it will be used in the Jenkins step later)
- When presented with the "Public access settings for this bucket," de-select all check boxes and change the manage option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



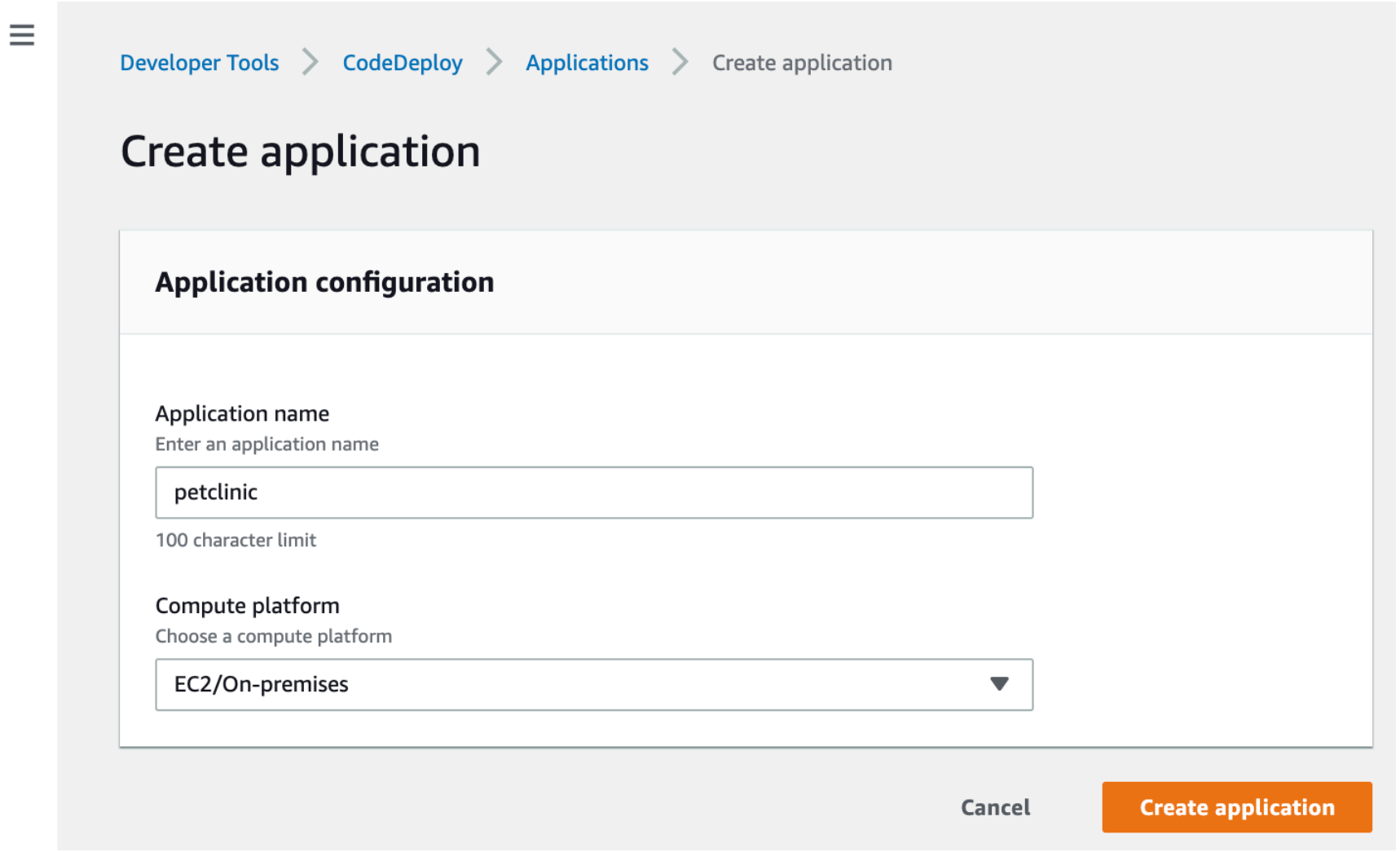
Create Codedeploy Application

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

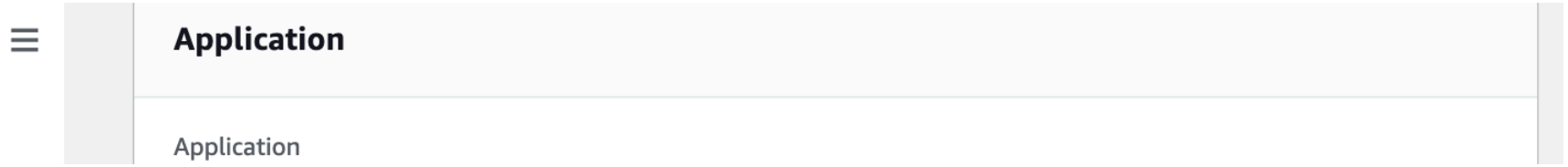
- From the services menu at the top of the page select CodeDeploy
- Press Create Application



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



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



petclinic

Compute type

EC2/On-premises

Deployment group name

Enter a deployment group name

petclinicDepGrp

100 character limit

Service role

Choose a service role

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

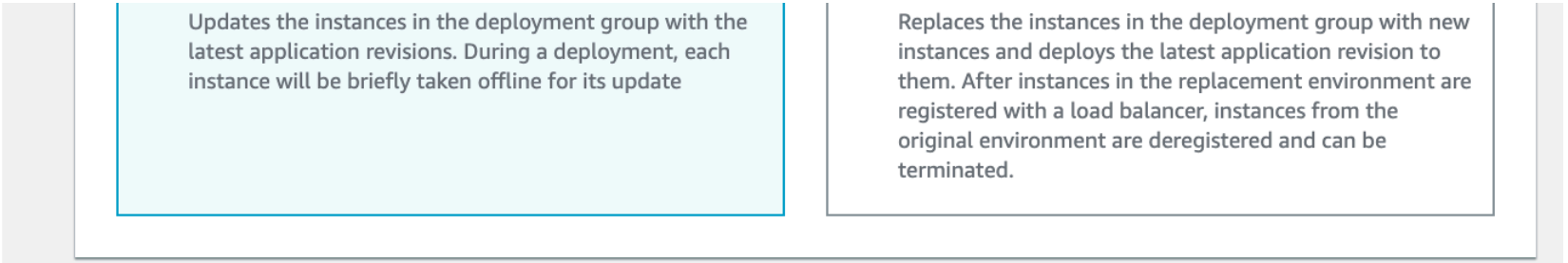
CodeDeployServiceRole

Deployment type

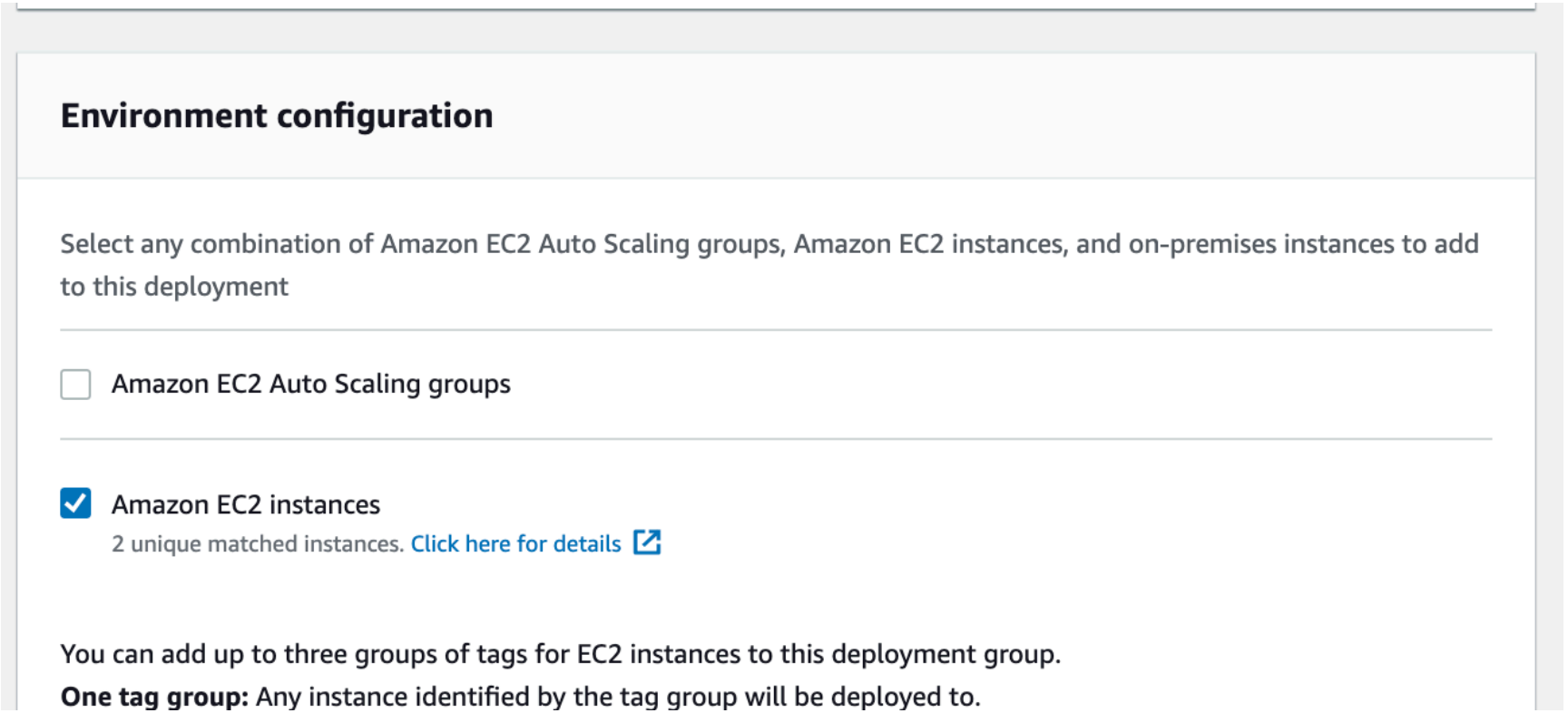
Choose how to deploy your application

In-place

Blue/green



- 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)



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

Tag group 1

Key

Name

Value - *optional*

Code_Deploy_Instances

Remove tag

Add tag

Add tag group

On-premises instances

Matching instances

2 unique matched instances. Click here for details

Deployment settings

Deployment configuration

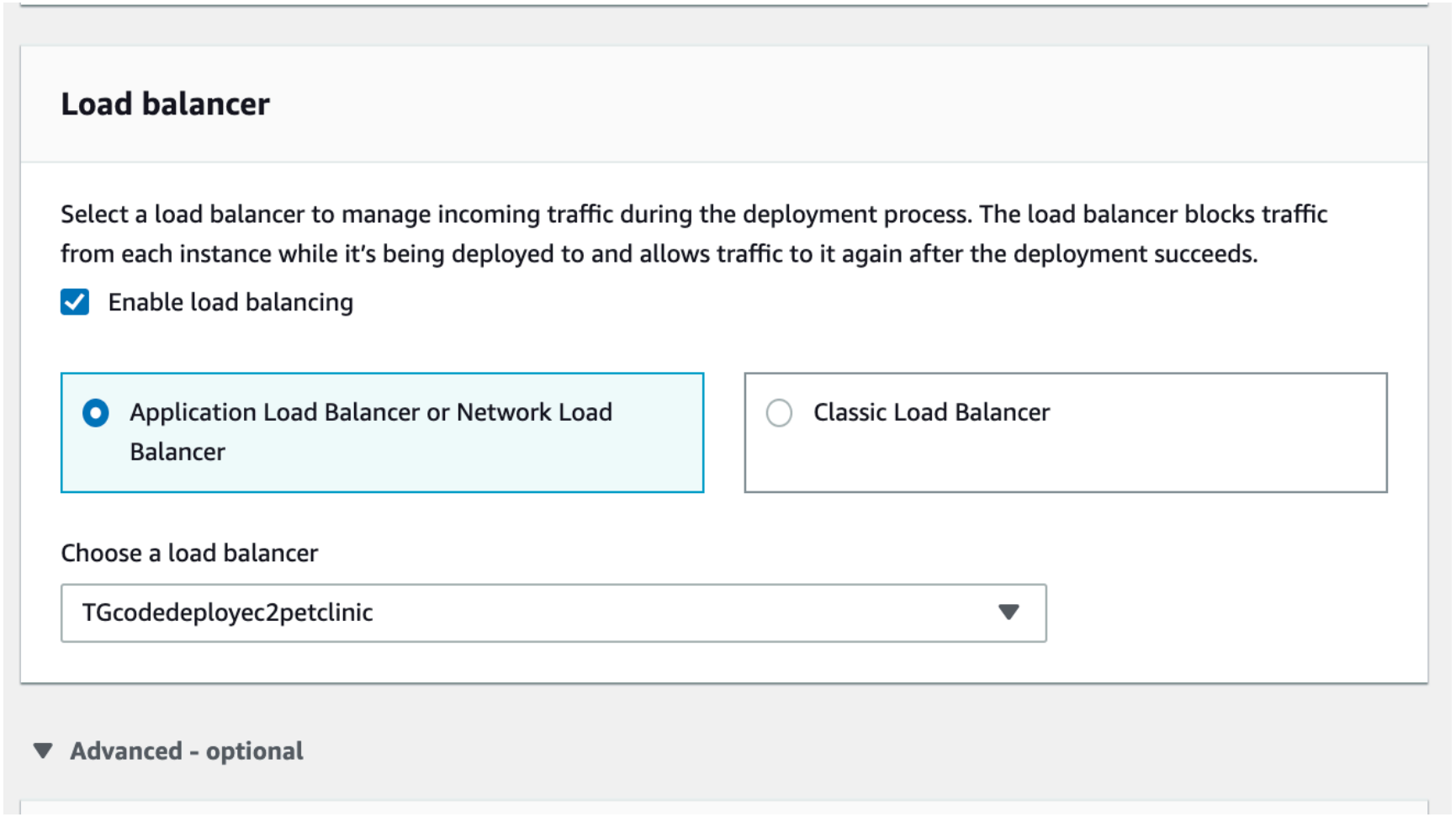
Choose from a list of default and custom deployment configurations. A deployment configuration is a set of rules that determines how fast an application will be deployed and the success or failure conditions for a deployment.

CodeDeployDefault.OneAtATime

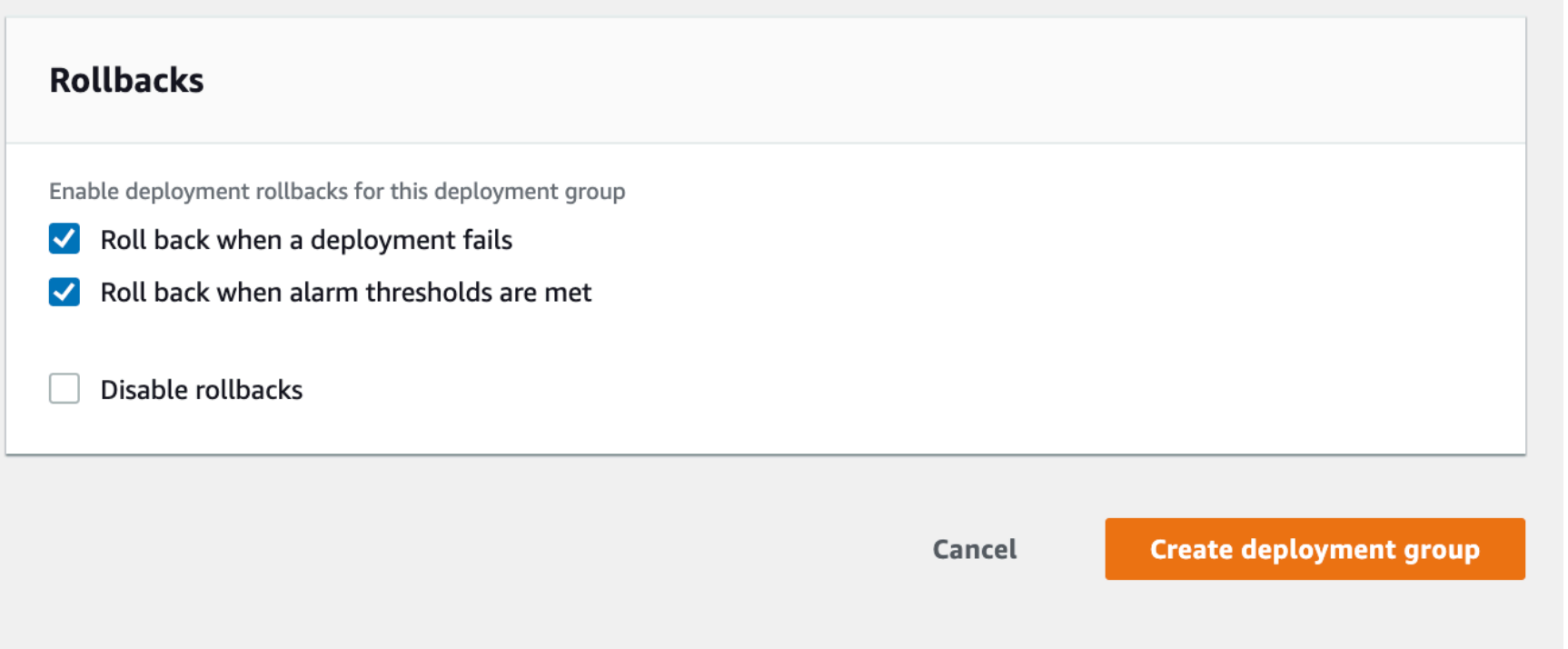
or

Create deployment configuration

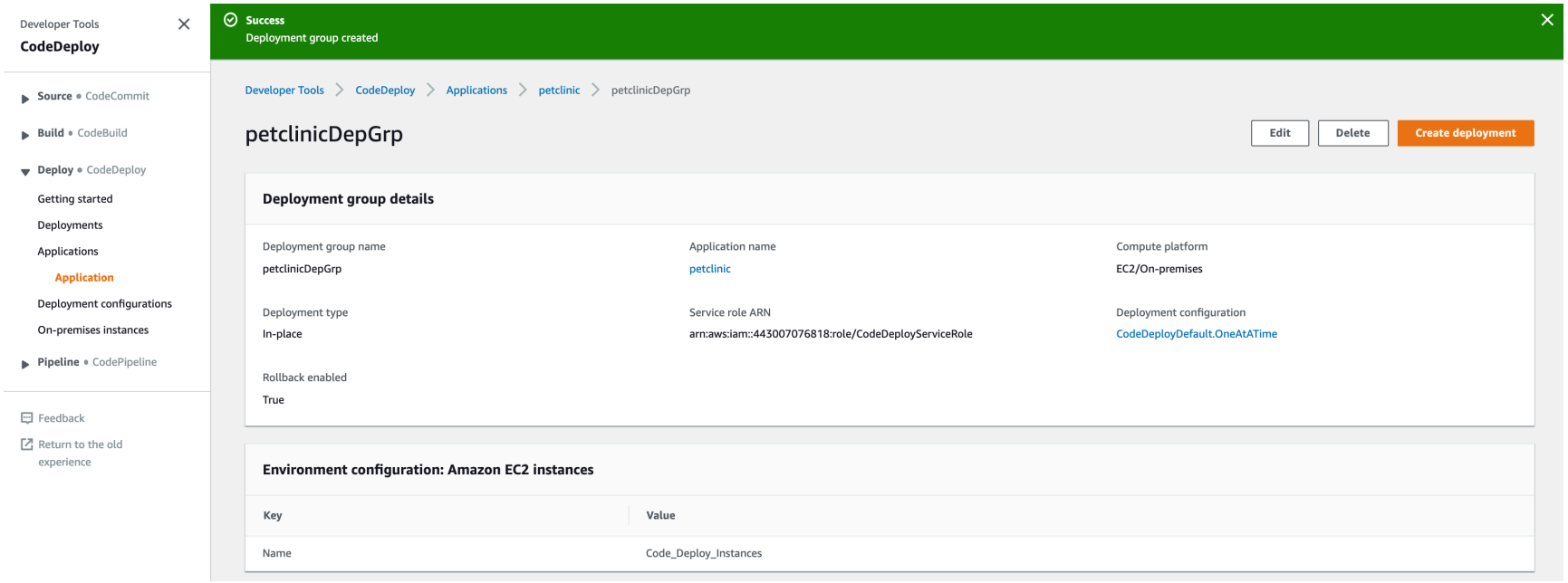
- Enable Load Balancing and select Application Load Balancer, then the Load Balancer created previously



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



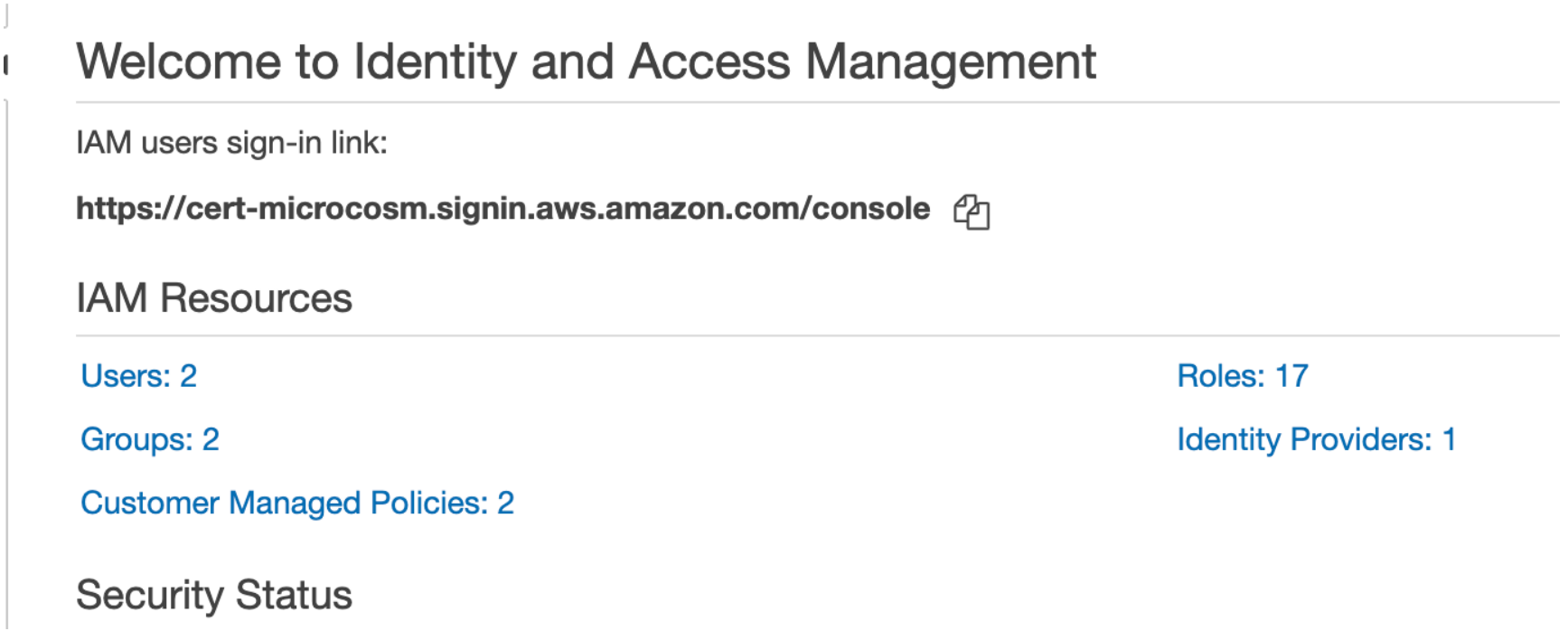
- Leave the Deployment Group Summary page open for reference in the Modify Jenkins Build Section



Create Secret Key

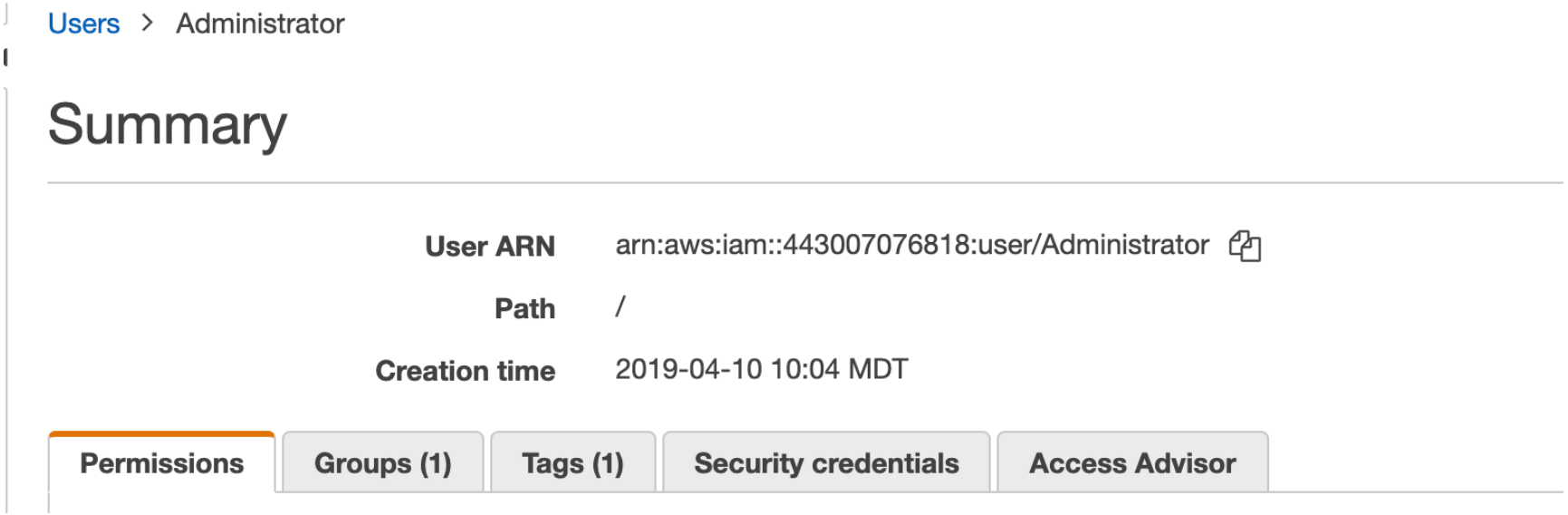
A Secret key is necessary for authenticating your Jenkins server with AWS. Where possible, the use of roles or PKI is preferred for authentication, however, in this instance the use of Secret keys is acceptable.

- Navigate to the IAM Concole

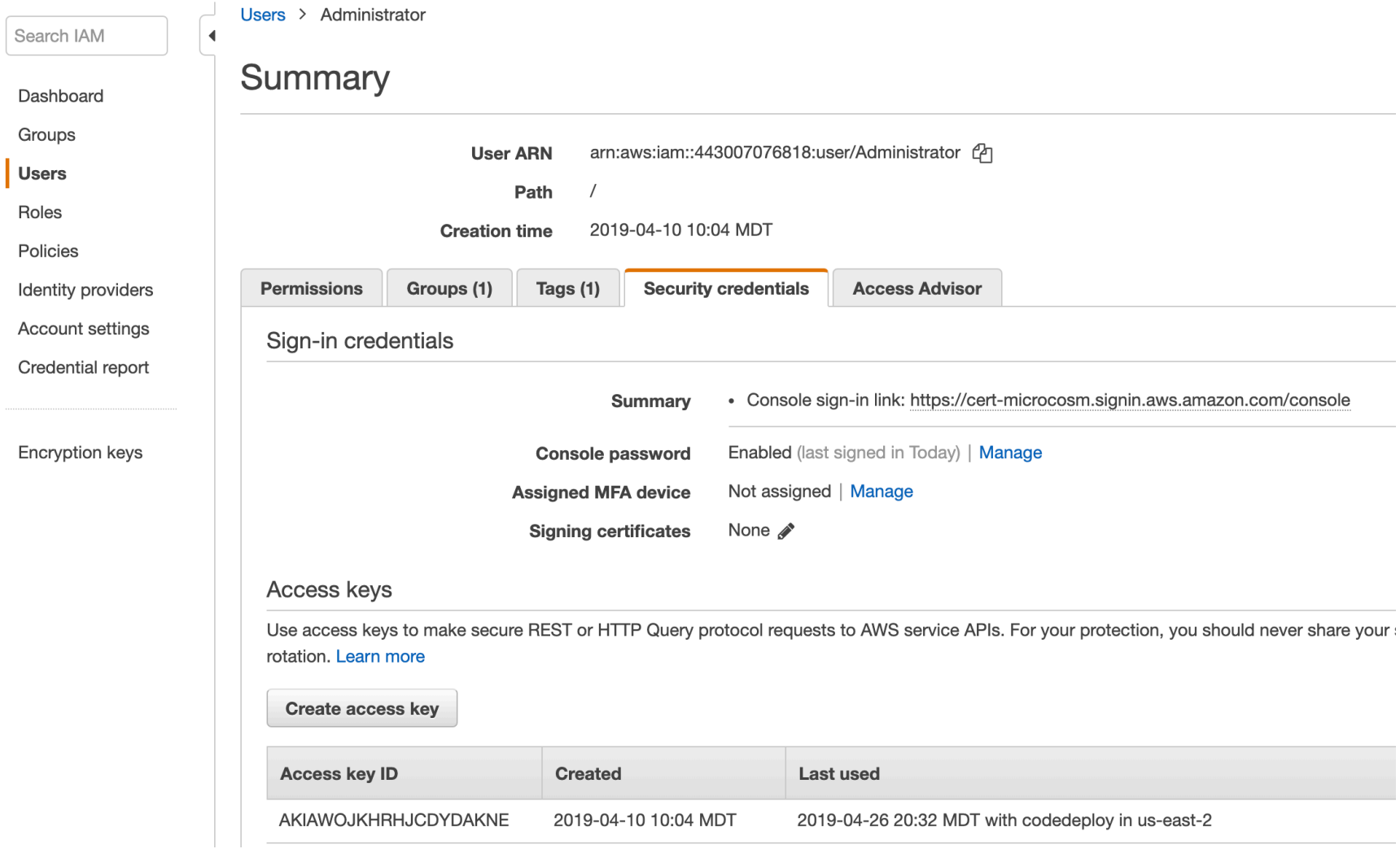


- Select your user

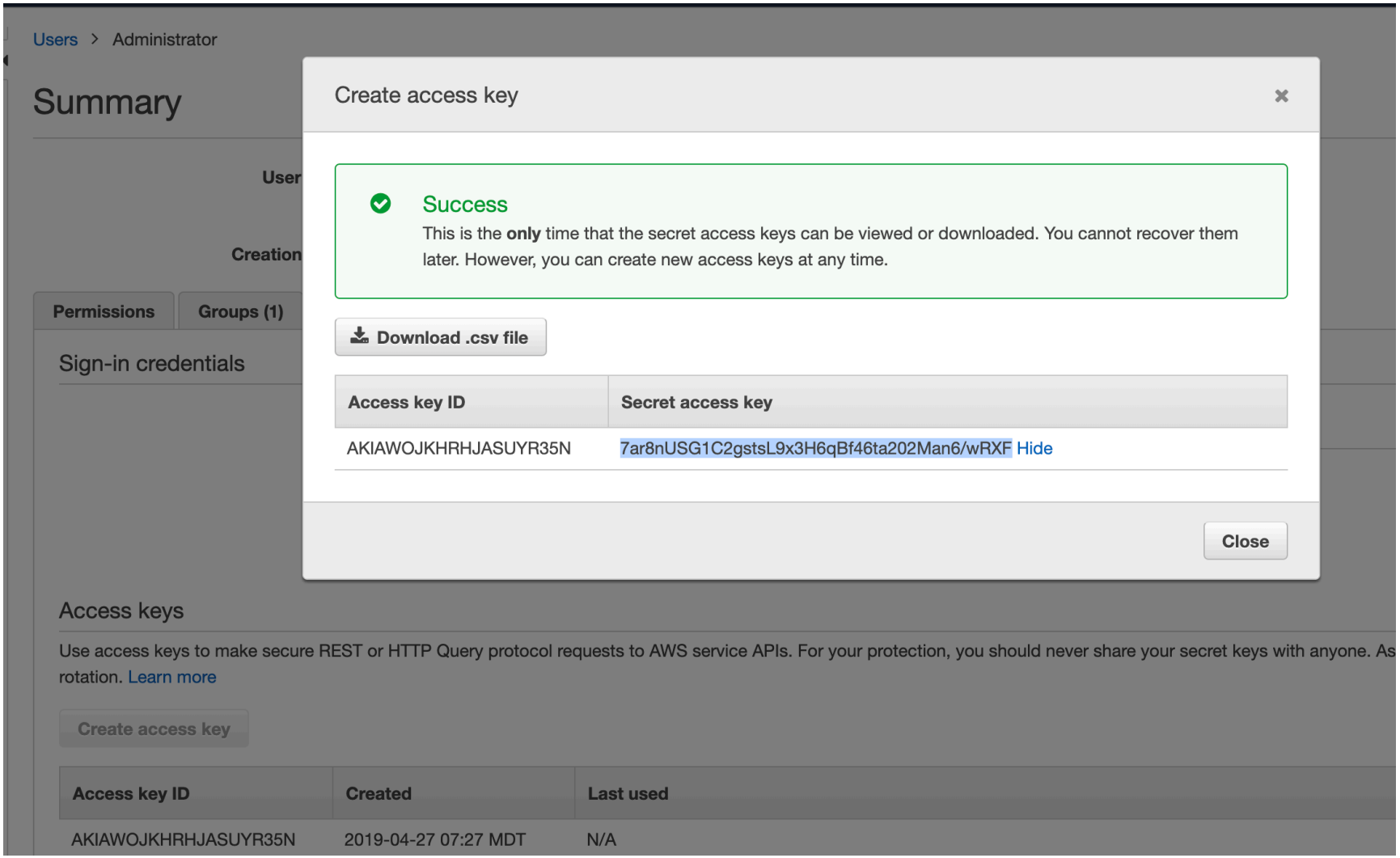
- Select the Security Credentials Tab



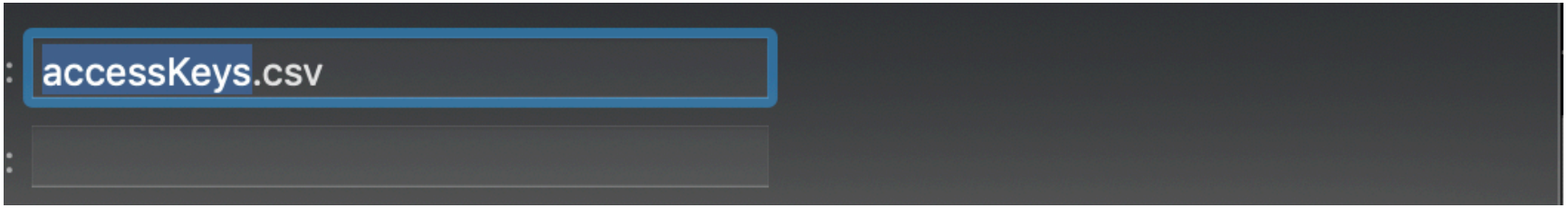
- Select Create Access Key

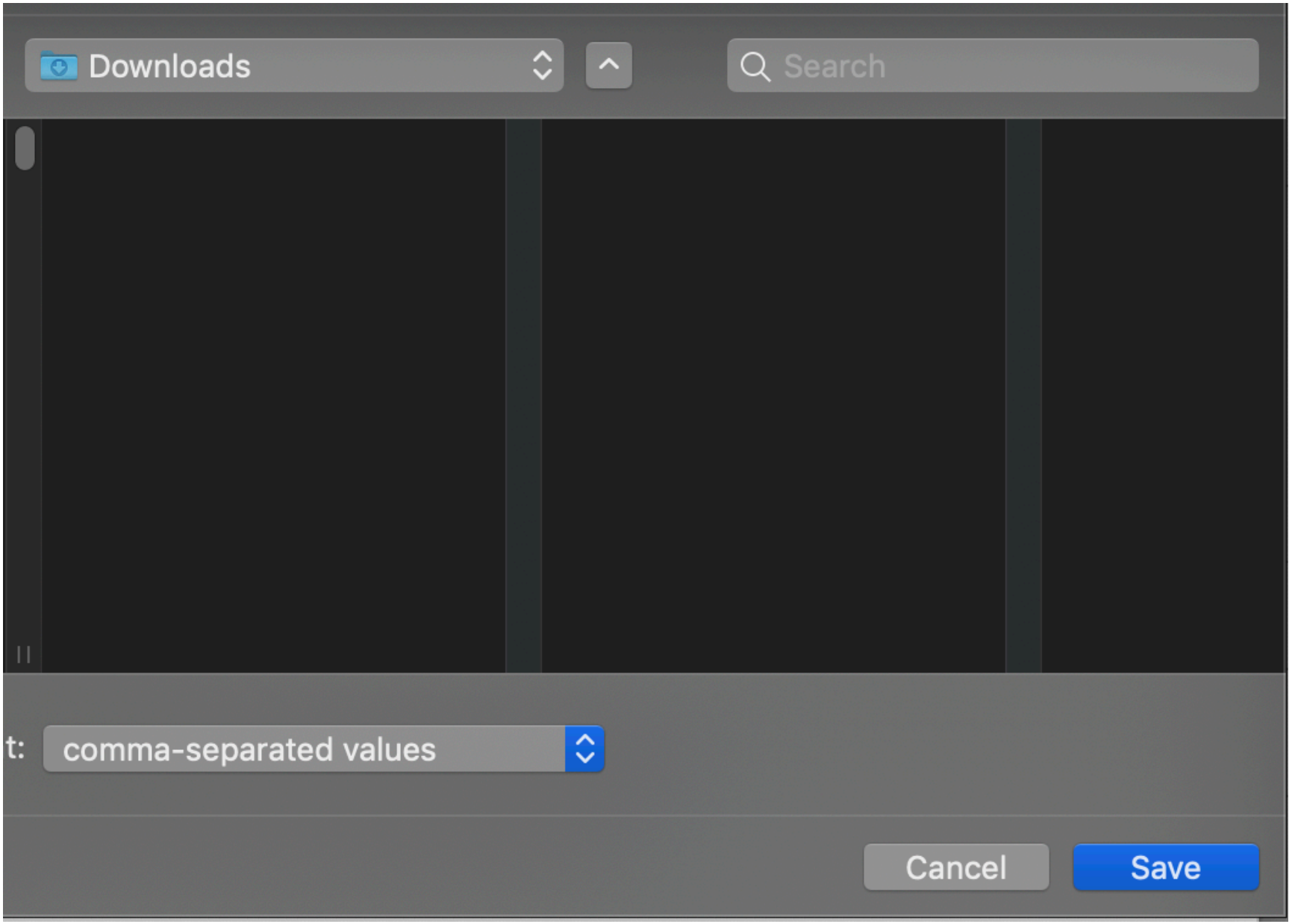


- You can unhide the access key to copy it and save it somewhere safe or press Download .csv file



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





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

Below.

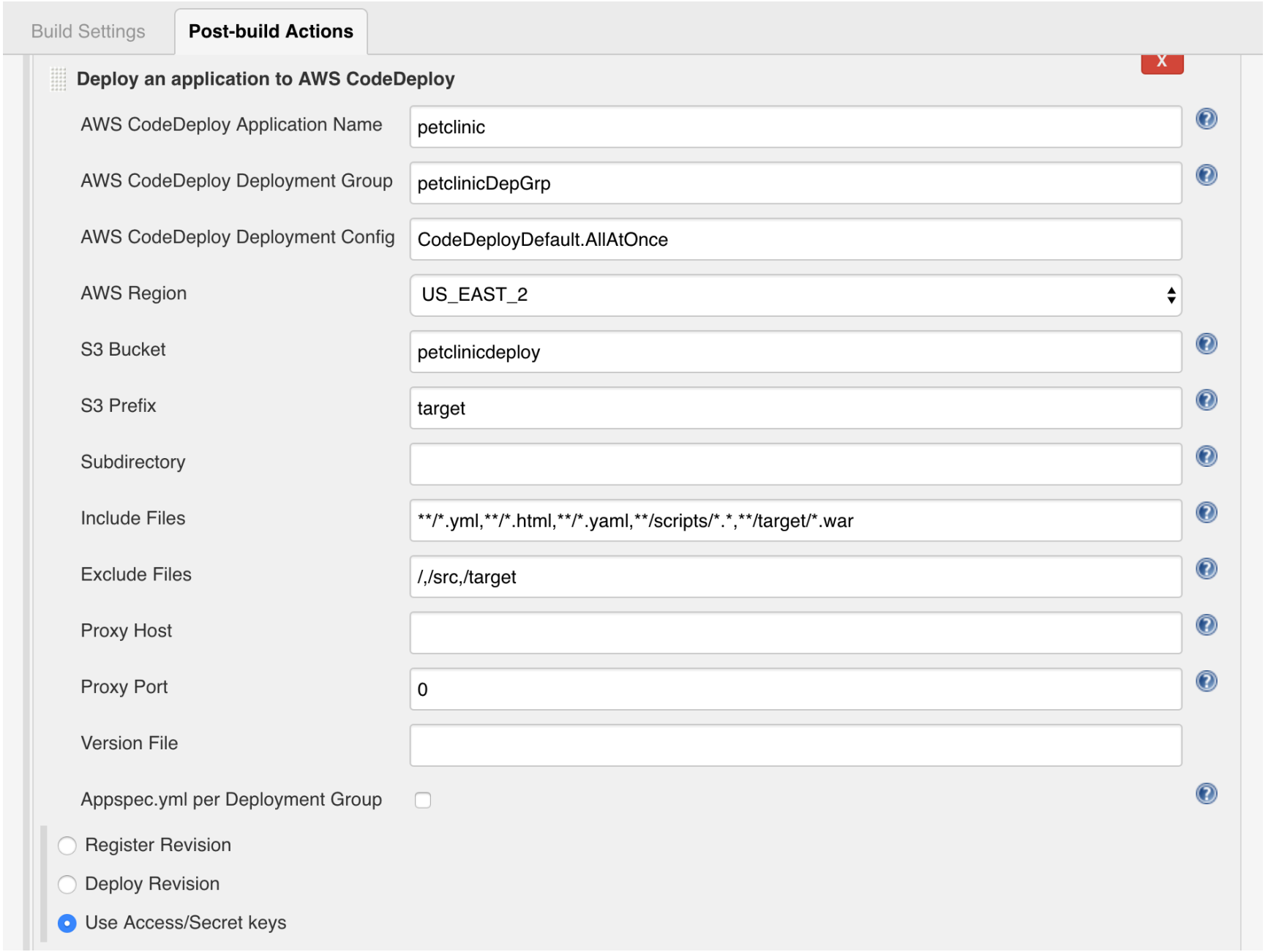
accessKeys

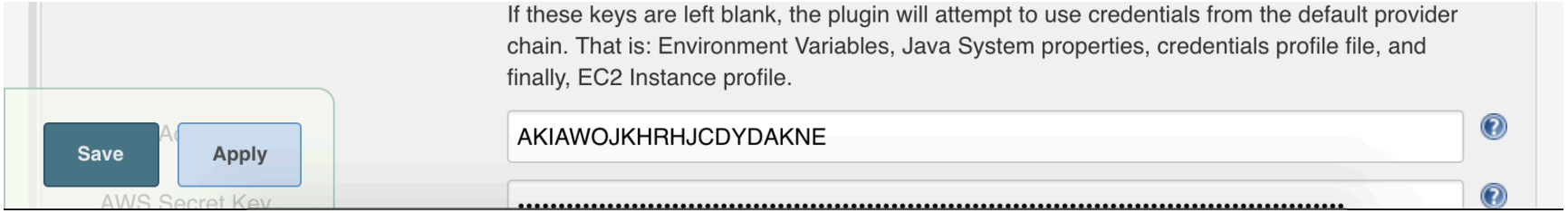
Access key ID	Secret access key
AKIAWOJKHRHJASUYR35N	7ar8nUSG1C2gstsL9x3H6qBf46ta202Man6/wRXF

Modify Jenkins Build

- Navigate to your product page, select configure, and scroll all the way down to Post-Build Actions
- From yout 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

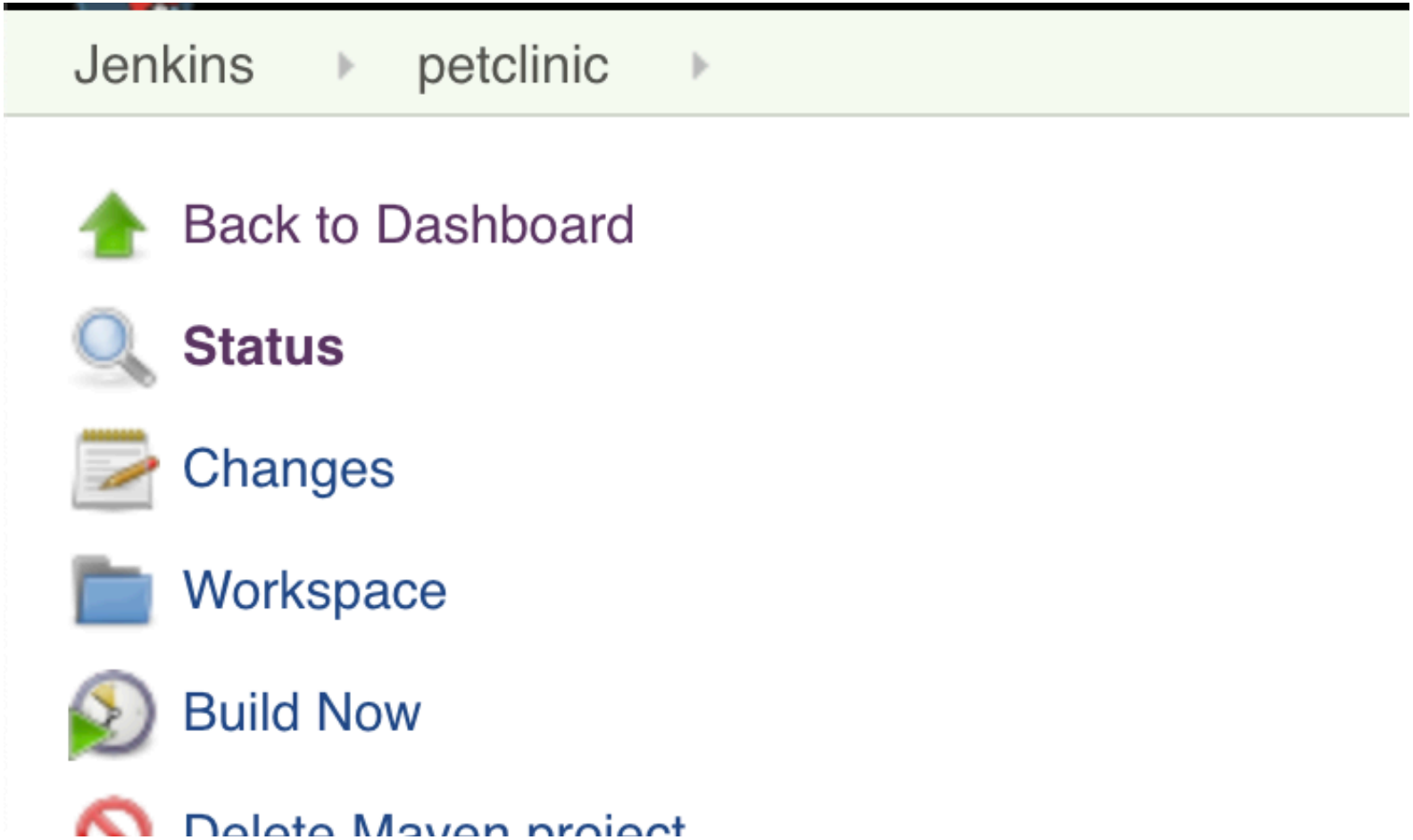
- Press Apply and Save

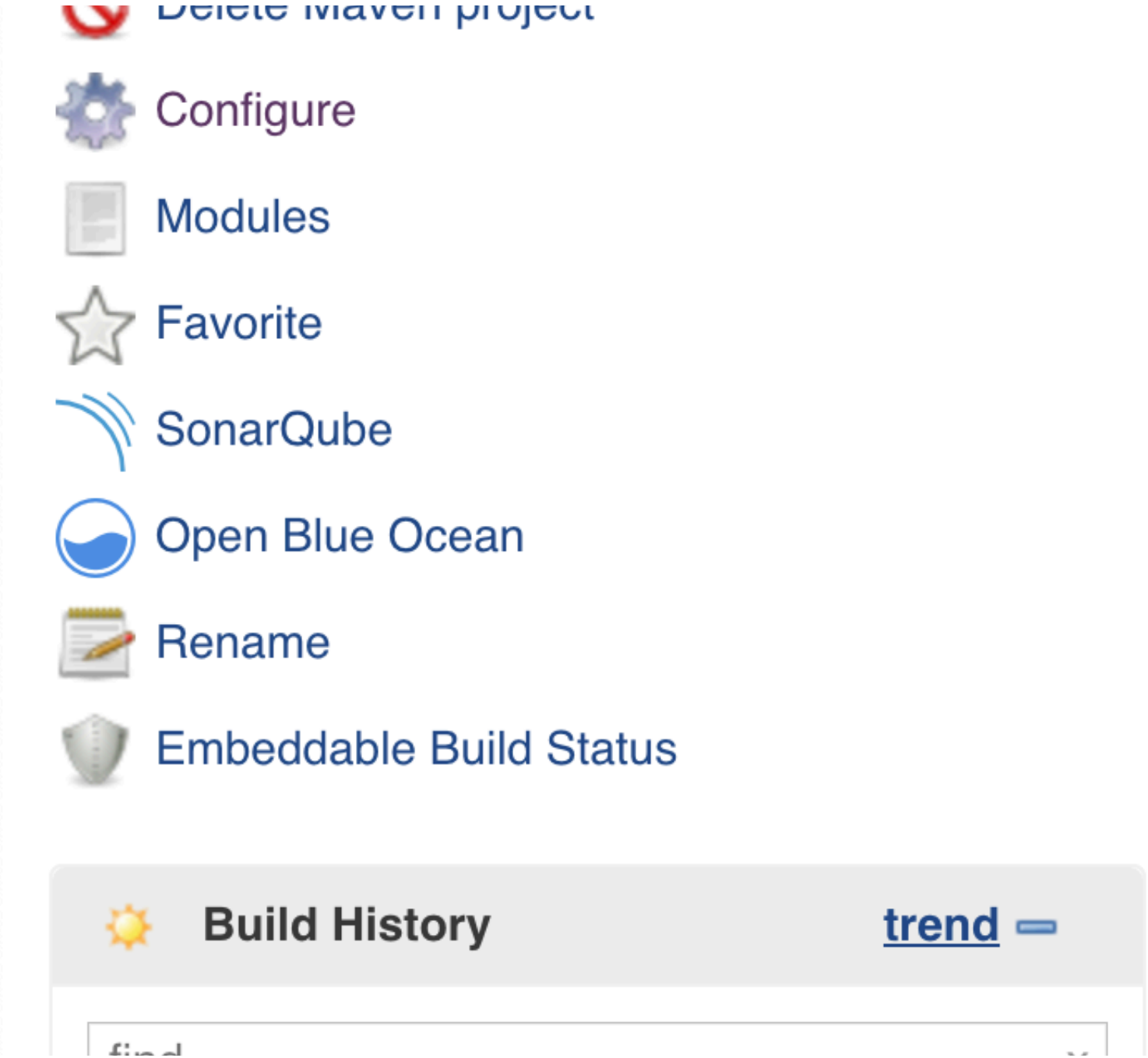


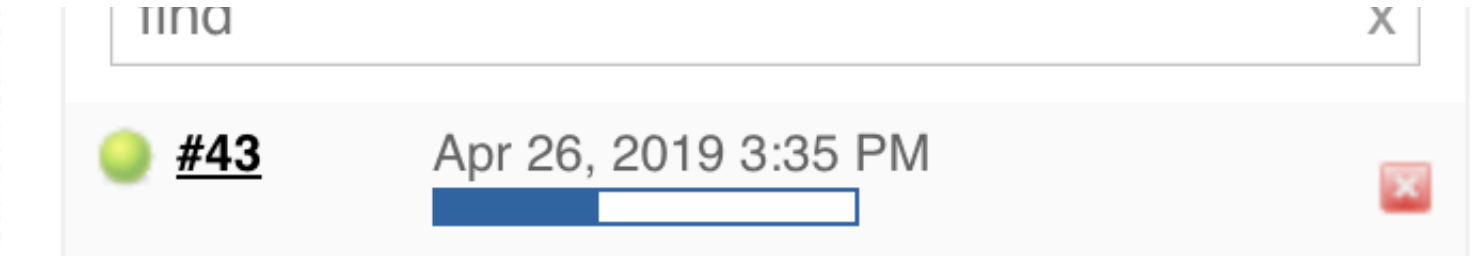


Build and Deploy

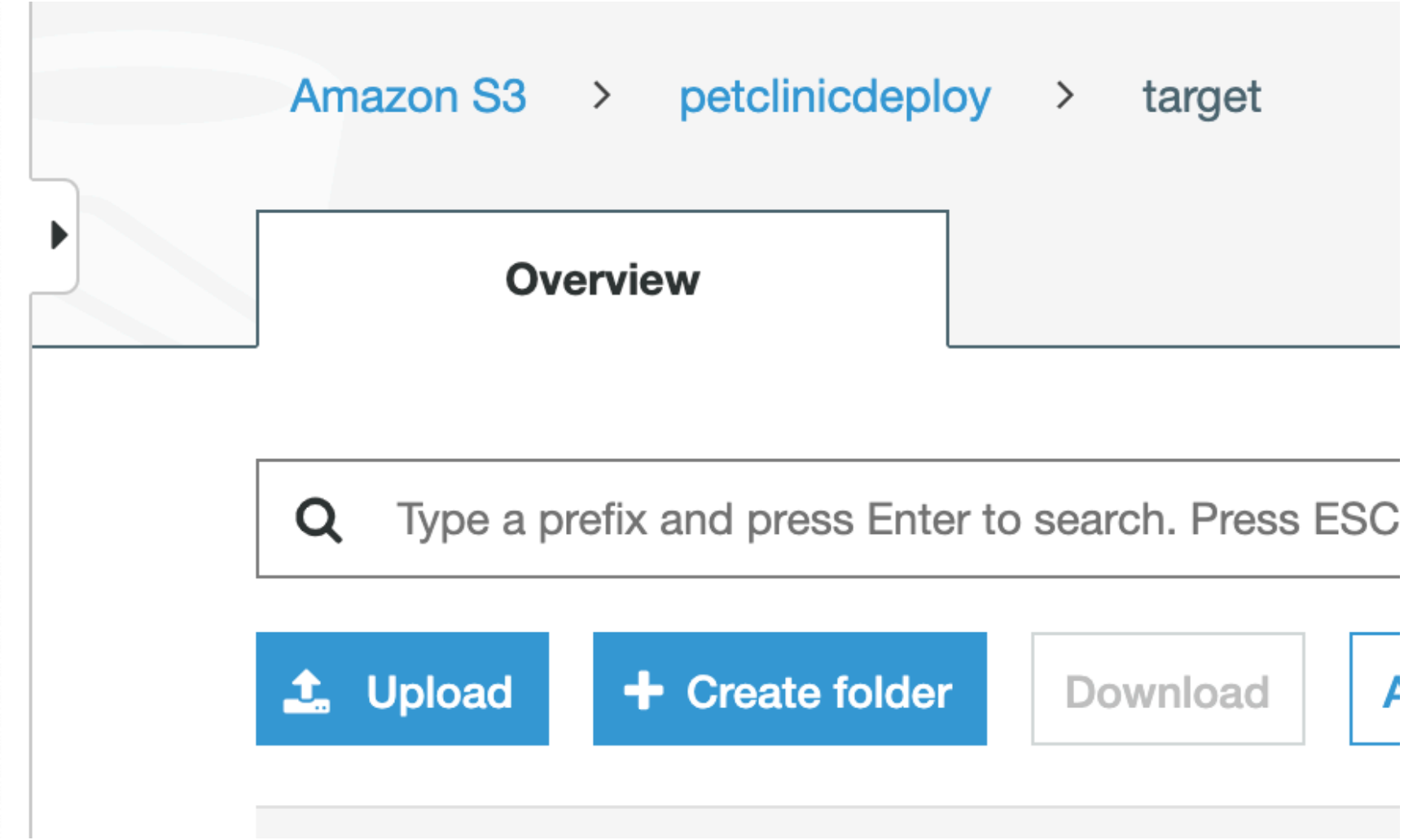
- From your Project Page in Jenkins select Build

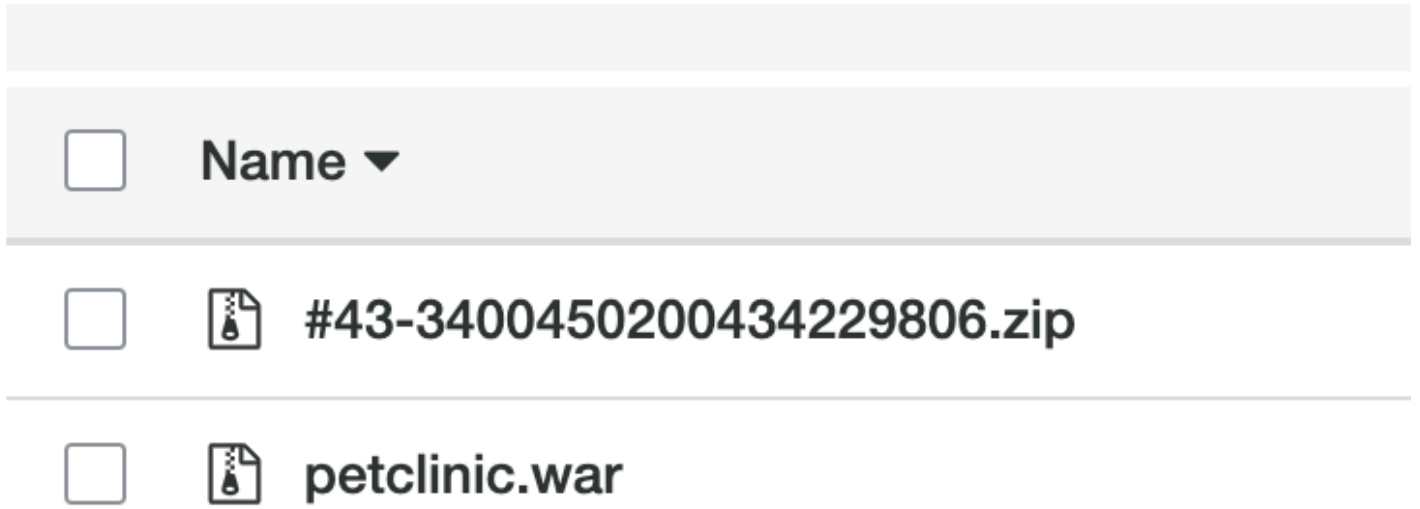



Delete Maven project
Configure
Modules
Favorite
SonarQube
Open Blue Ocean
Rename
Embeddable Build Status
Build History
trend

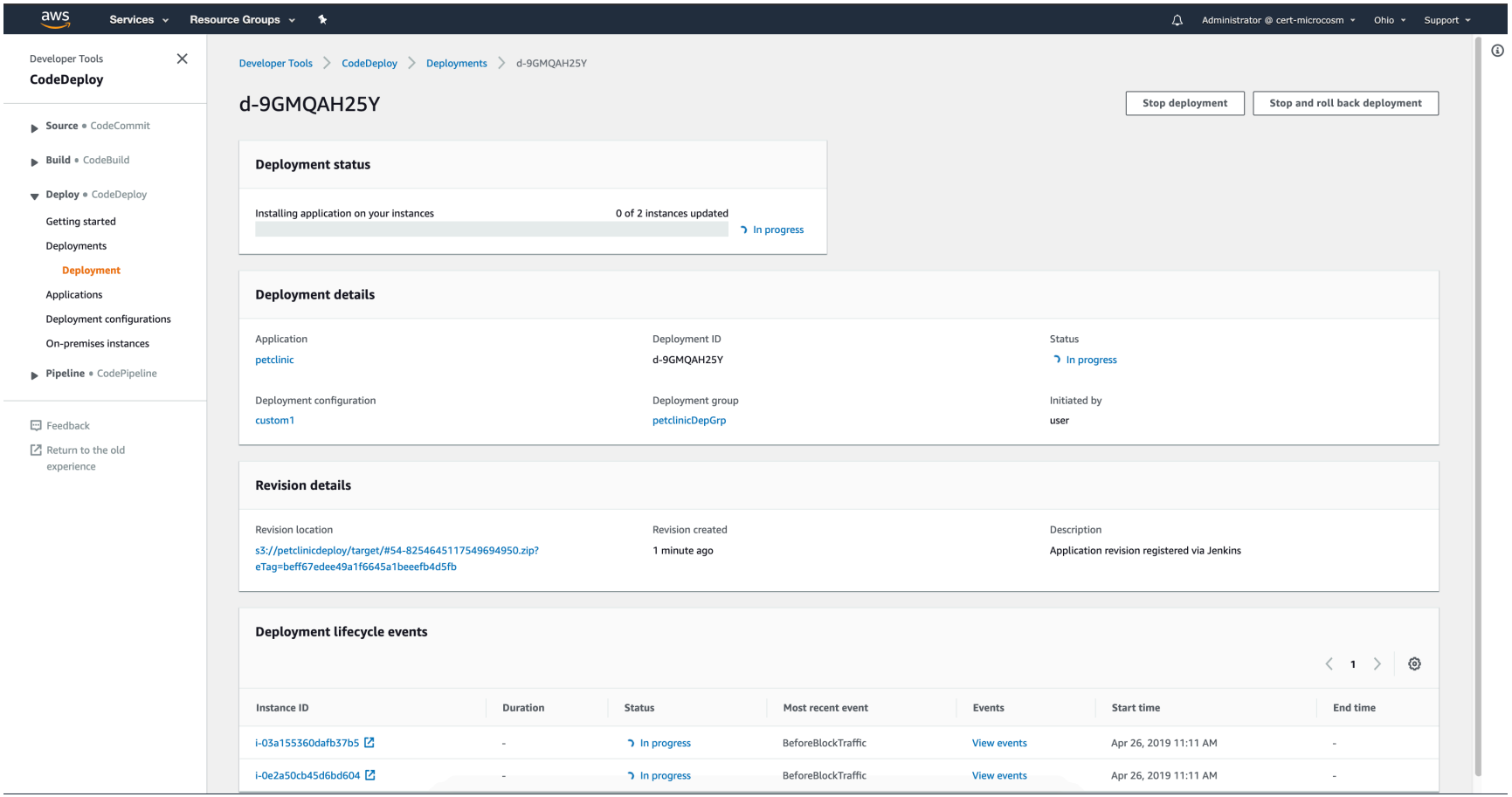


- Post build, you can navigate to the target folder in your S3 bucket to see the package uploaded

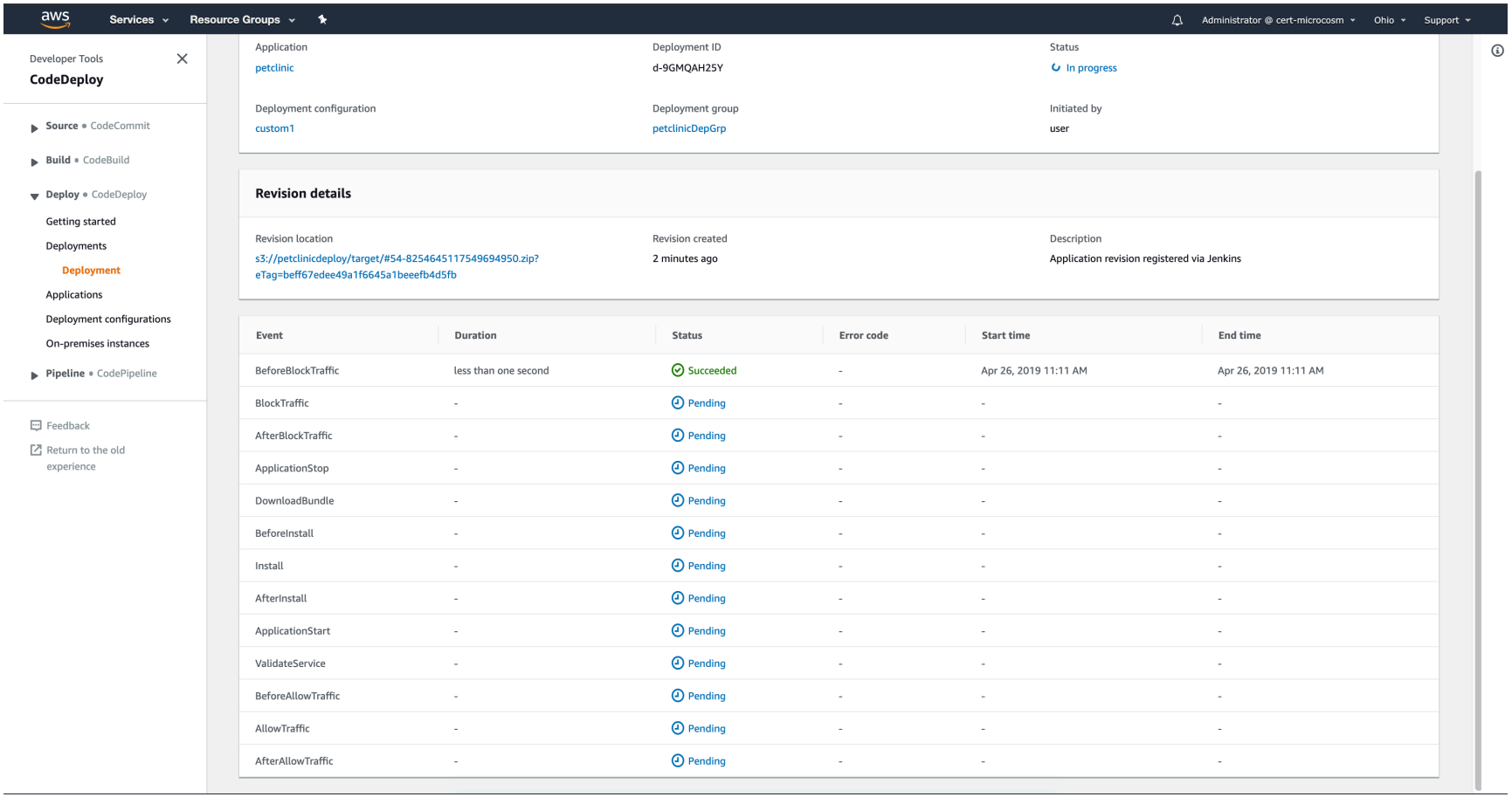


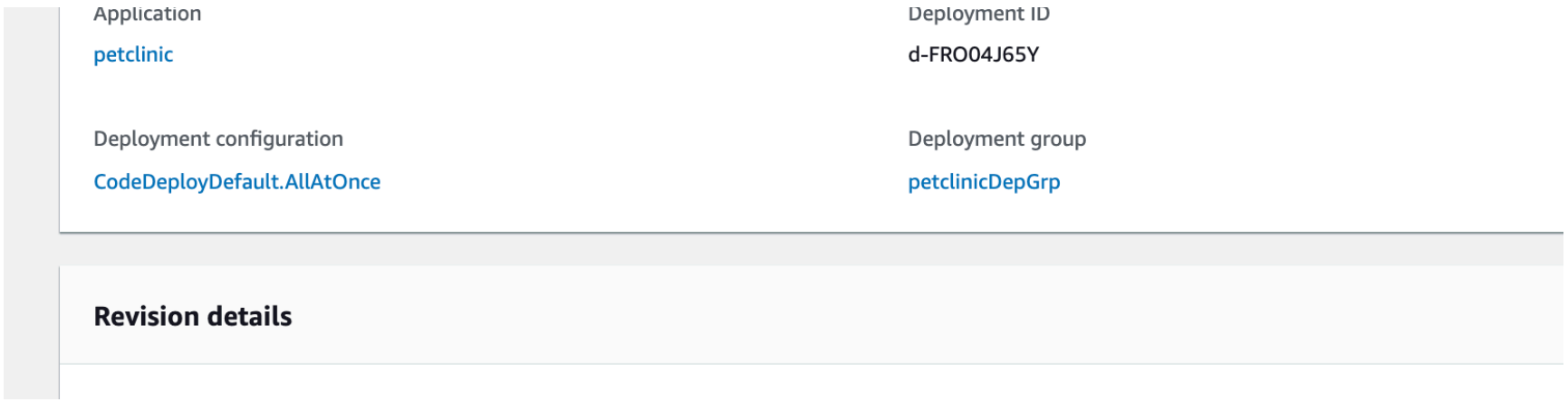


- Return to the CodeDeploy Console and Select Deployments
- Select the topmost Deployment and view the deployment progress



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


aws
Services
Resource Groups
Administrator @ cert-microcosm
Ohio
Support
Developer Tools
CodeDeploy
Source • CodeCommit
Build • CodeBuild
Deploy • CodeDeploy
Getting started
Deployments
Deployment
Applications
Deployment configurations
On-premises instances
Pipeline • CodePipeline
Feedback
Return to the old experience
Application
petclinic
Deployment ID
d-9GMQAH25Y
Status
In progress
Deployment configuration
custom1
Deployment group
petclinicDepGrp
Initiated by
user
Revision details
Revision location
s3://petclinicdeploy/target/#54-8254645117549694950.zip?eTag=beff67edee49a1f6645a1beeefb4d5fb
Revision created
2 minutes ago
Description
Application revision registered via Jenkins
Event
Duration
Status
Error code
Start time
End time
BeforeBlockTraffic
less than one second
Succeeded
Apr 26, 2019 11:11 AM
Apr 26, 2019 11:11 AM
BlockTraffic
Pending
AfterBlockTraffic
Pending
ApplicationStop
Pending
DownloadBundle
Pending
BeforeInstall
Pending
Install
Pending
AfterInstall
Pending
ApplicationStart
Pending
ValidateService
Pending
BeforeAllowTraffic
Pending
AllowTraffic
Pending
AfterAllowTraffic
Pending

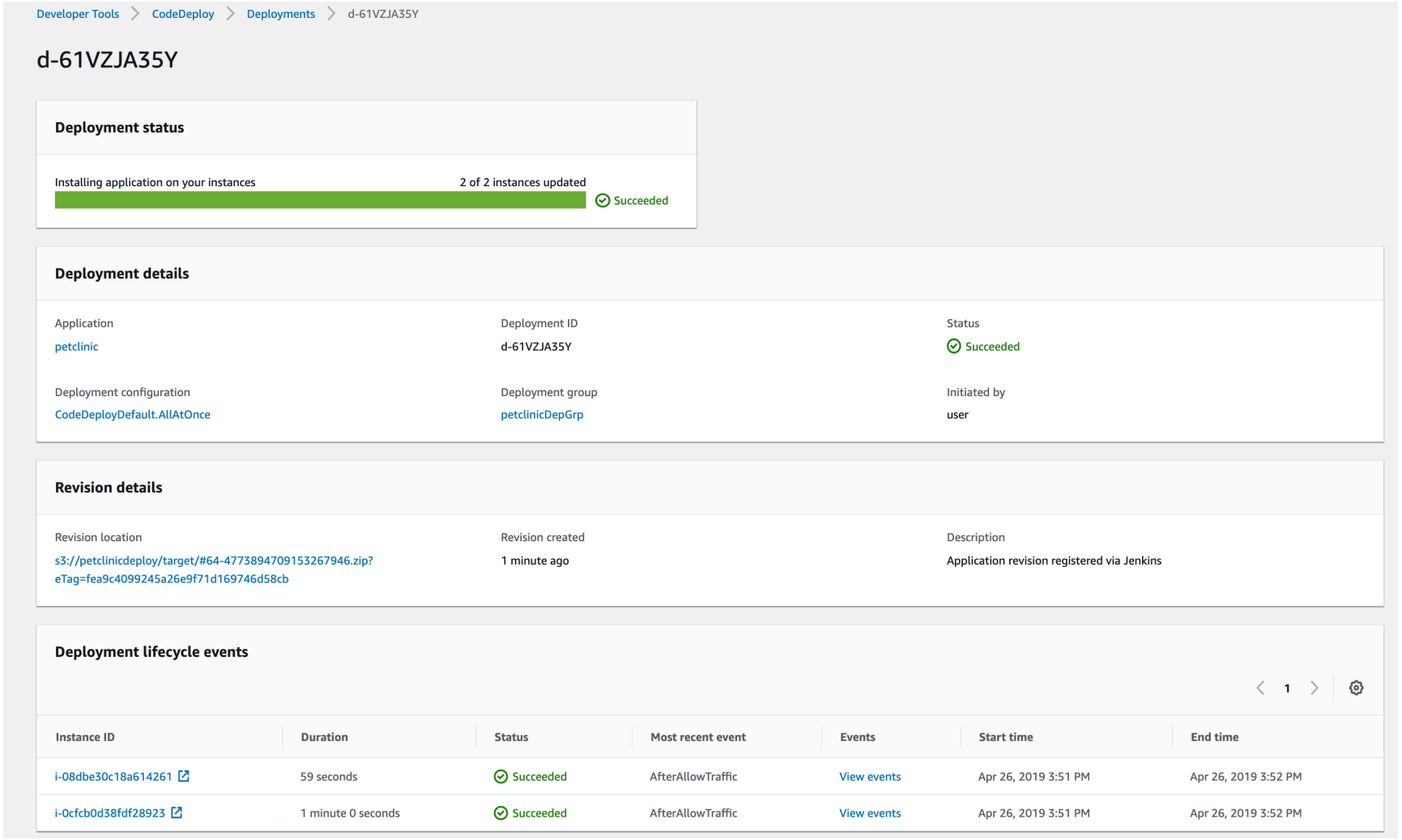

Application
petclinic
Deployment ID
d-FRO04J65Y
Deployment configuration
CodeDeployDefault.AllAtOnce
Deployment group
petclinicDepGrp
Revision details

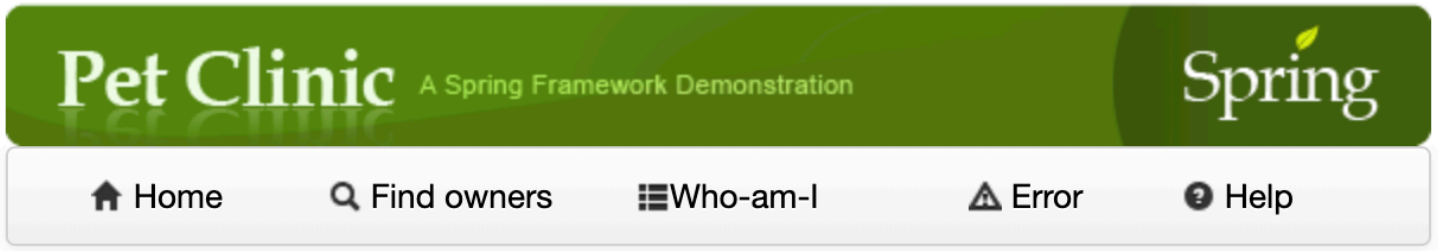
Revision location	Revision created
s3://petclinicdeploy/target/#62-2472083856417875777.zip?eTag=5c72ef1adb2c717b3611effbefe97645	33 minutes ago

Event	Duration	Status	Error code
BeforeBlockTraffic	less than one second	Succeeded	-
BlockTraffic	22 seconds	Succeeded	-
AfterBlockTraffic	less than one second	Succeeded	-
ApplicationStop	less than one second	Succeeded	-
DownloadBundle	less than one second	Succeeded	-
BeforeInstall	less than one second	Succeeded	-
Install	less than one second	Succeeded	-
AfterInstall	less than one second	Succeeded	-
ApplicationStart	less than one second	Succeeded	-
ValidateService	less than one second	Succeeded	-
BeforeAllowTraffic	less than one second	Succeeded	-
AllowTraffic	29 minutes 21 seconds	Succeeded	-
AfterAllowTraffic	less than one second	Succeeded	-

- Once the Deployment is successful, view your deployed project - Navigate to the EC2 Instance IP at port 80 to

view the Tomcat management console, and navigate to the [EC2 Instance Ip]/petclinic to view the deployed Java Project





Welcome

Return to Table of Contents

AWS ECS Based Code Deploy

Additional Task Definitions

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

Create EC2 ECS AIM

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

Create Petclinic ECS Service

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

Set up CodeDeploy Application

Create Build Step in Jenkins

To DO

Create Cluster

Create Task definition

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

Initialize Service

Create Role

Create Load Ballancer

Create CodeDeploy -> Deploy -> Application

http://[petclinic IP]:8080/petclinic

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

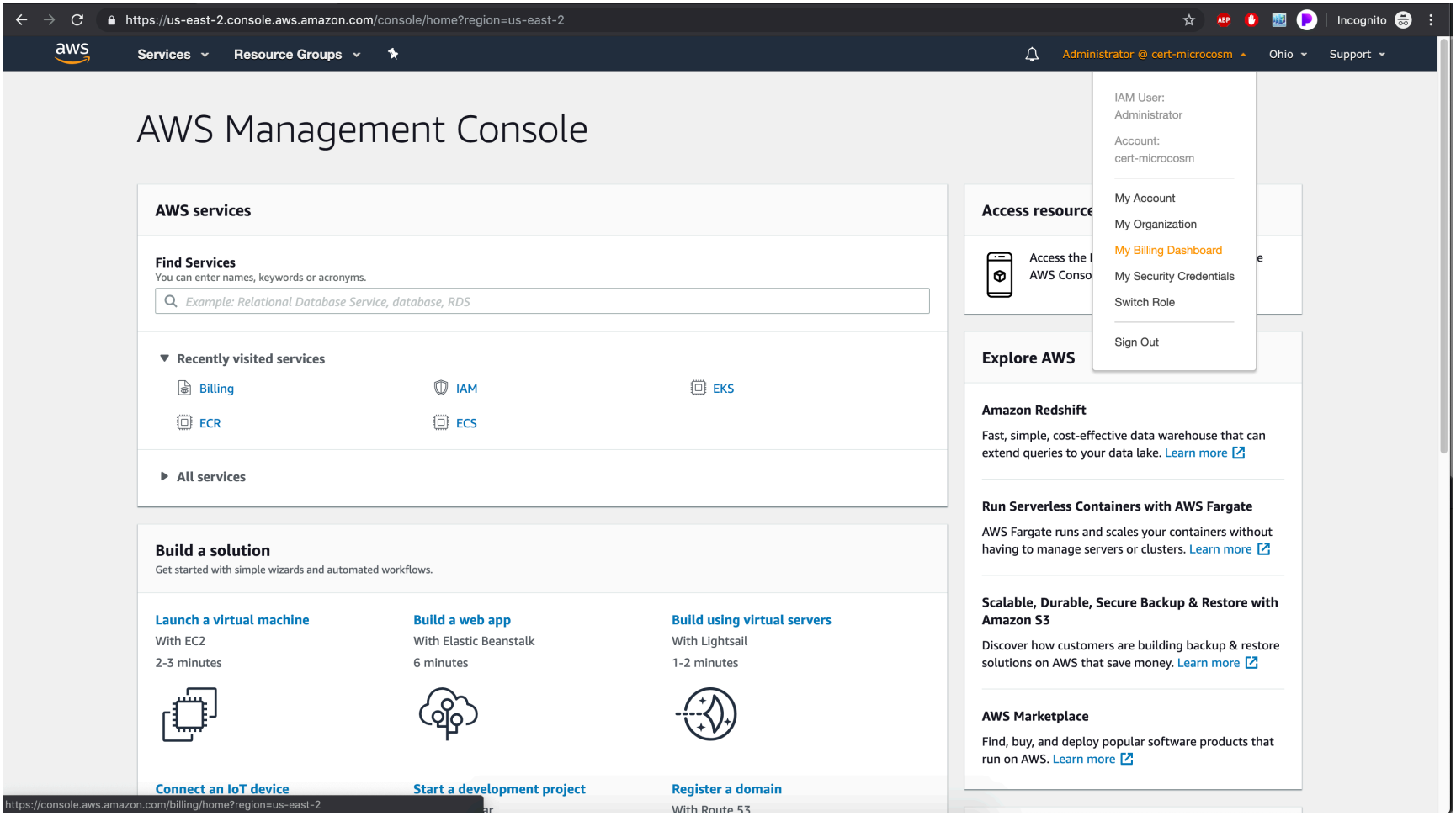
object key if the path within the bucket eg target/petclinic.war

Return to Table of Contents

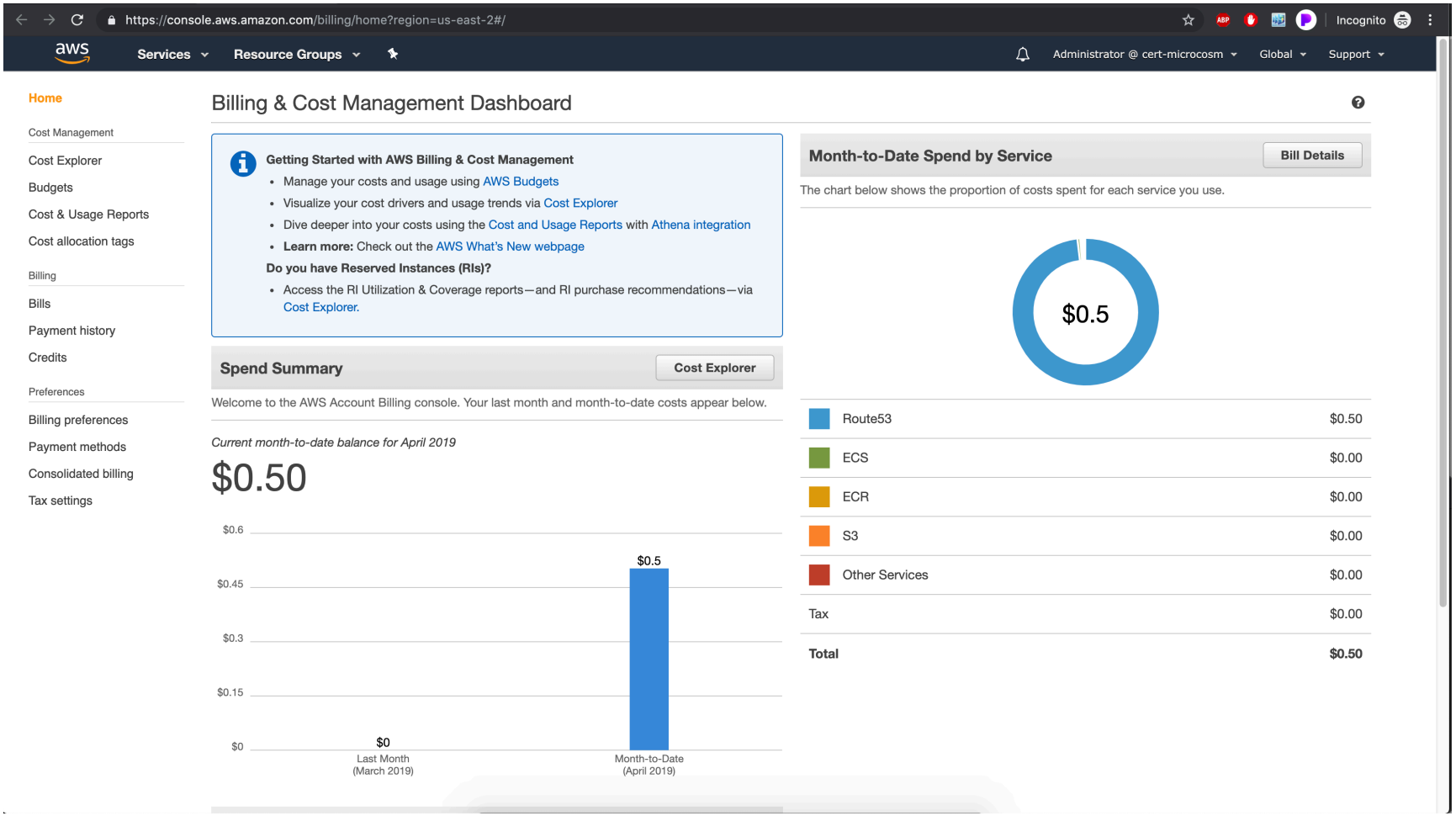
APPENDIX A

Billing

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



- Explore your billing information, current charges and projections



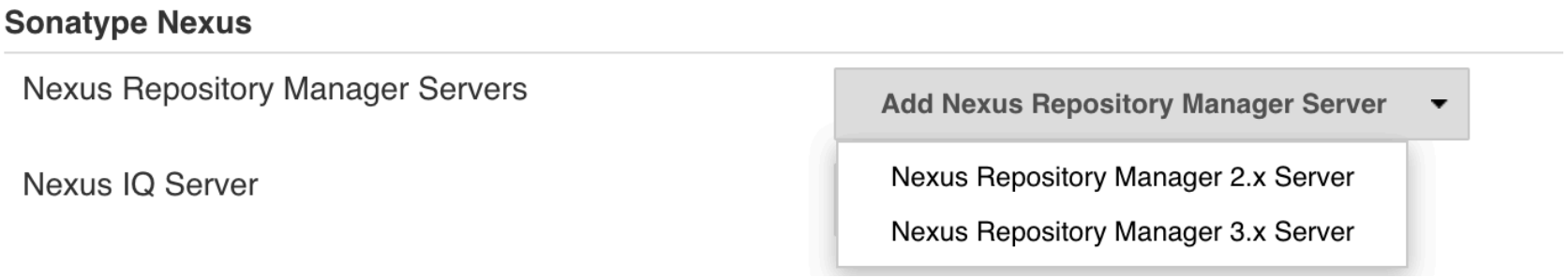
Return to Table of Contents

Add Sonatype Nexus build step

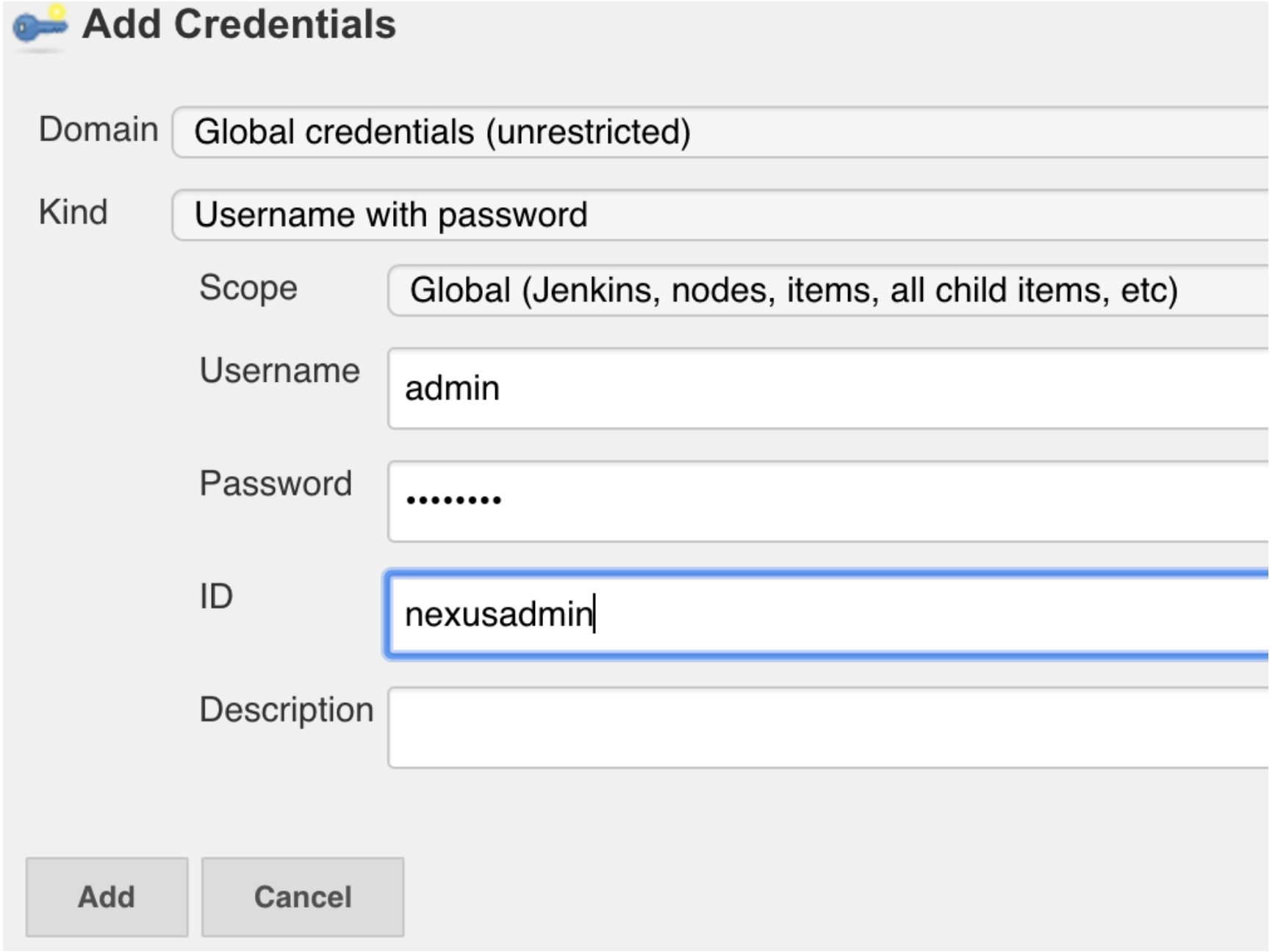
Use the following instructions to configure Jenkins to connect to Nexus Repository Manager:

- Select Manage Jenkins from the Dashboard's left-navigation menu

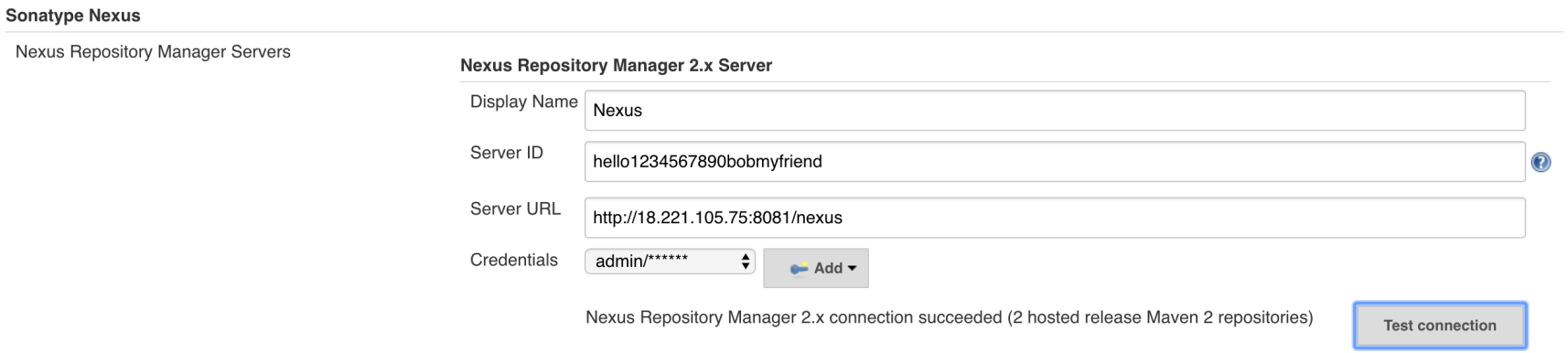
- Select Configure System from the list of configuration options
- In the Sonatype Nexus section, click the Add Nexus Repository Manager Server dropdown menu and then select Nexus Repository Manager 2.x Server.



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


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

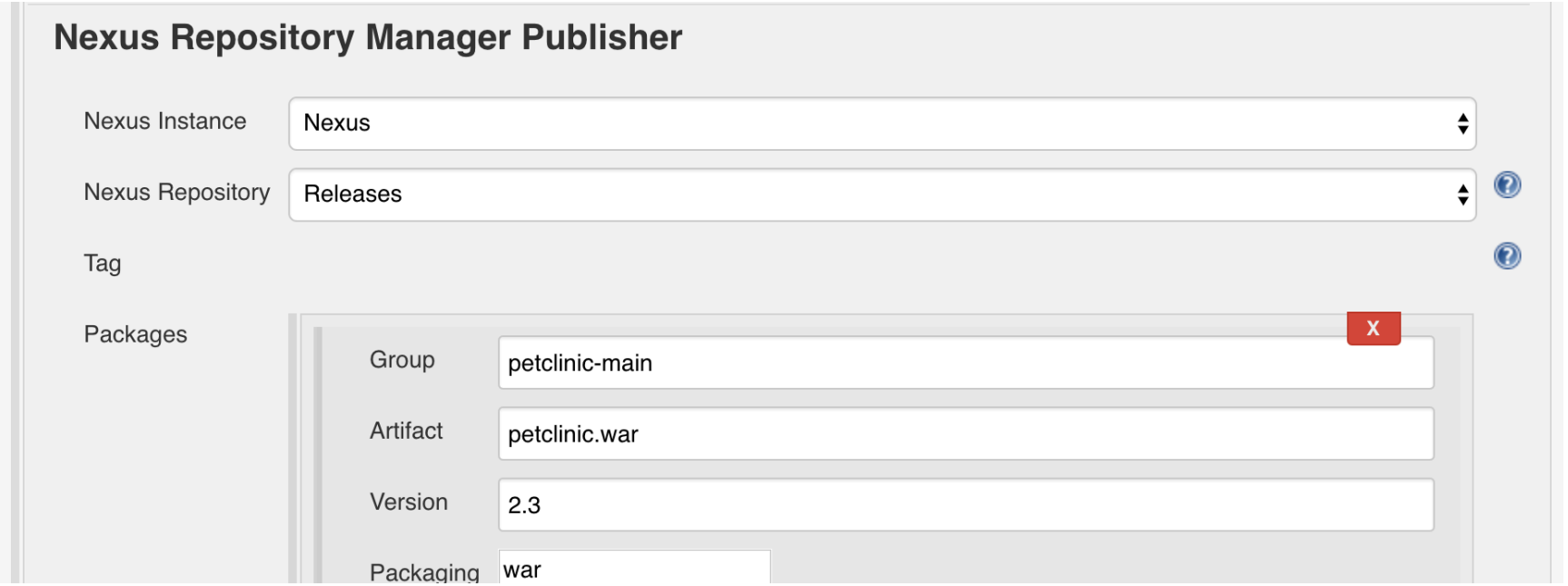
- Enter the following:
  - Display Name: Name of the server you want shown when selecting Nexus Repository Manager instances for build jobs
  - Server ID: A unique ID used to reference Nexus Repository Manager in Build Pipeline scripts. It should be alphanumeric without spaces (eg make up a string like hello1234567890bobmyfriend)
  - Server URL: Location of your Nexus Repository Manager server (ex: http://**[NEXUS_IP_ADDRESS]**:8081/nexus)
  - 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

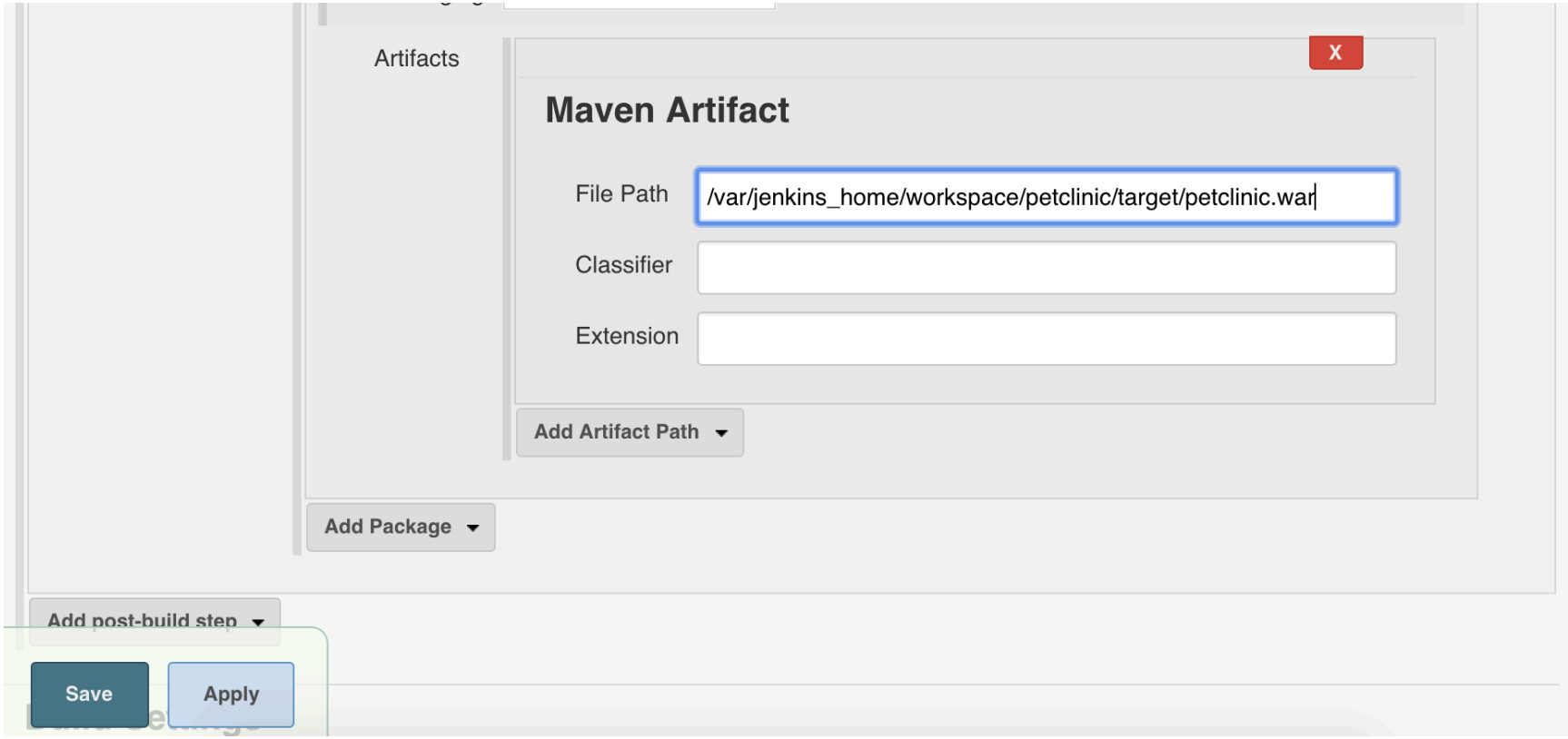


- Return to the PetClinic build configuration page
- Select the Post Steps Tab and click Add post-build step
- Select Nexus Repository Manager Publisher

    - 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

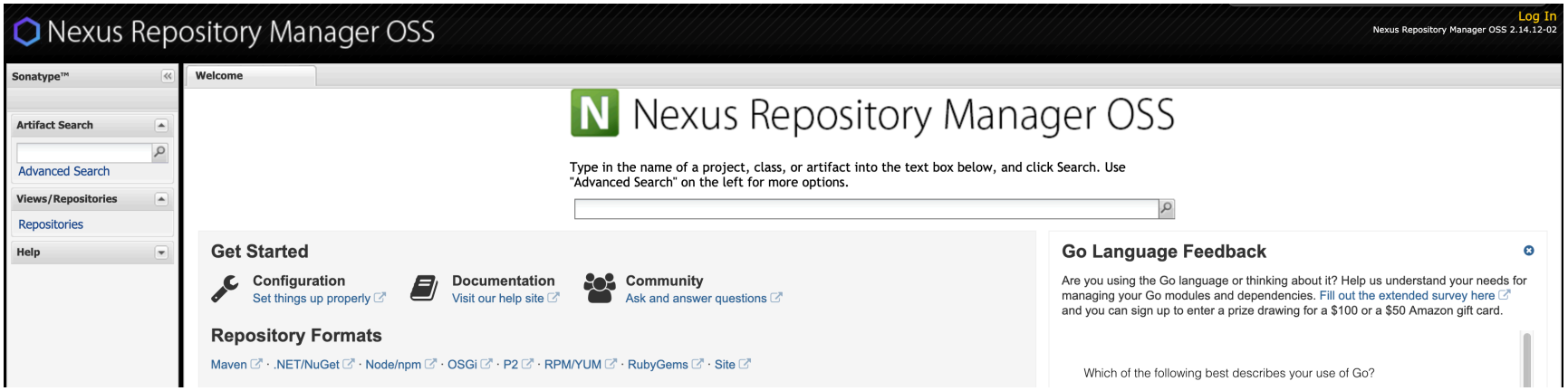




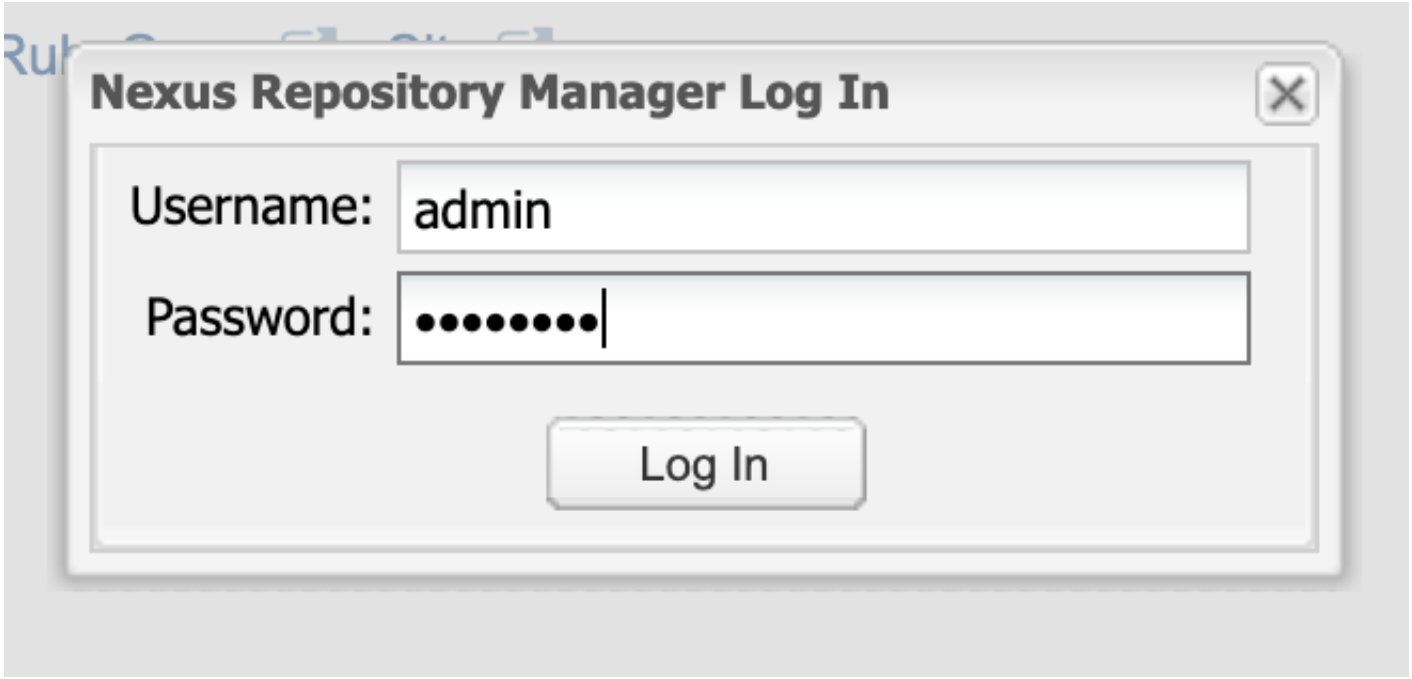
- After a successful Jenkins build, if you look in the build console output, you should see, near the bottom of the page, a message indicating success, similar to:

```
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
```

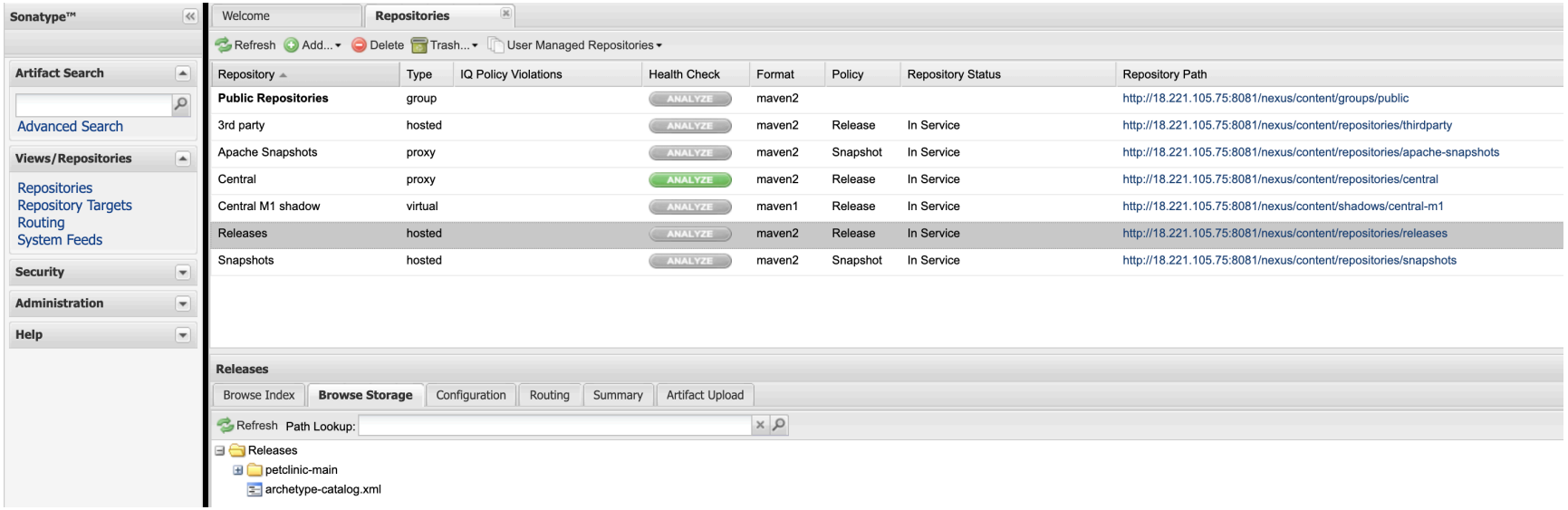
- To verify navigate to the Nexus Repository manager web UI (http://**[NEXUS_IP_ADDRESS]**:8081/nexus)



- Login using the default credentials admin:admin123



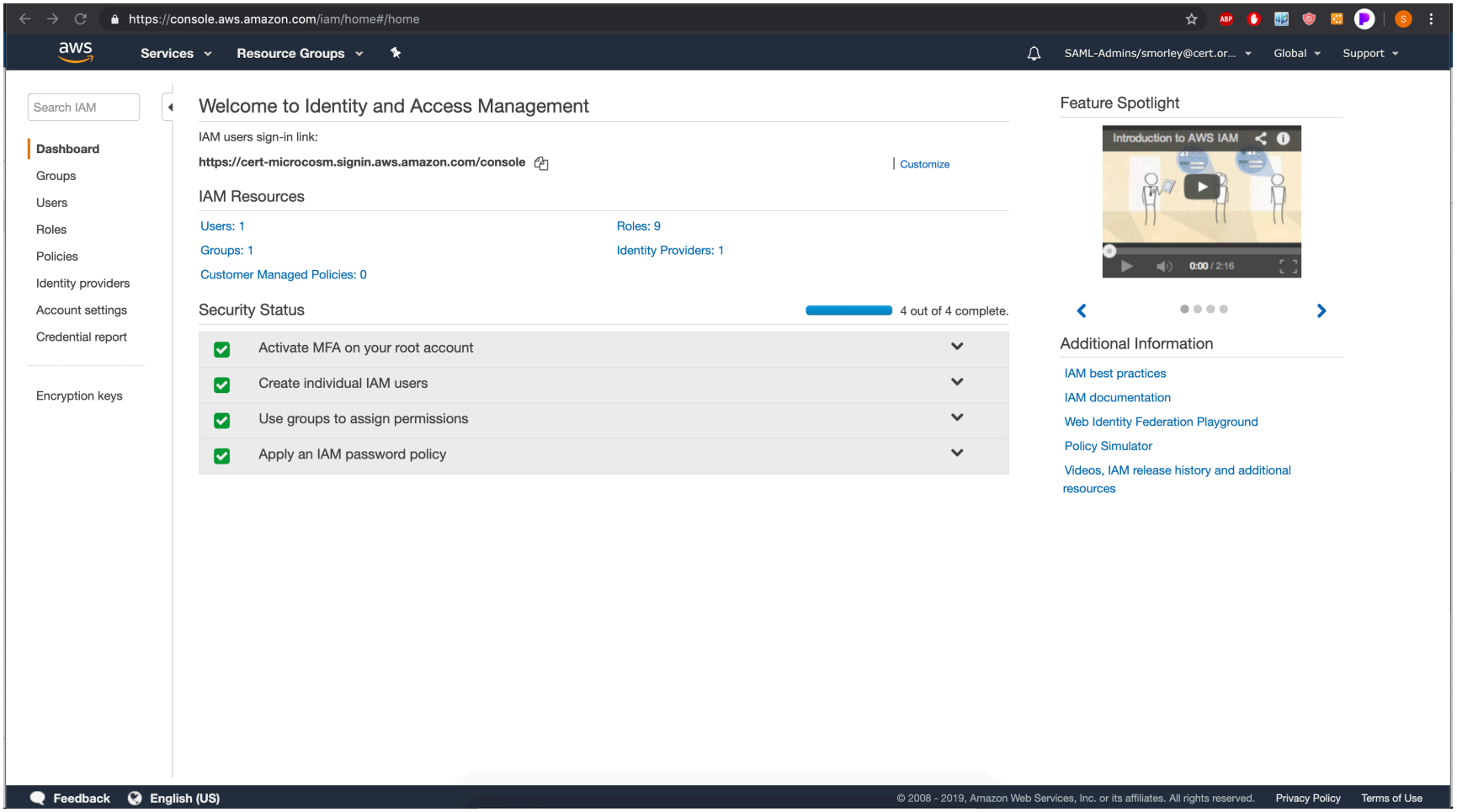
- View your selected packages in the under the "Releases" repository



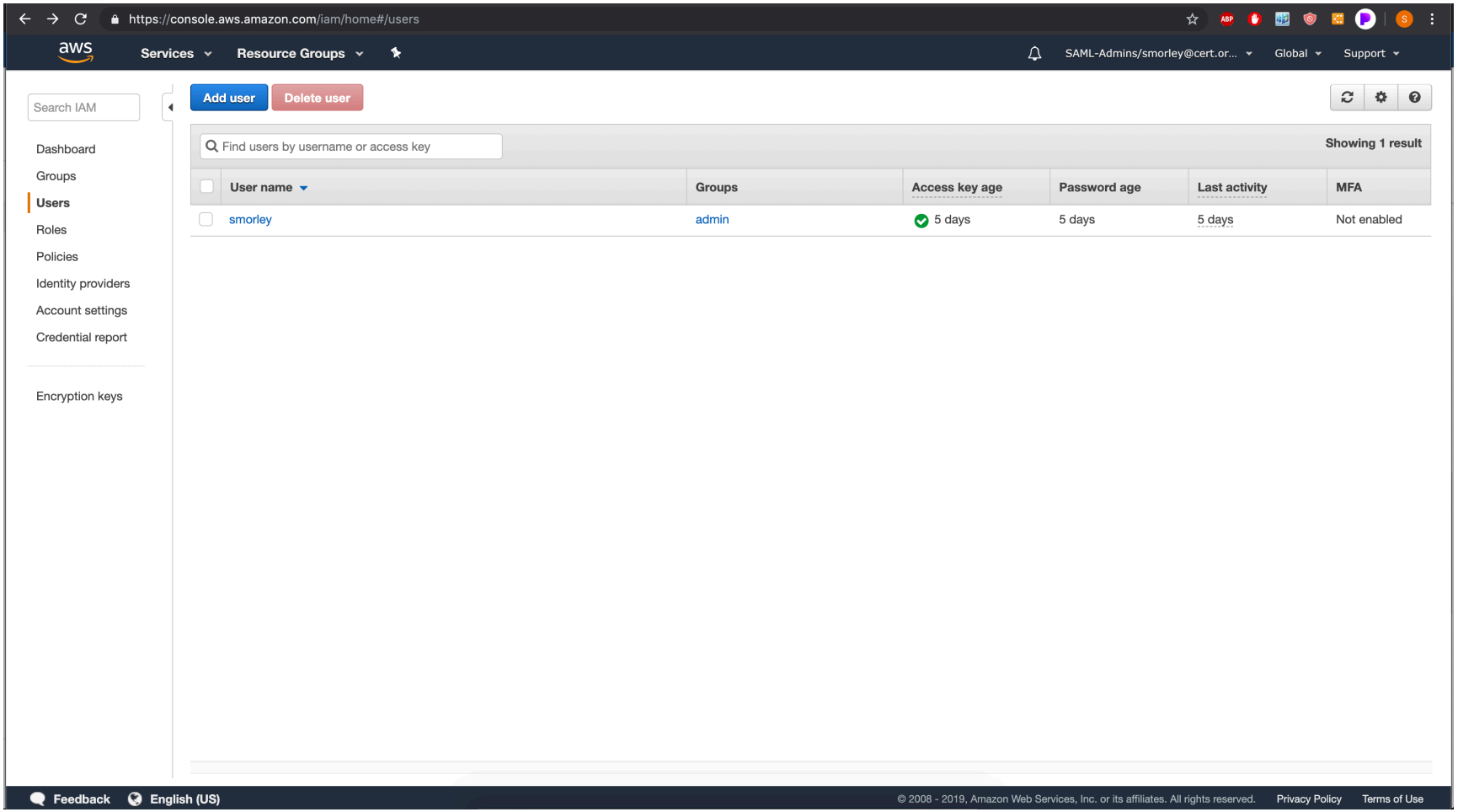
Return to Table of Contents

Creating Identity and Access Management (IAM) Users

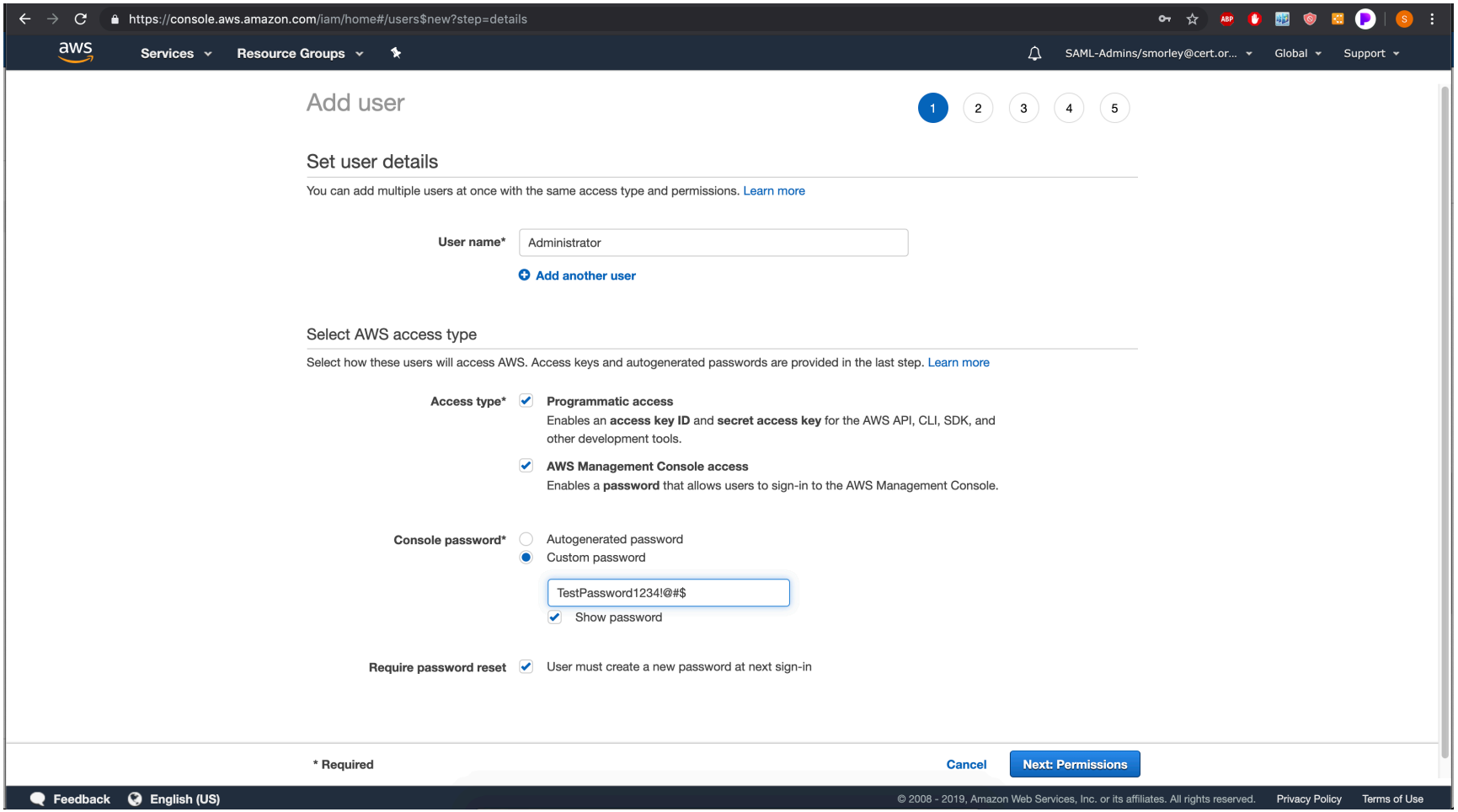
- From the Services drop down on the top left, select IAM to get to the IAM Dashboard



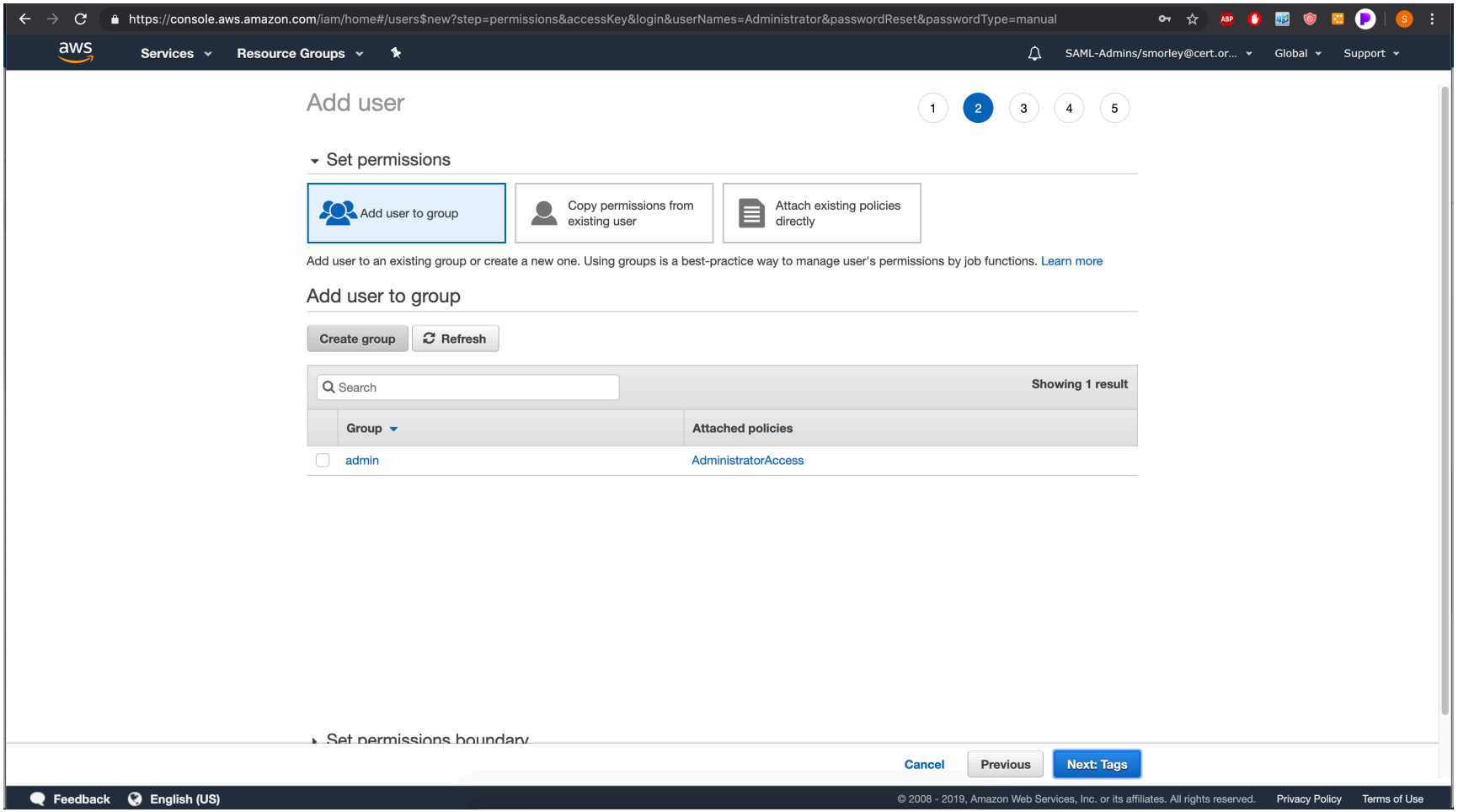
- Select Users on the Left
- Click Add User



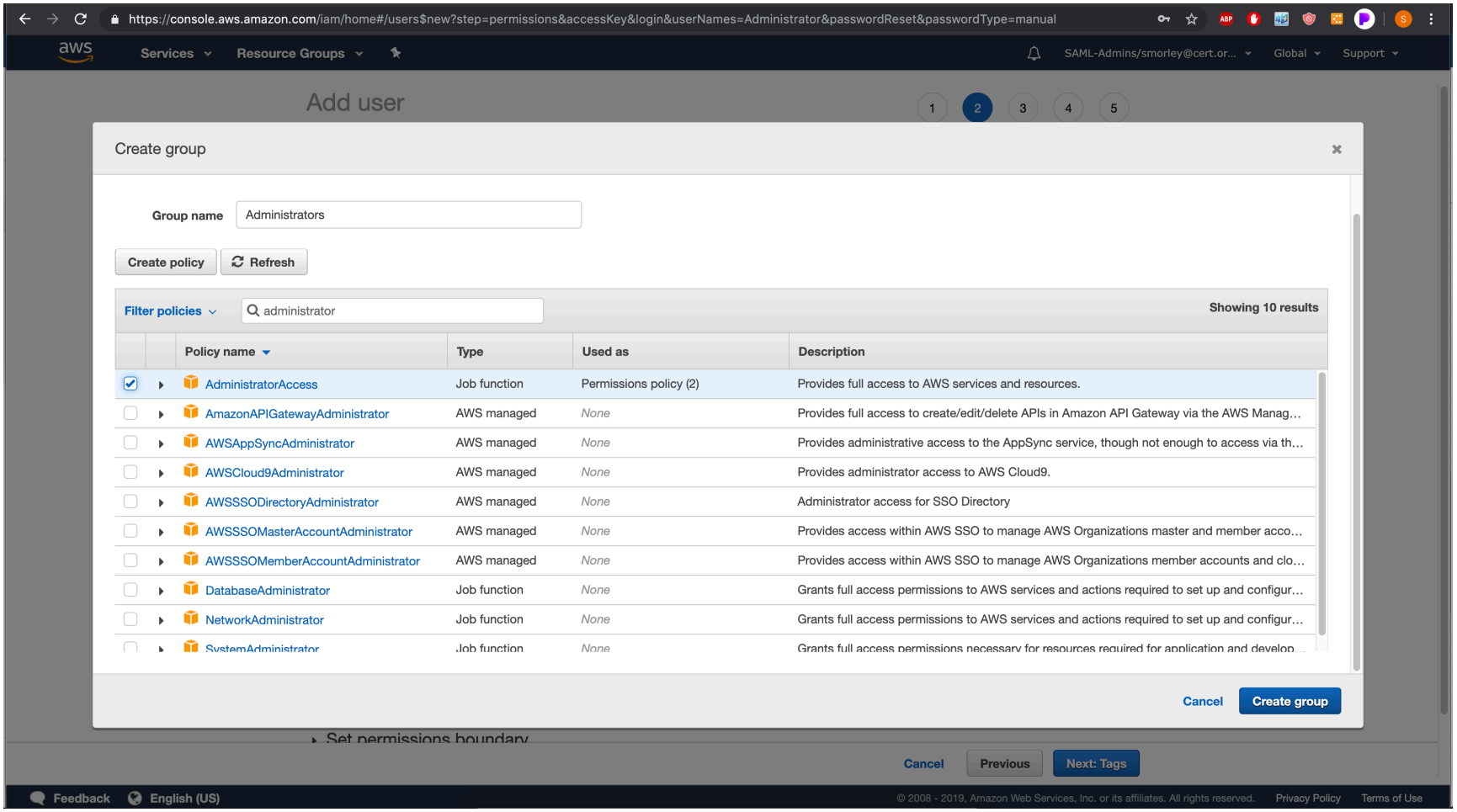
- Enter the username and custom password
- Check Programmatic Access if this user will need CLI or API access
- Check AWS management Console Access
- Check require password reset to require user to create a secret password
- Click Next: Permissions



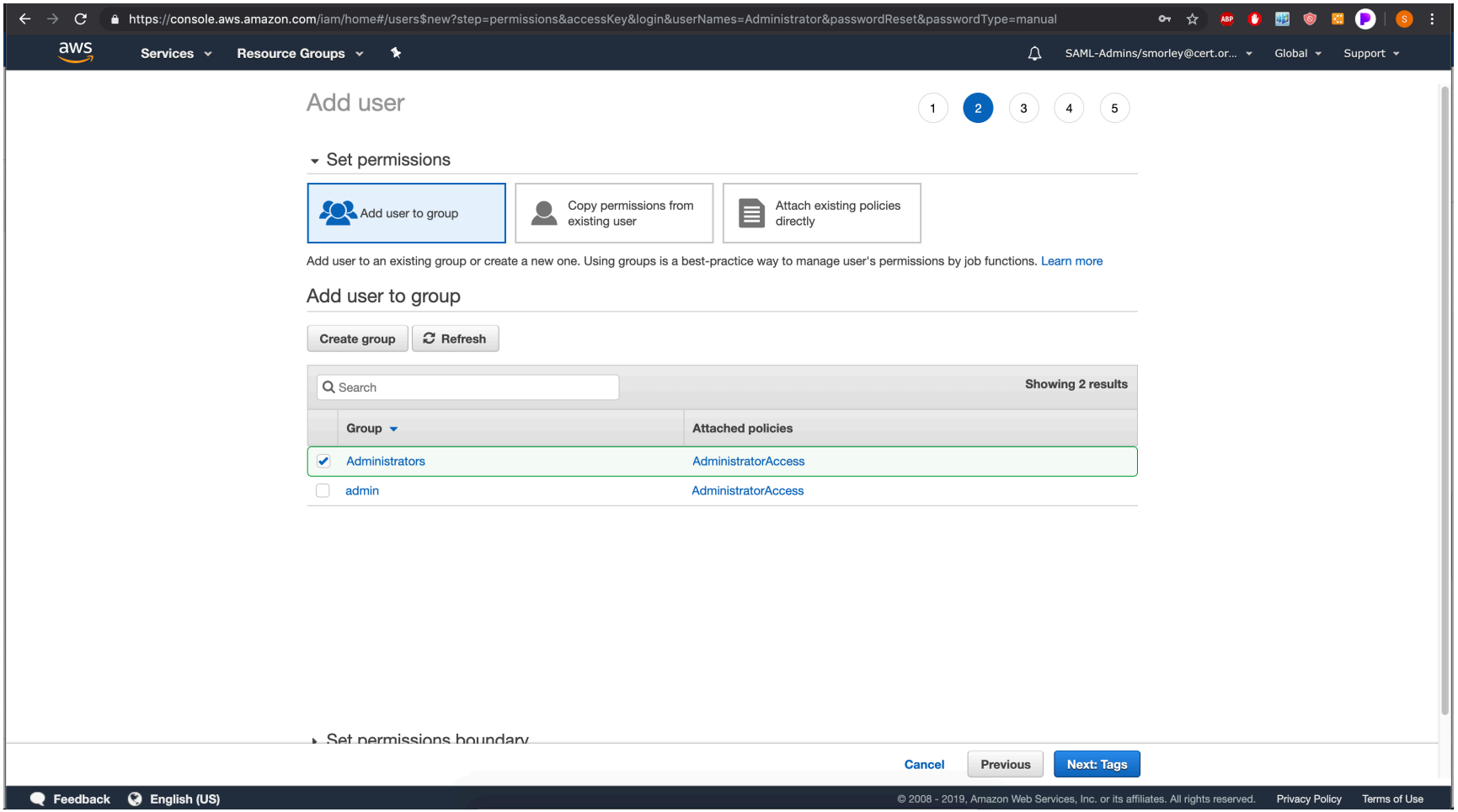
- Click add user to Group (or Copy Permissions from and existing user, if applicable)
- If the desired group does not yet exist, select Create Group



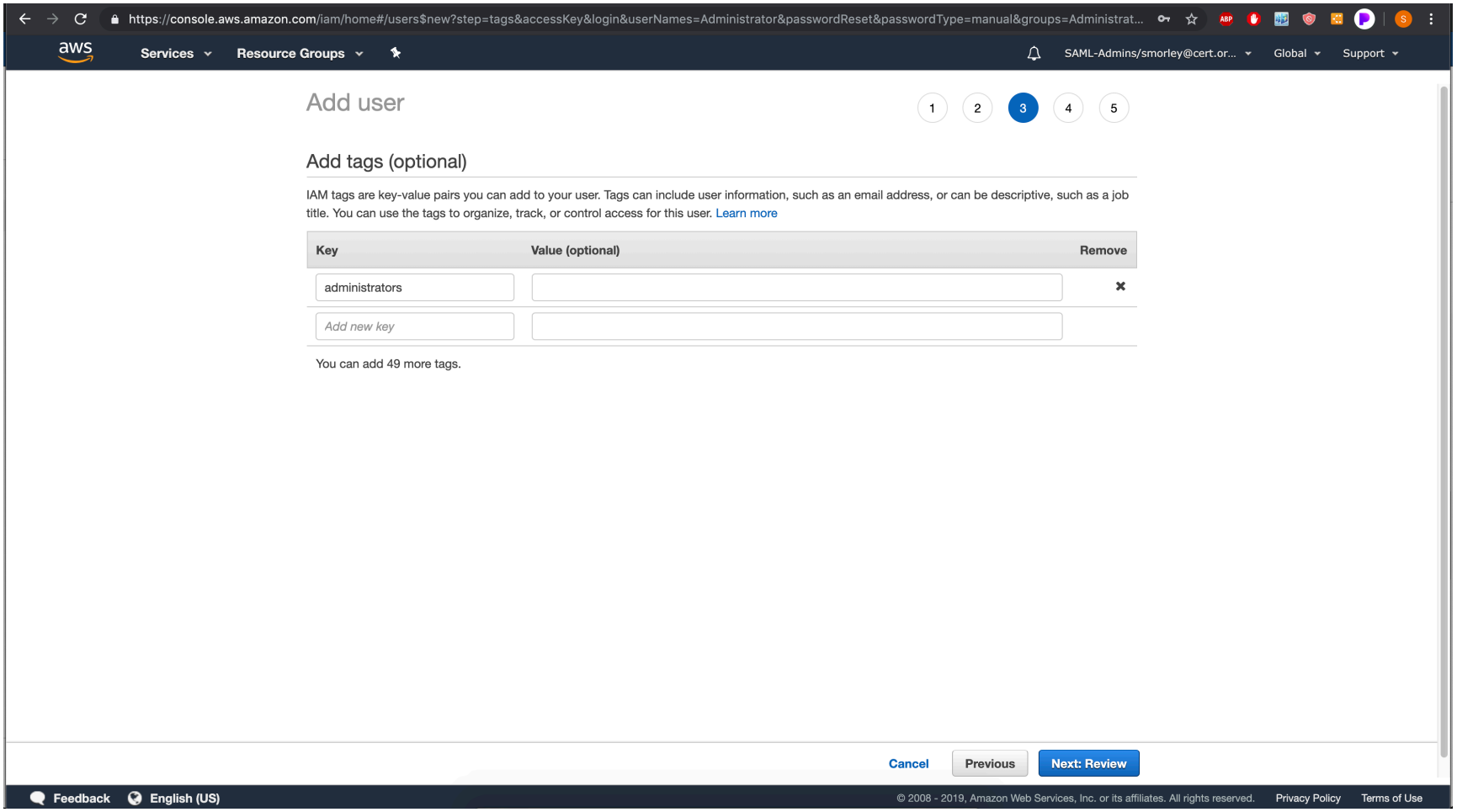
- Name your Group
- Select the permissions appropriate for the group
  - To create a user capable of any actions within AWS, filter on Administrator, and select AdministratorAccess
- Select Create Group



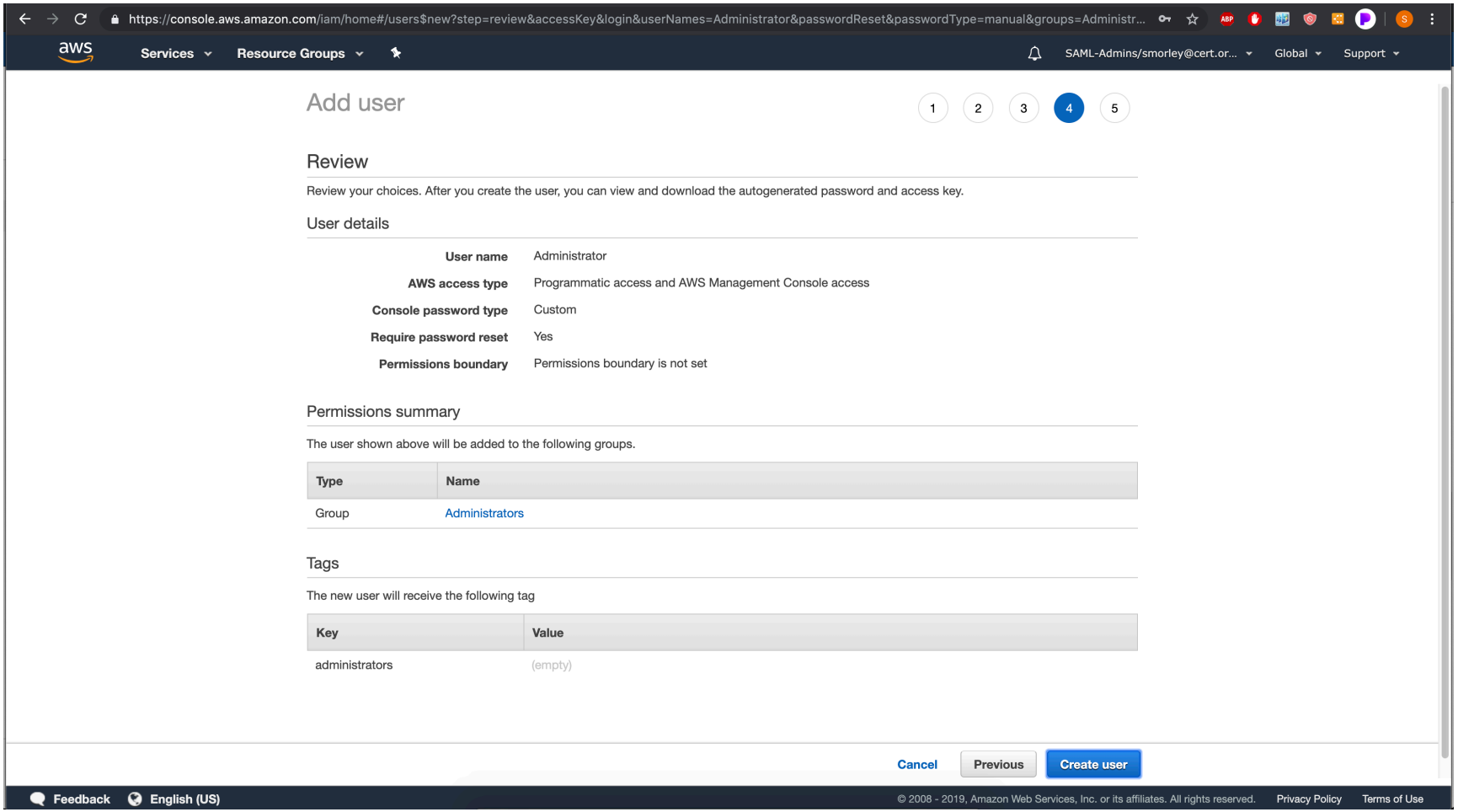
- Select the created group
- Press Next:Tags



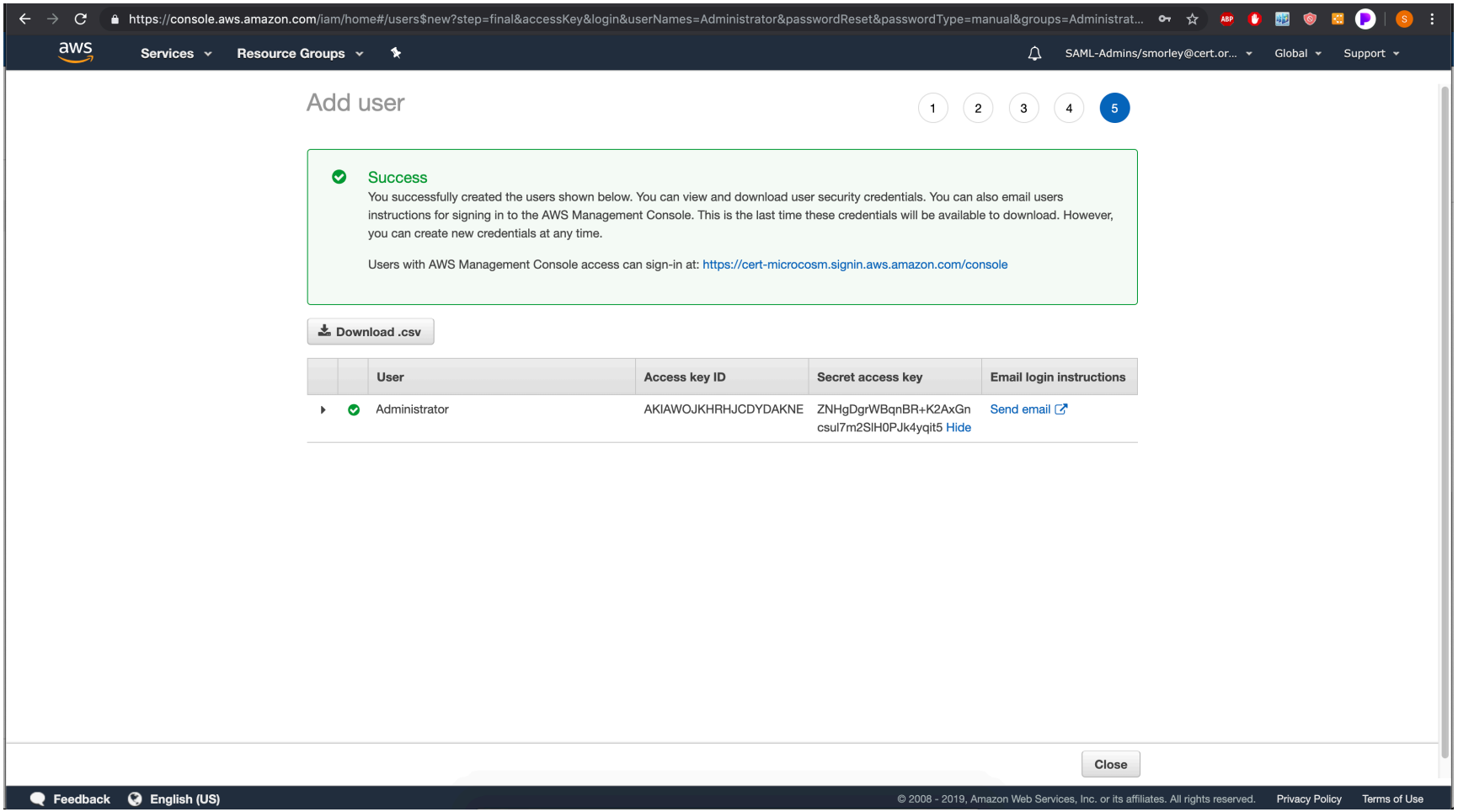
- Add tags if desired
- Click Review



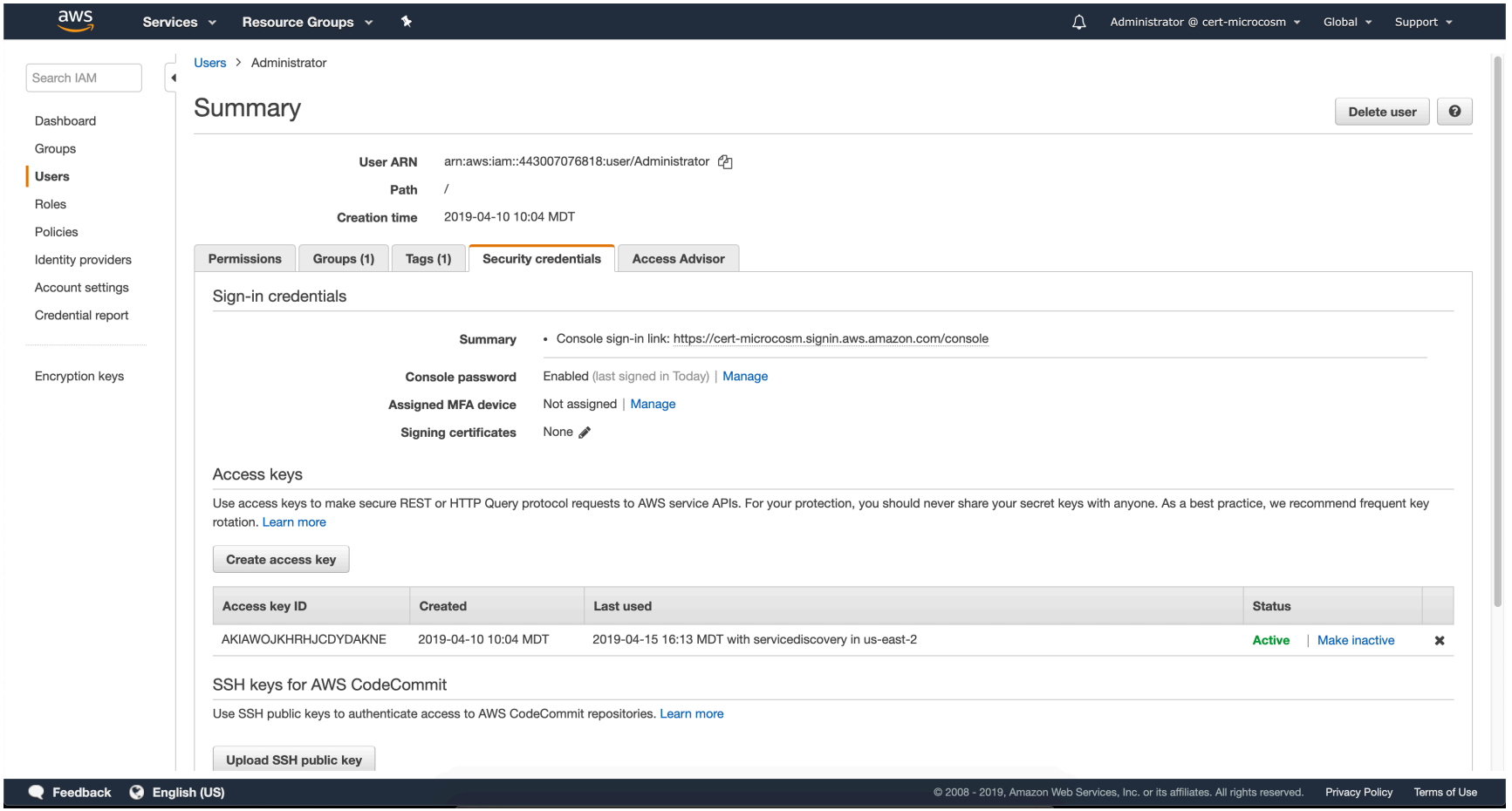
- After reviewing, click Create User



- 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



- 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.



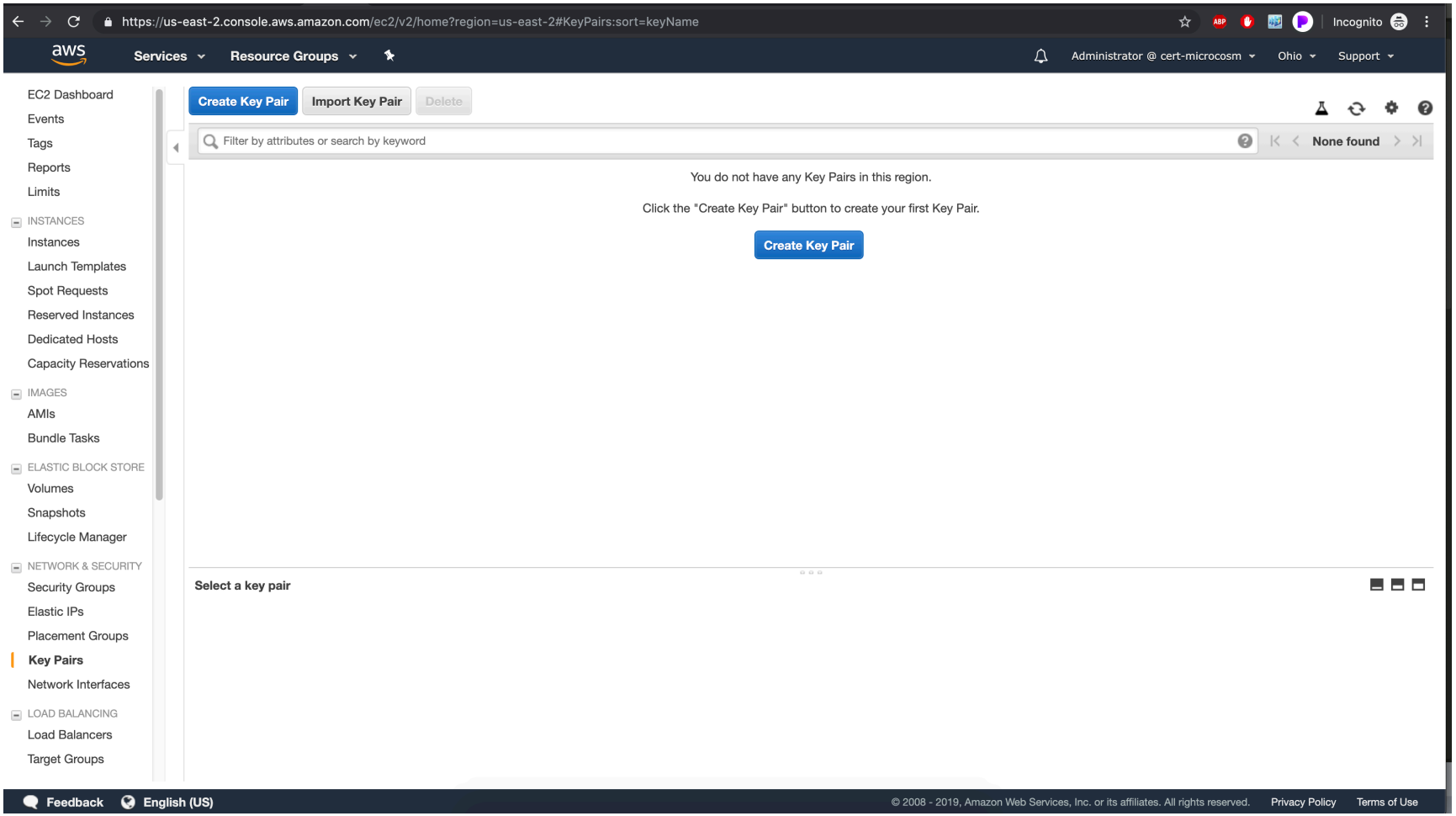
Return to Table of Contents

Create Key Pair

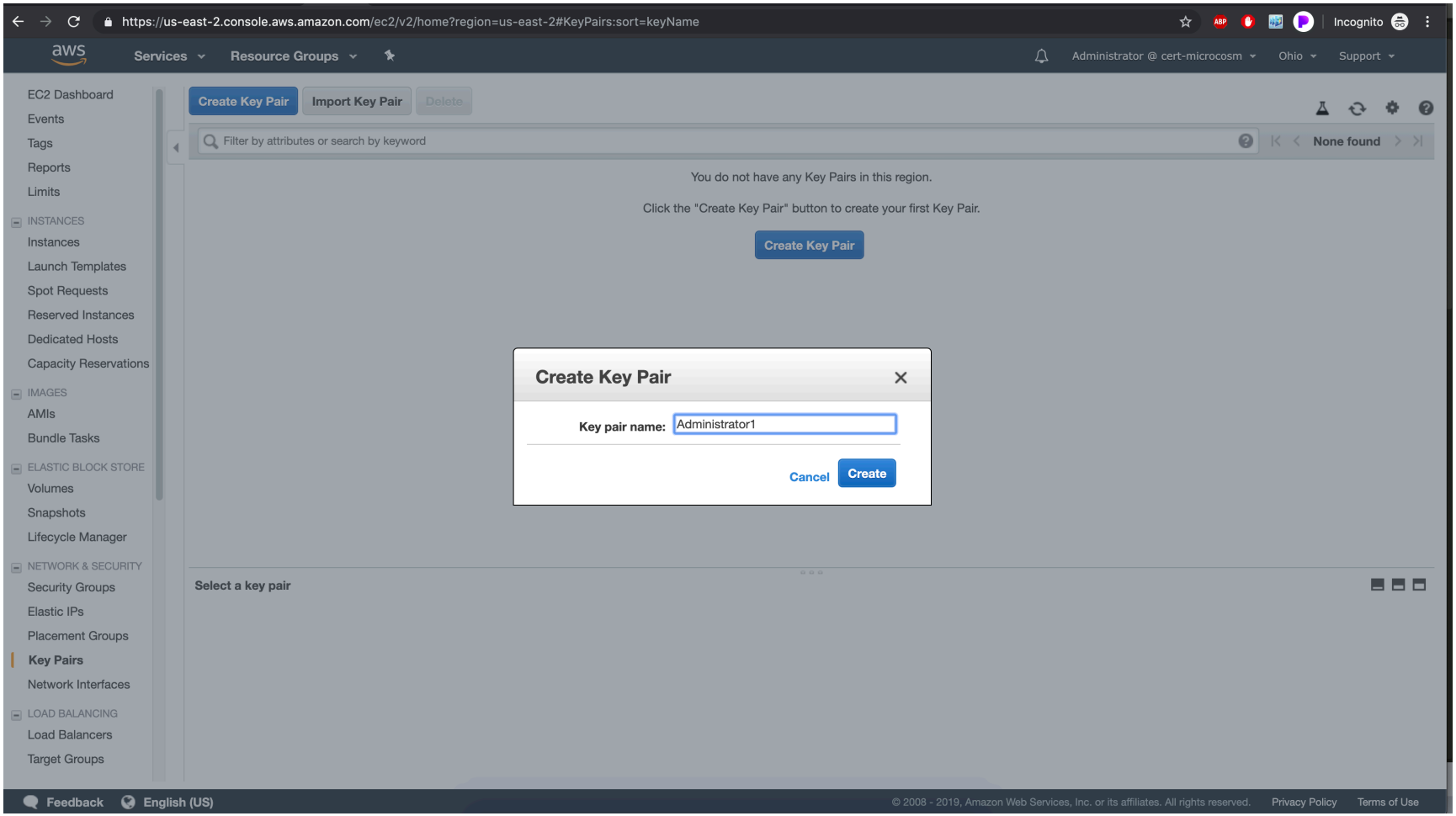
Key pairs are used for SSH and other authentication with AWS Amazon Machine Image instances (AMIs).

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

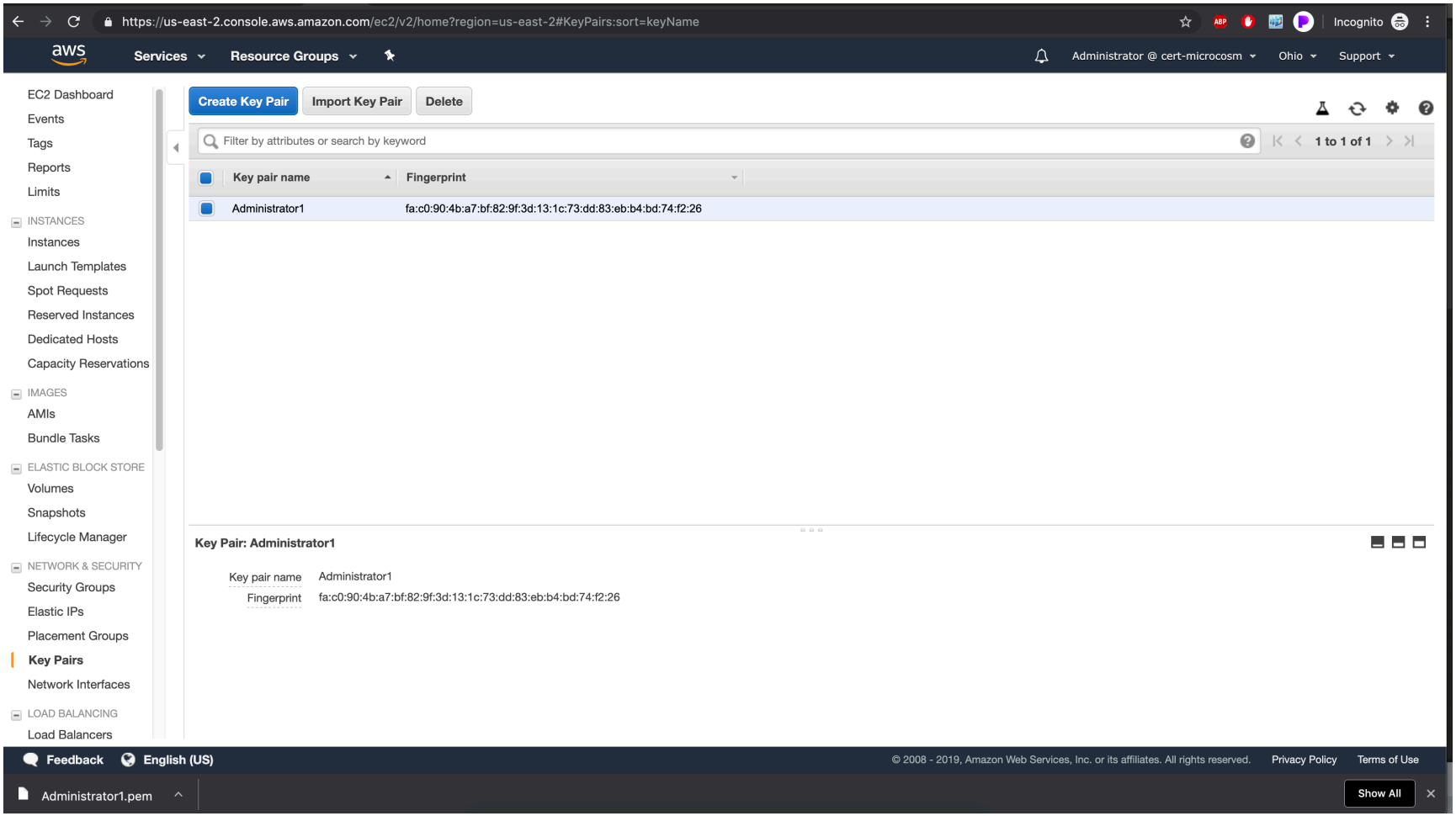
- Select Create Key Pair



- 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.



- View key pair details



Return to Table of Contents

Creating Roles

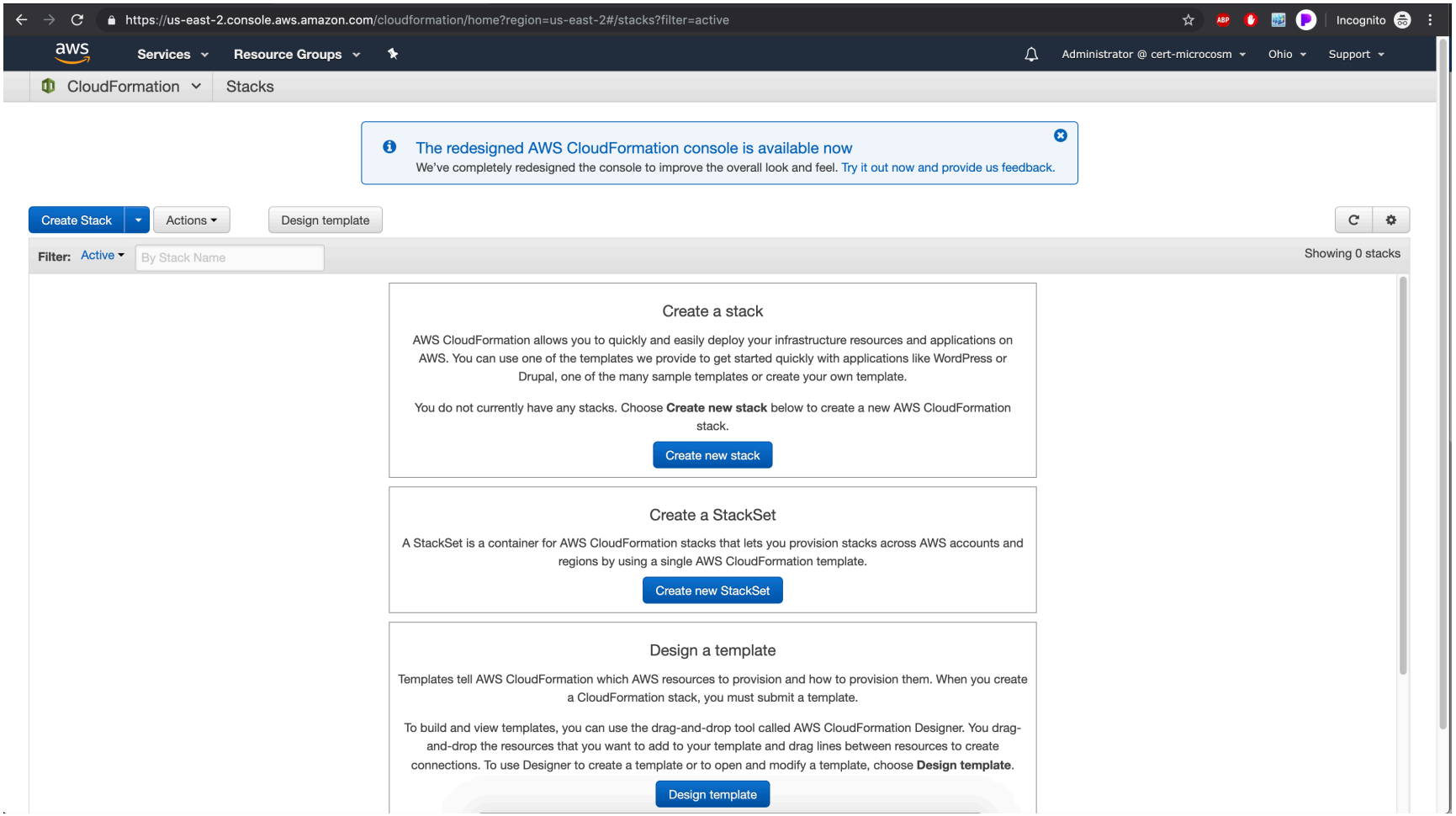
See the instructions in the Create IAM Role for EC2 Based CodeDeploy section.

Return to Table of Contents

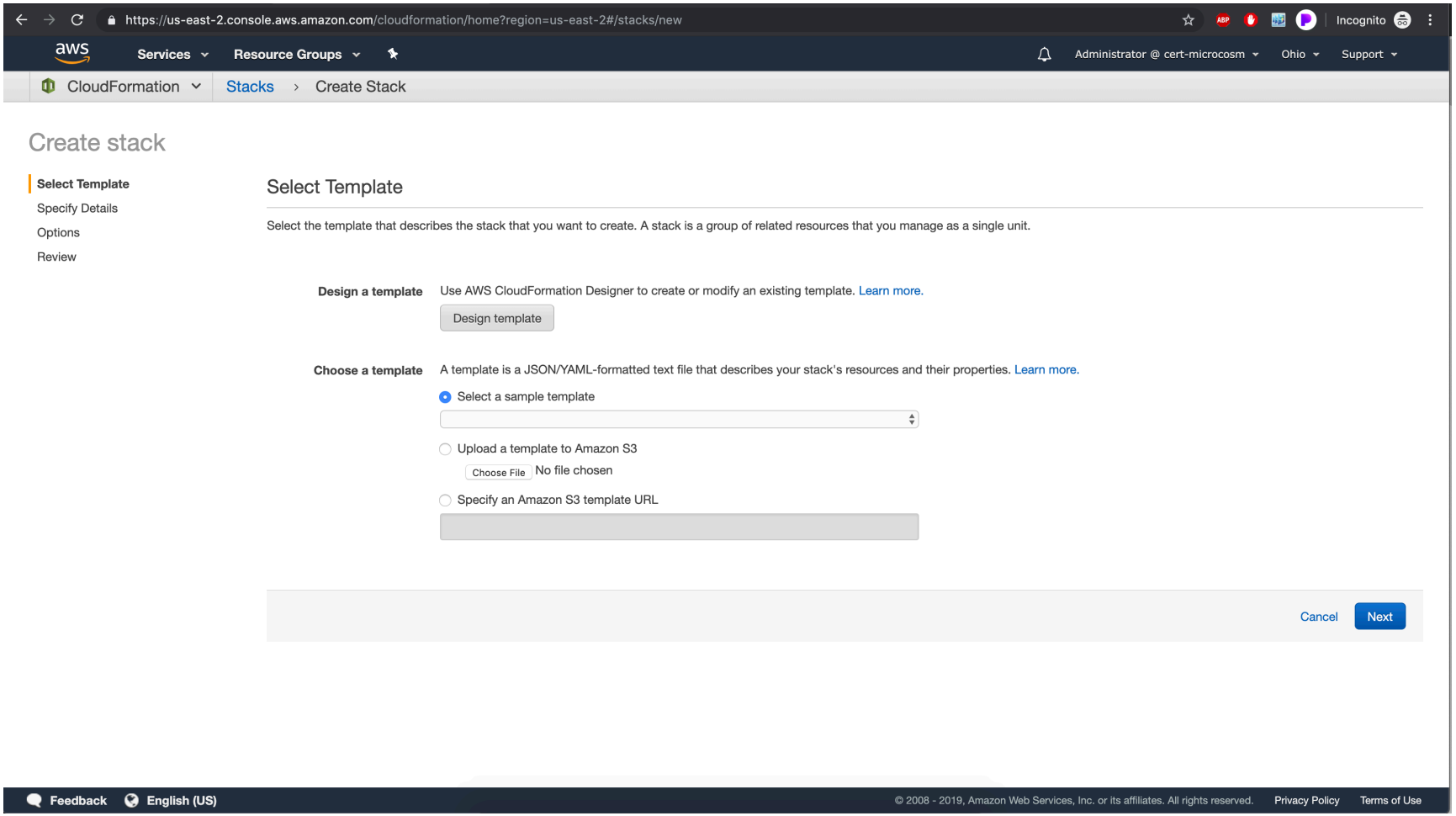
Create VPCs and user Accounts using CloudFormation Templates

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

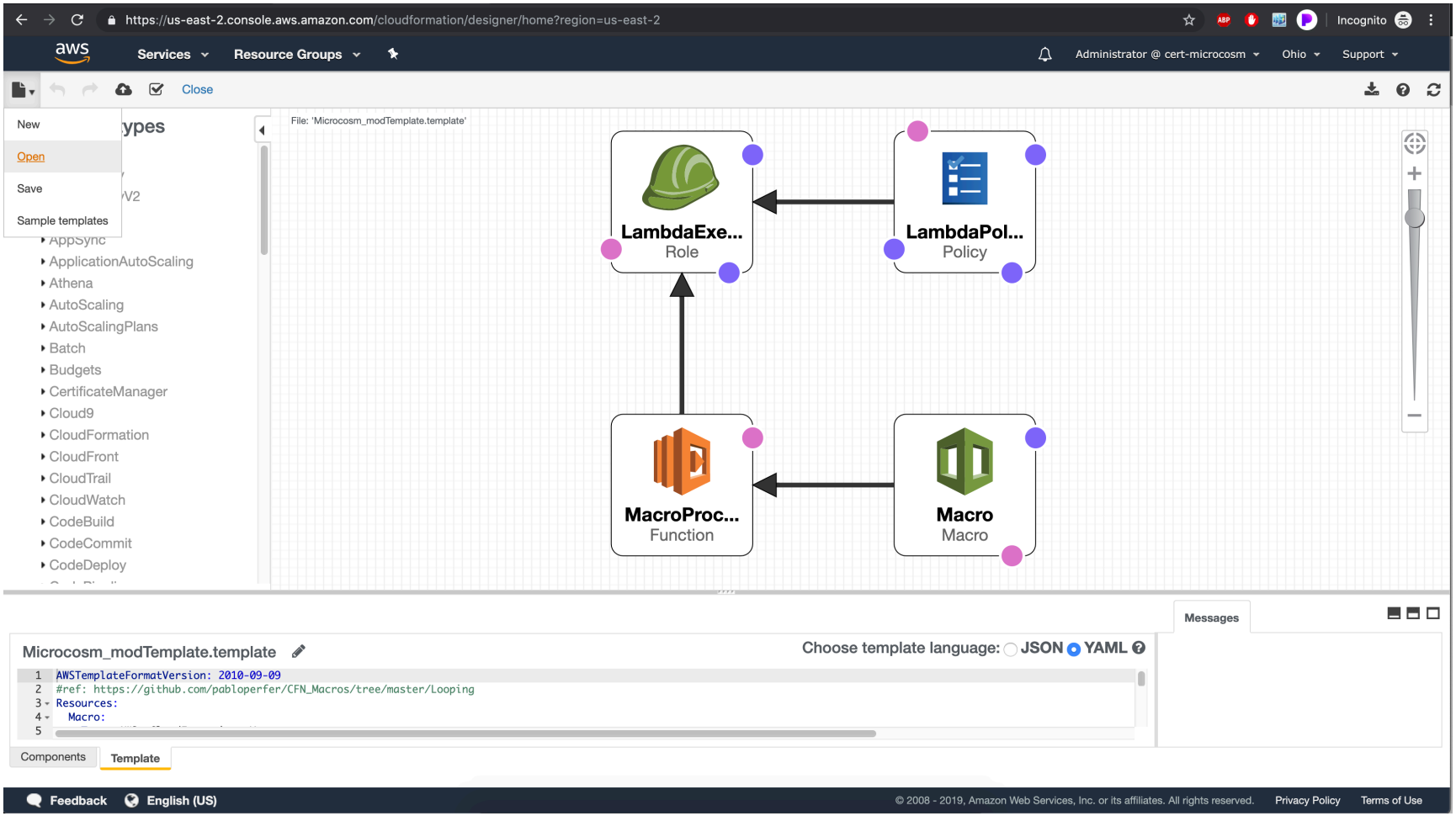
- Using the Services dropdown at the top on most AWS pages, navigate to CloudFormation. and select Create Stack



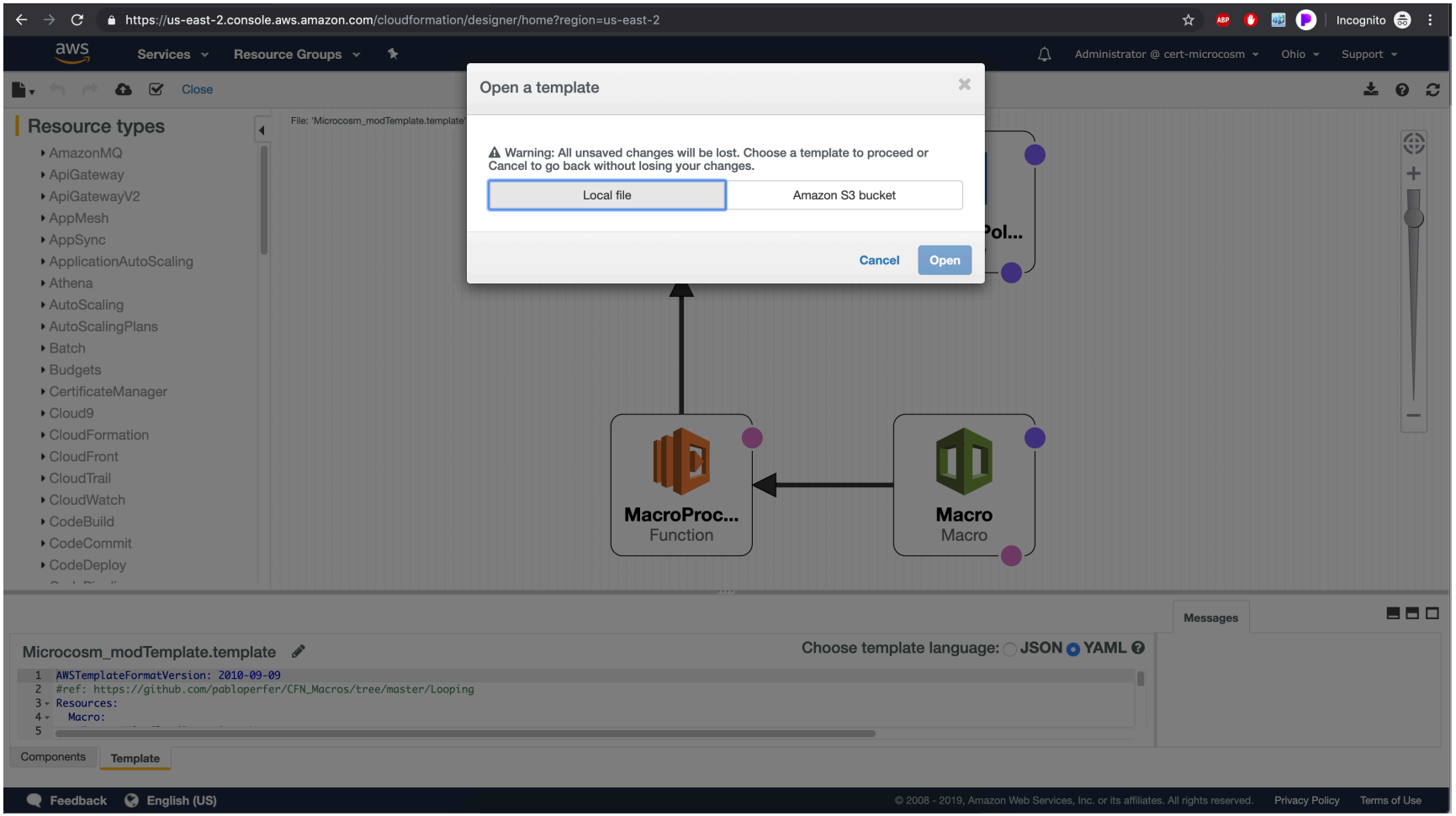
- To create or view a template, on the Select Template page, click Design Template



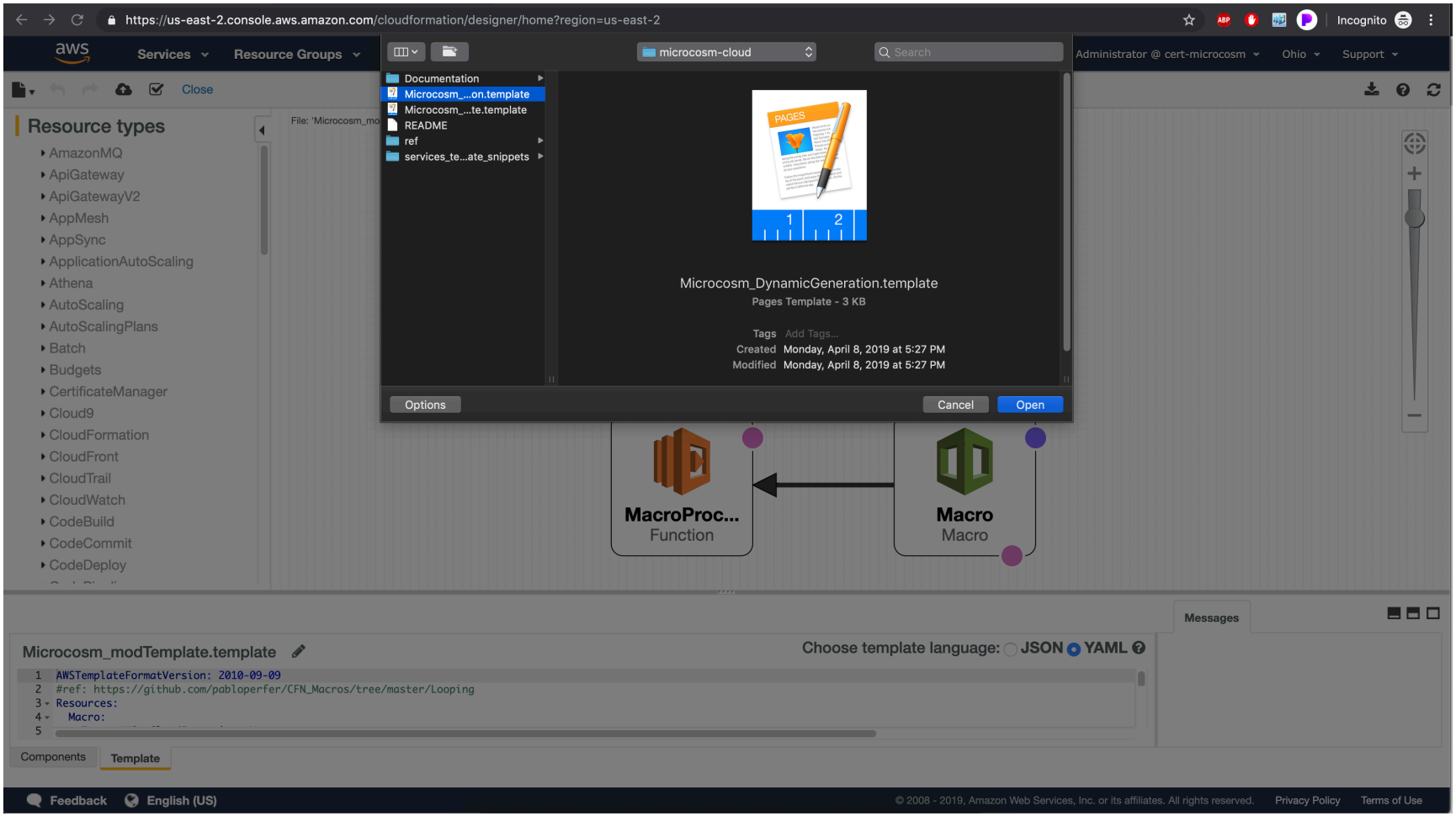
- To load an existing template, select open from the file icon in the left menu. (Note, the template pictured graphically here represents the Macro stack we'll be deploying later)



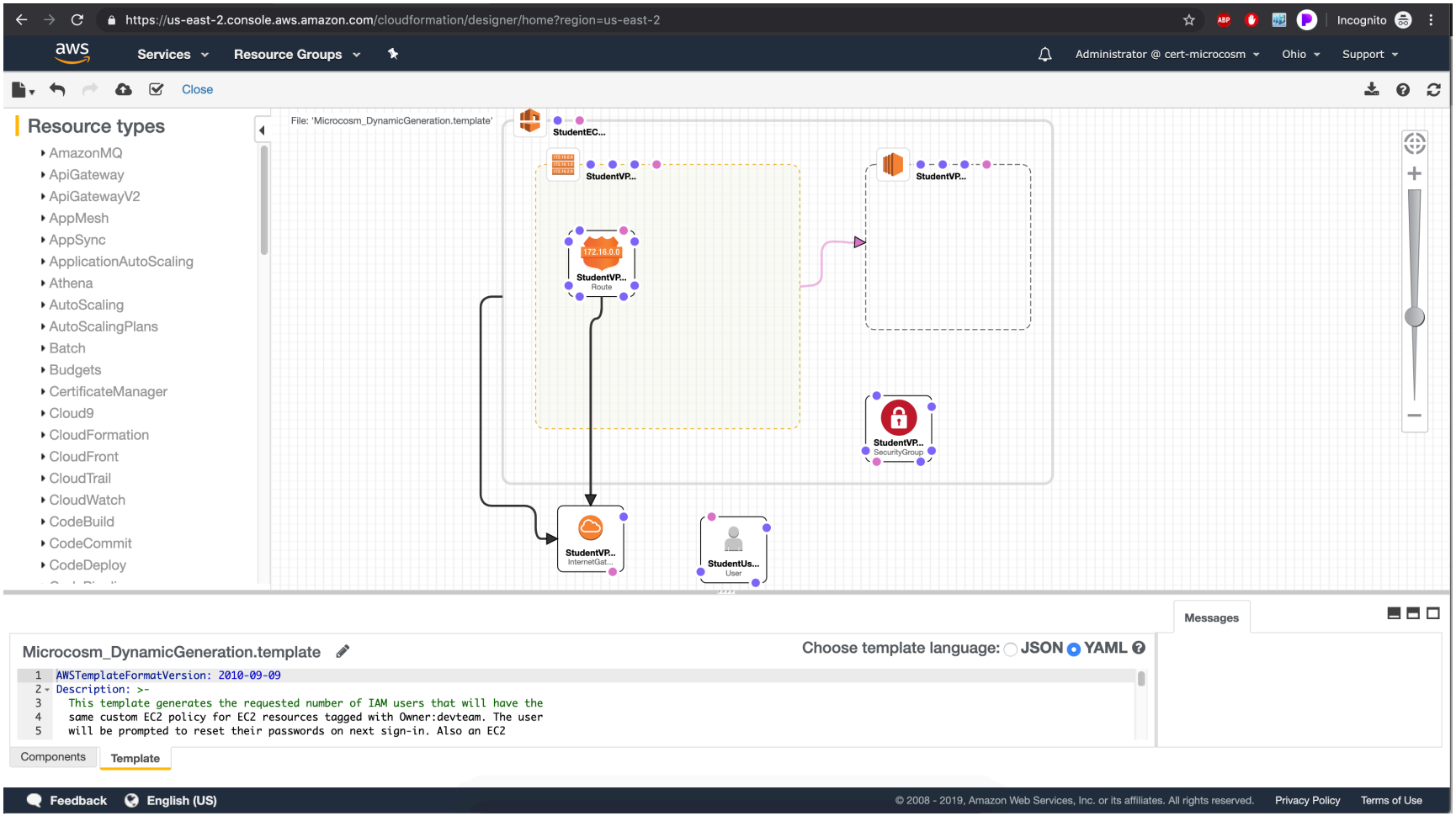
- Select the source of your template. It is good practice to name your templates with the extension `.template` or some variation thereof



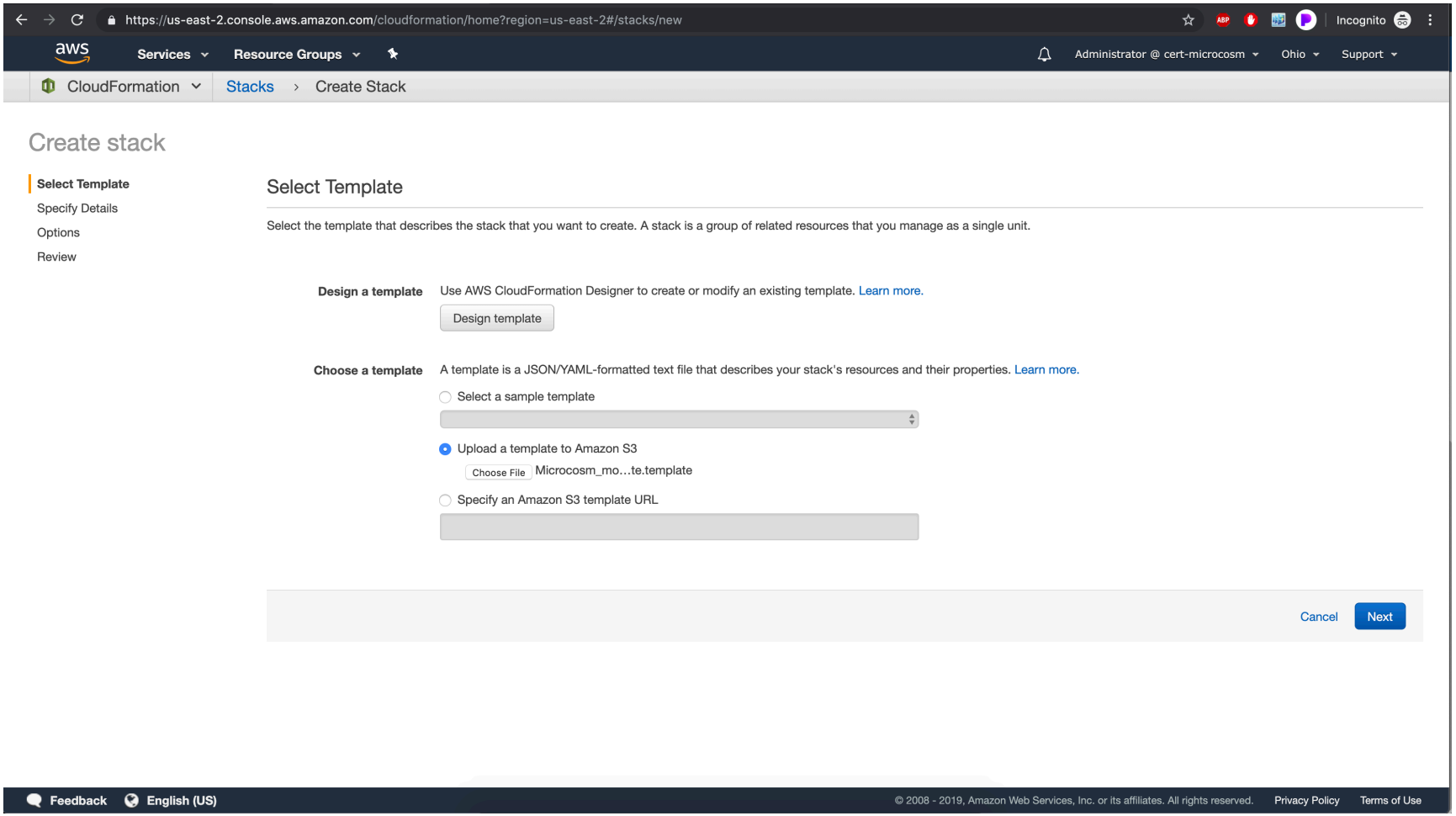
- Navigate to and select your template and press Open



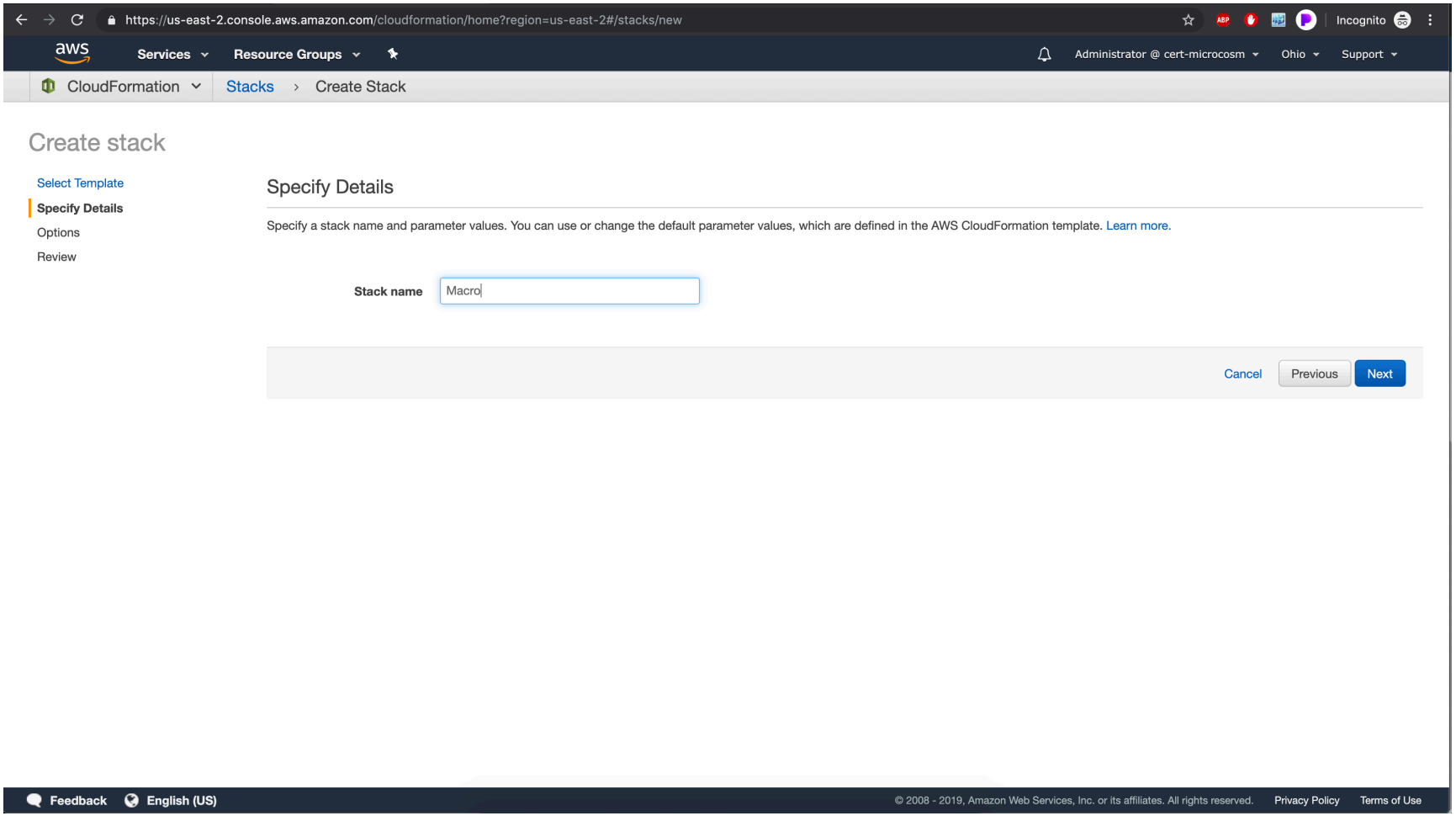
- With the template loaded, you are free to inspect, augment, change, or update its various elements. The template depicted here instantiates a VPC with the accompanying internet gateway, routes, security group, a user account, etc. This is our Dynamic template - as in when the template is loaded, the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



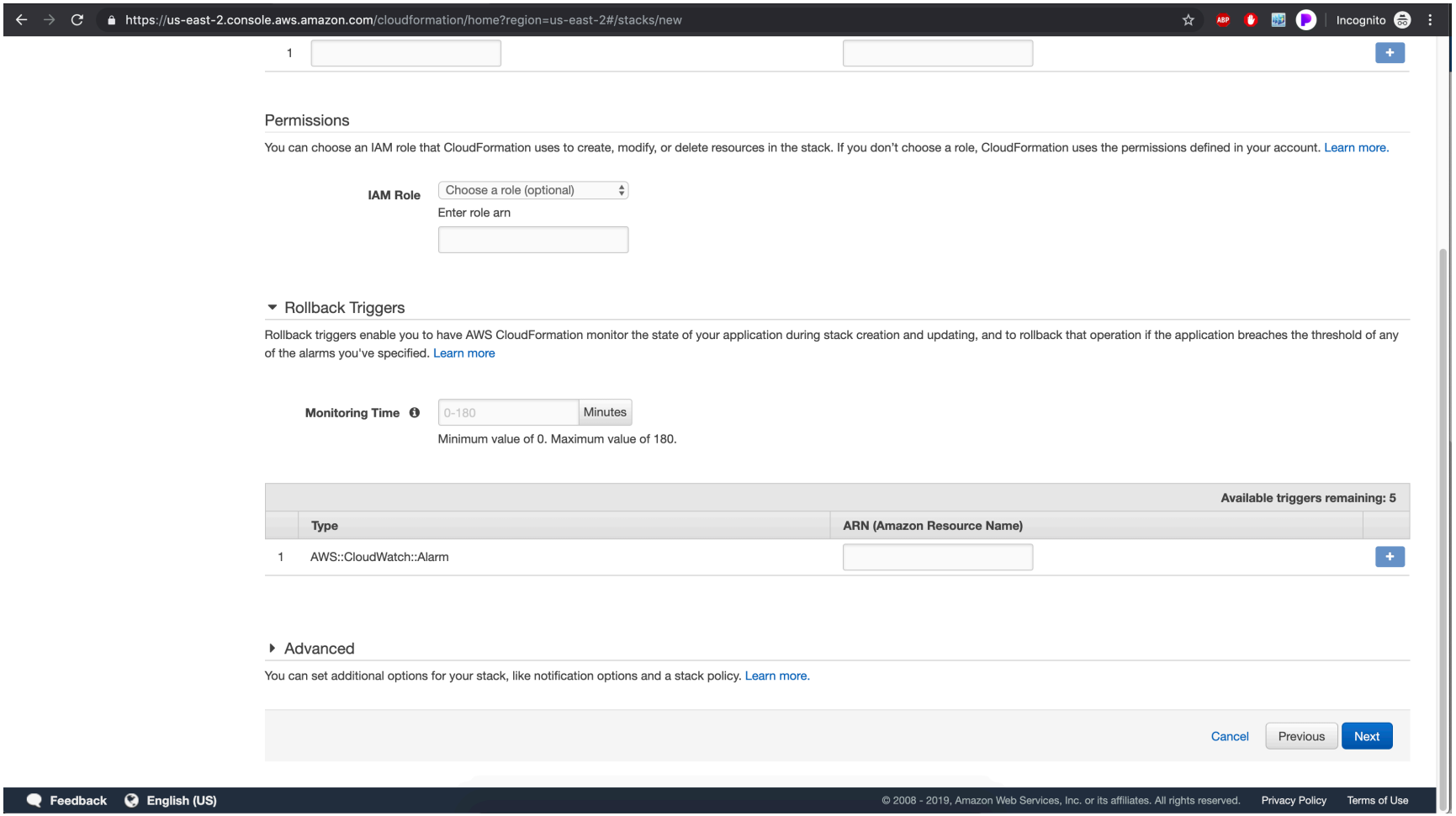
- Returning to the select template screen, select upload a template to Amazon S3
- Navigate to the Microcosm Macro template and select open
- Press Next



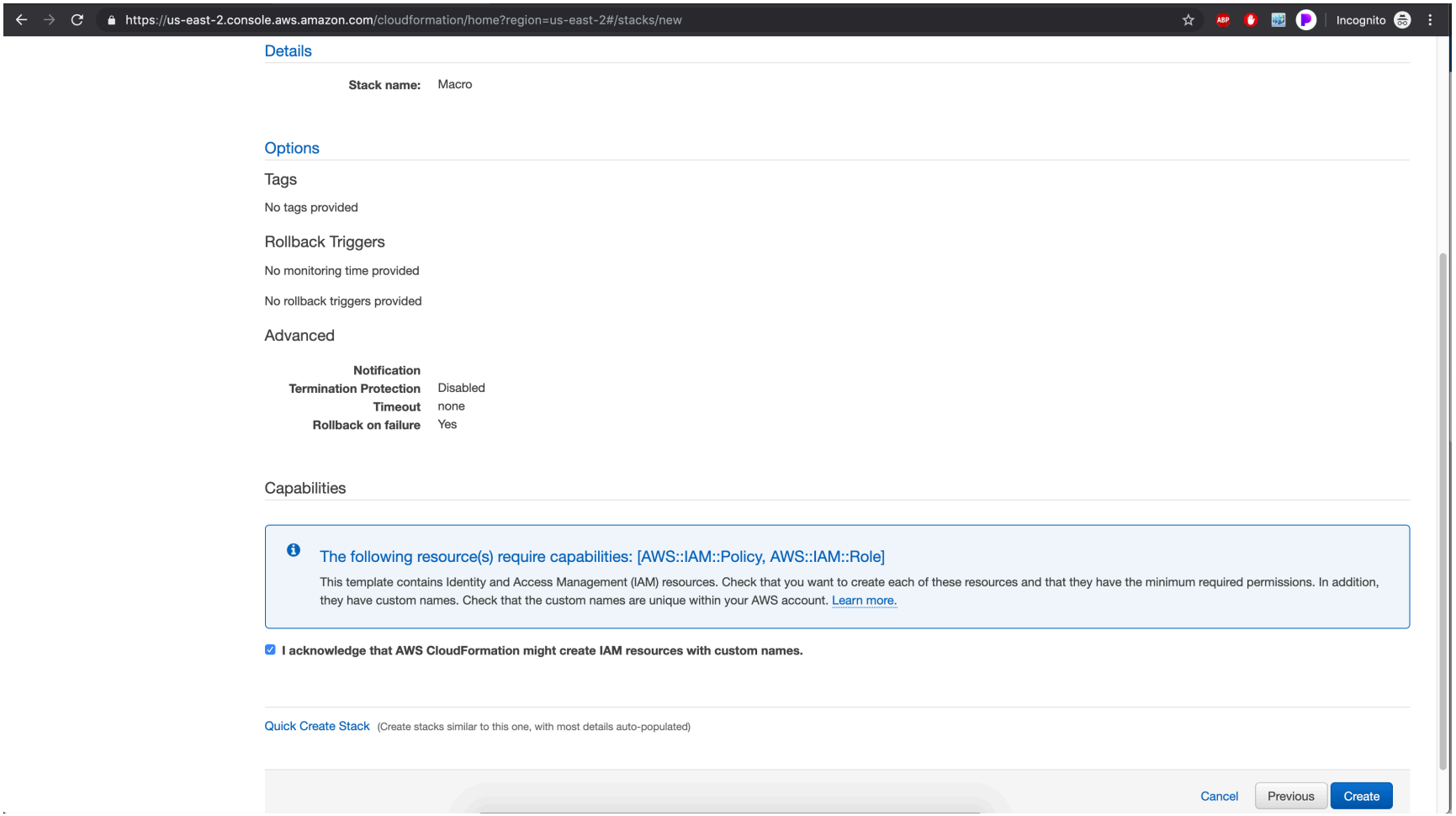
- **IMPORTANT** name the stack being created from this template `Macro`. The dynamic template being loaded next requires this naming for proper deployment and variable reference.
- Press Next


https://us-east-2.console.aws.amazon.com/cloudformation/home?region=us-east-2#/stacks/new
Incognito
aws
Services
Resource Groups
Administrator @ cert-microcosm
Ohio
Support
CloudFormation
Stacks
Create Stack
Create stack
Select Template
Specify Details
Options
Review
Specify Details
Specify a stack name and parameter values. You can use or change the default parameter values, which are defined in the AWS CloudFormation template. Learn more.
Stack name
Macro
Cancel
Previous
Next
Feedback
English (US)
© 2008 - 2019, Amazon Web Services, Inc. or its affiliates. All rights reserved.
Privacy Policy
Terms of Use

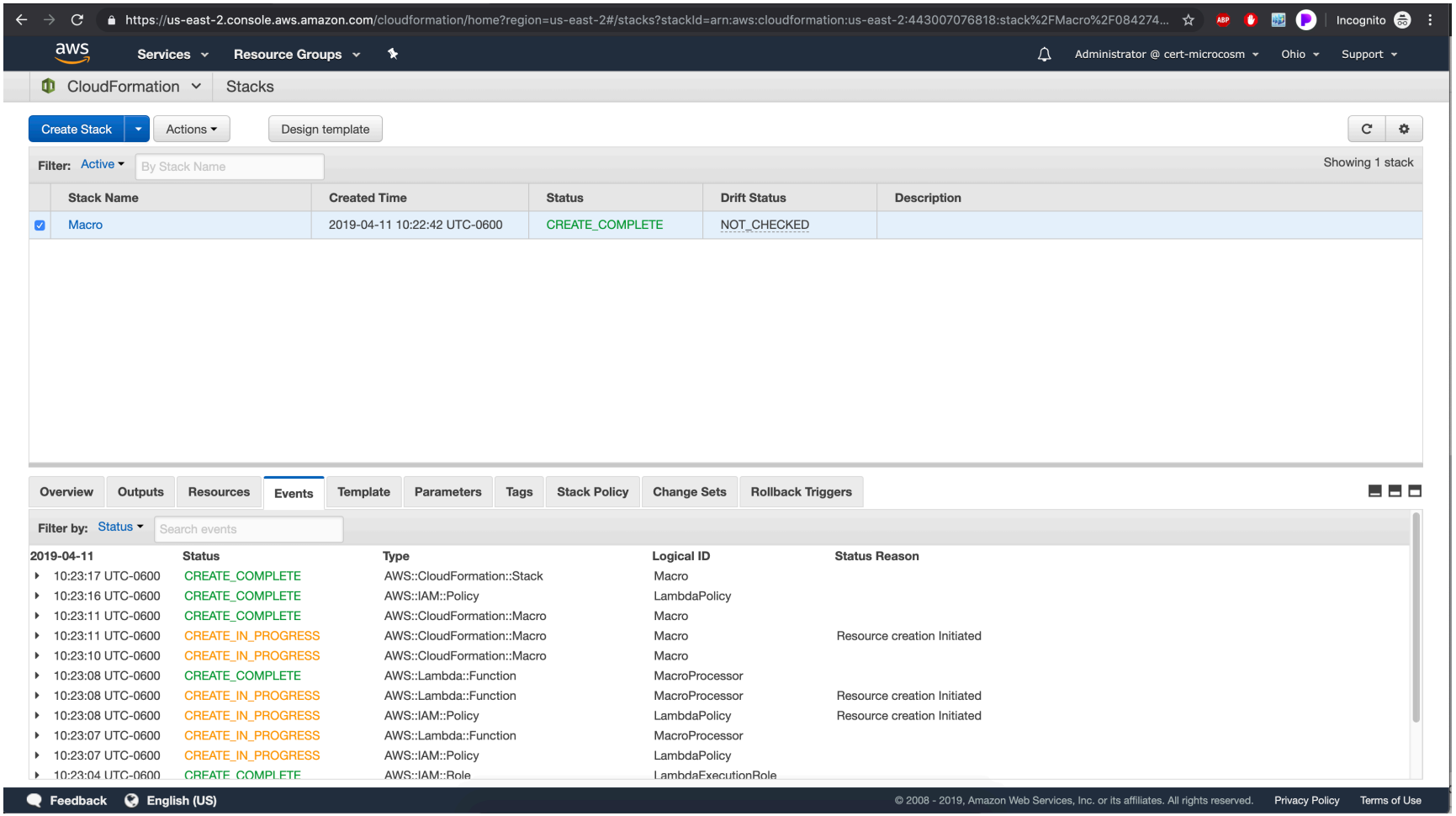
- Press Next



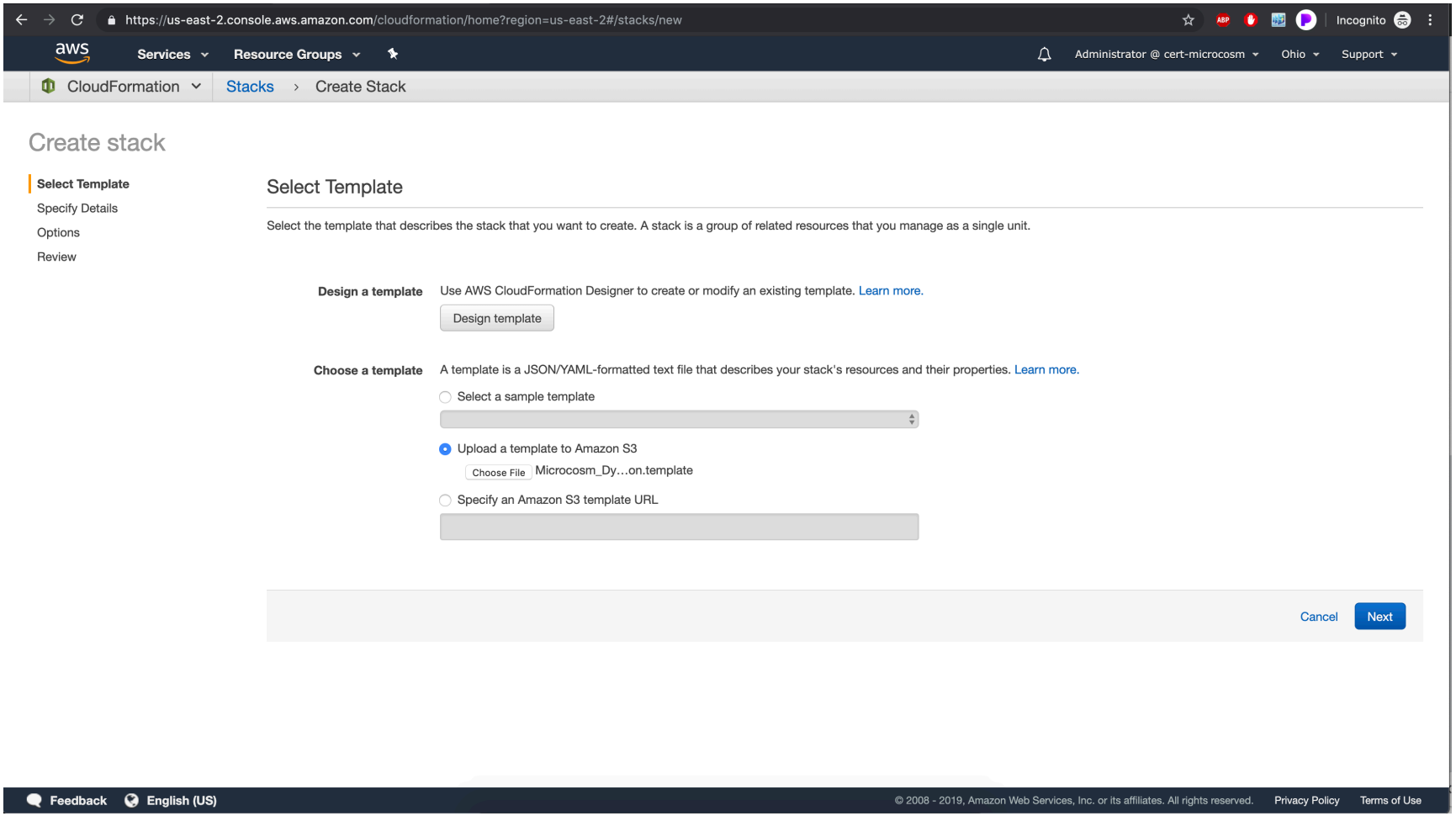
- Select the acknowledgement and press Create



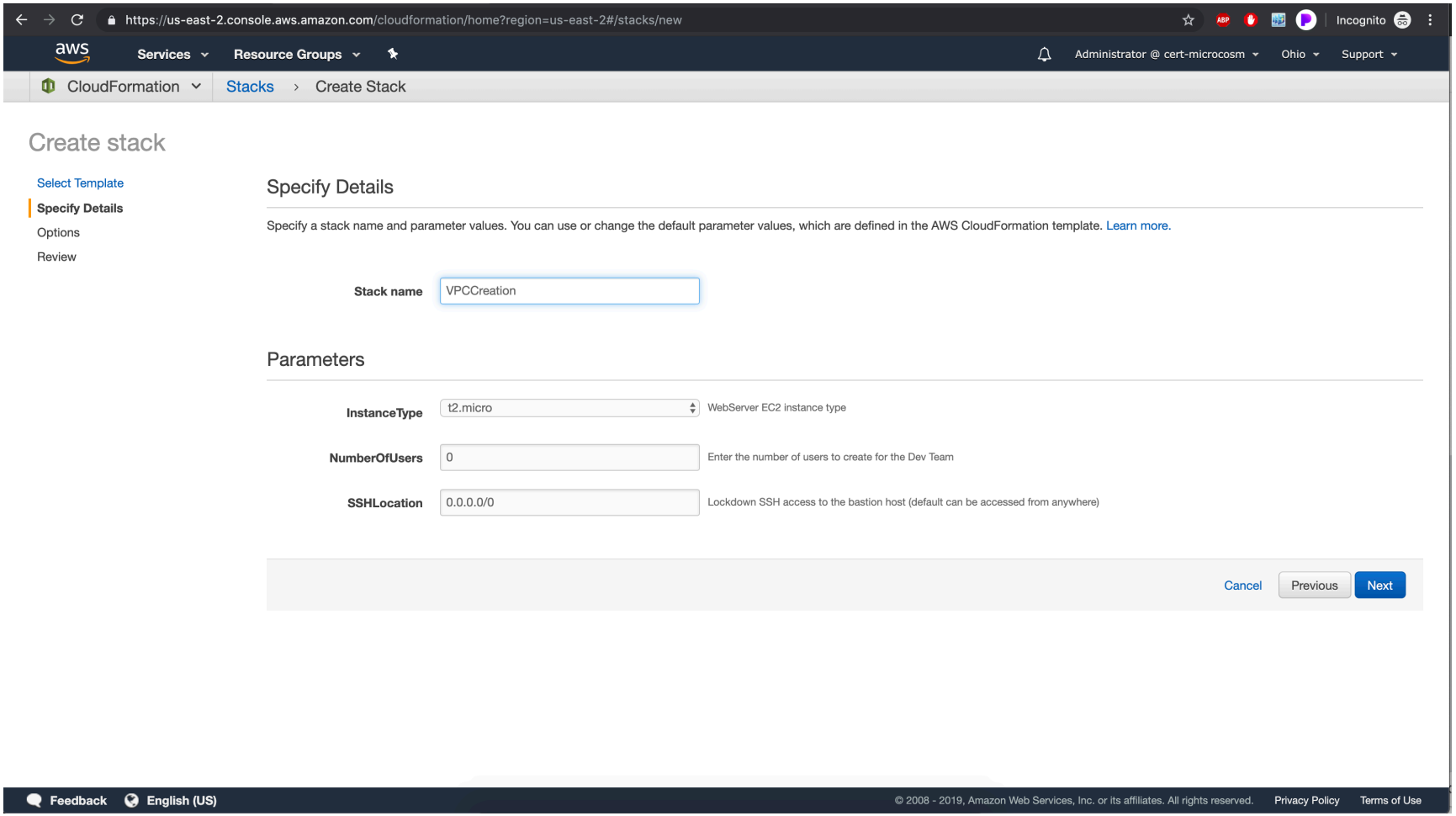
- On the following screen, you will see the Macro stack being created. While waiting for the creation to complete, explore the tabs giving details of the stacks and creation process



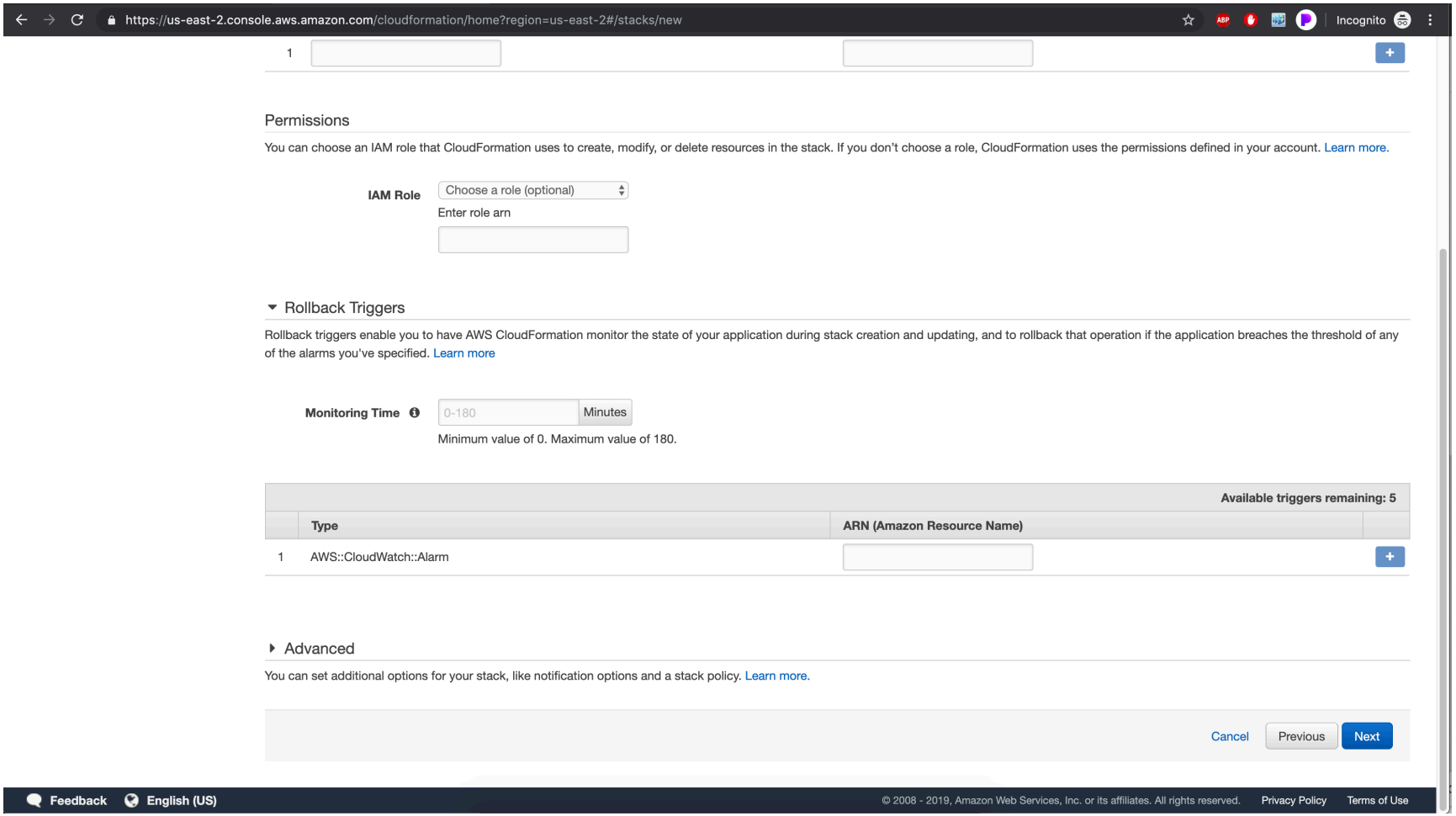
- Repeat the Stack creation process for the Microcosm Dynamic Template



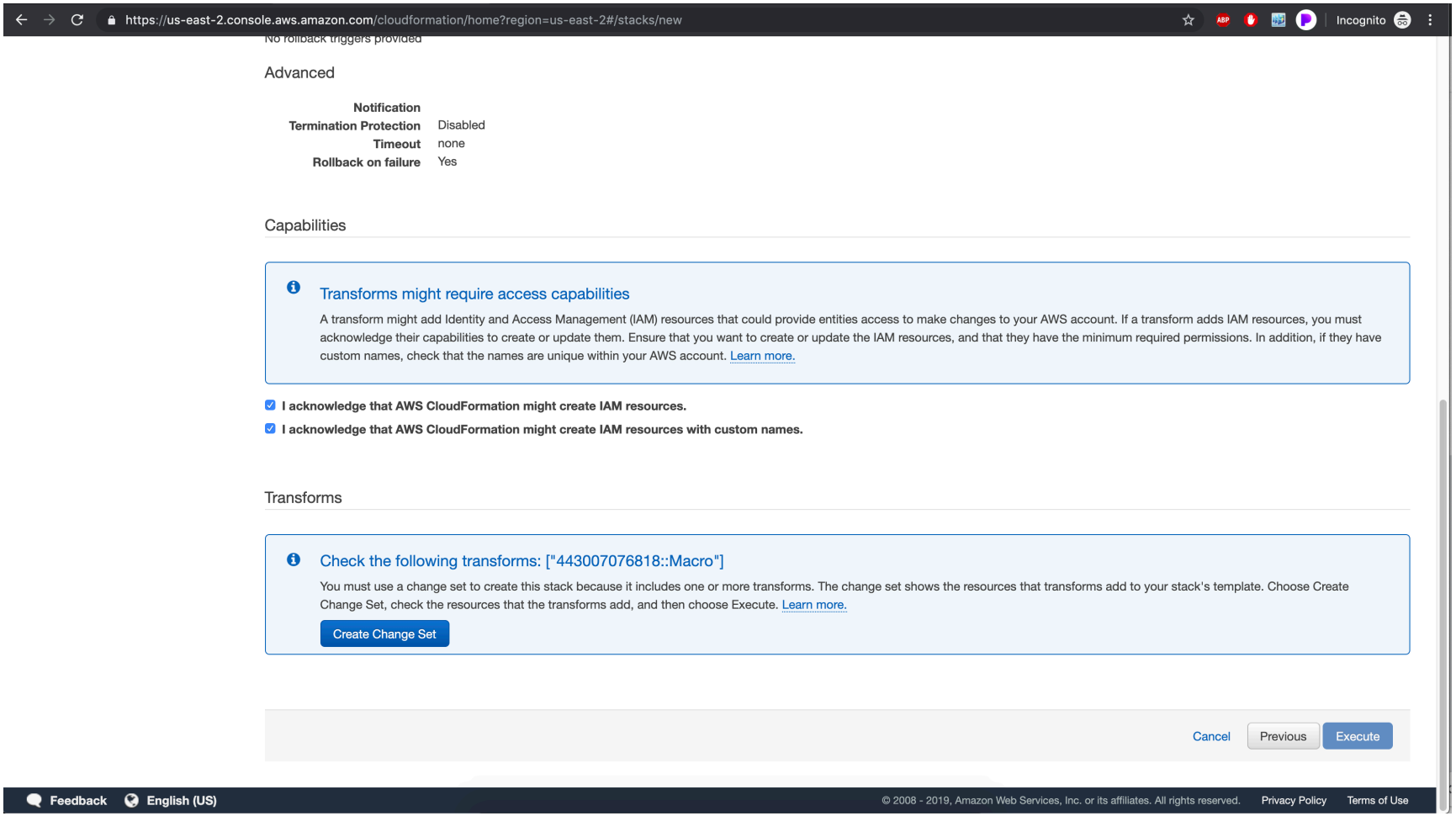
- Give the stack a name
- Enter the number of users to be created. If zero is set, 1 user and the associated VPC, etc are created. If 1.n.n are entered, 1...n *additional* users and VPCs are created (creating 2...n total users and VPCs). Remeber not to exceed the number of VPCs in your limit, nor the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


https://us-east-2.console.aws.amazon.com/cloudformation/home?region=us-east-2#/stacks/new
Incognito
aws
Services
Resource Groups
Administrator @ cert-microcosm
Ohio
Support
CloudFormation
Stacks
Create Stack
Create stack
Select Template
Specify Details
Options
Review
Specify Details
Specify a stack name and parameter values. You can use or change the default parameter values, which are defined in the AWS CloudFormation template. Learn more.
Stack name
VPCCreation
Parameters
InstanceType
t2.micro
WebServer EC2 instance type
NumberOfUsers
0
Enter the number of users to create for the Dev Team
SSHLocation
0.0.0.0/0
Lockdown SSH access to the bastion host (default can be accessed from anywhere)
Cancel
Previous
Next
Feedback
English (US)
© 2008 - 2019, Amazon Web Services, Inc. or its affiliates. All rights reserved.
Privacy Policy
Terms of Use

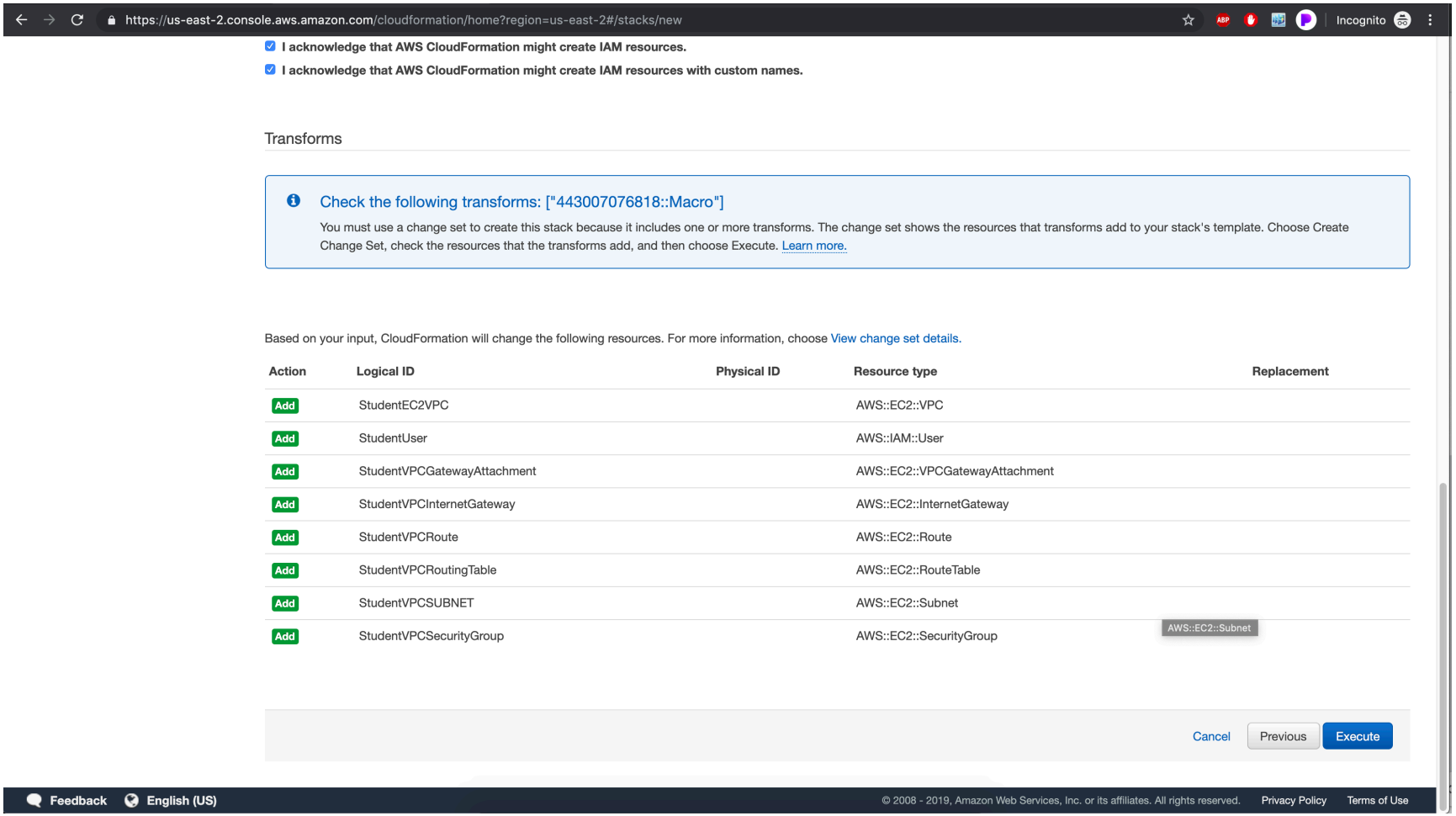
- Press Next



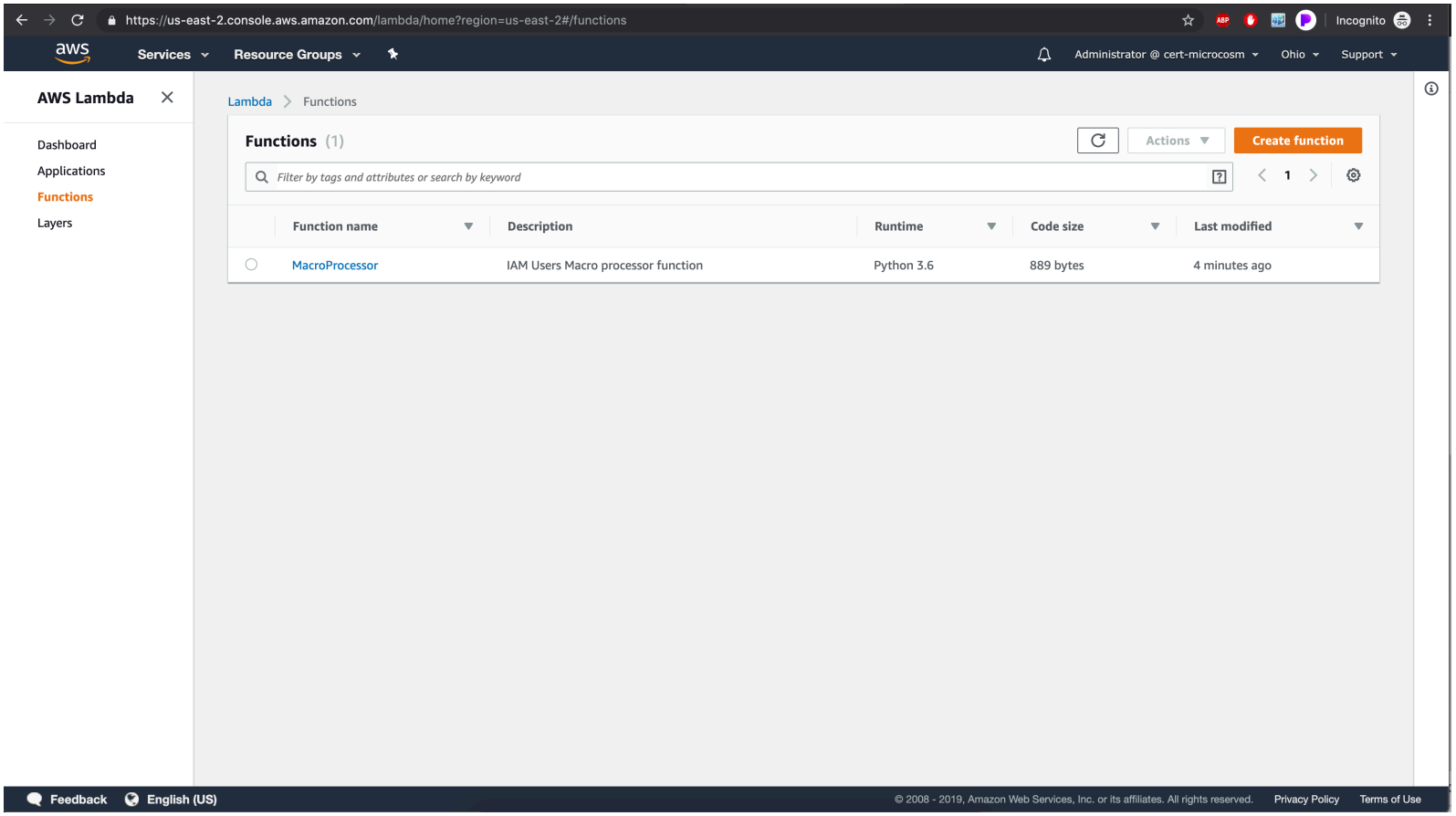
- Check the boxes acknowledging the behaviour of the template
- Select Create Change Set. This initiates the lamda function in the Macro stack deployed previously



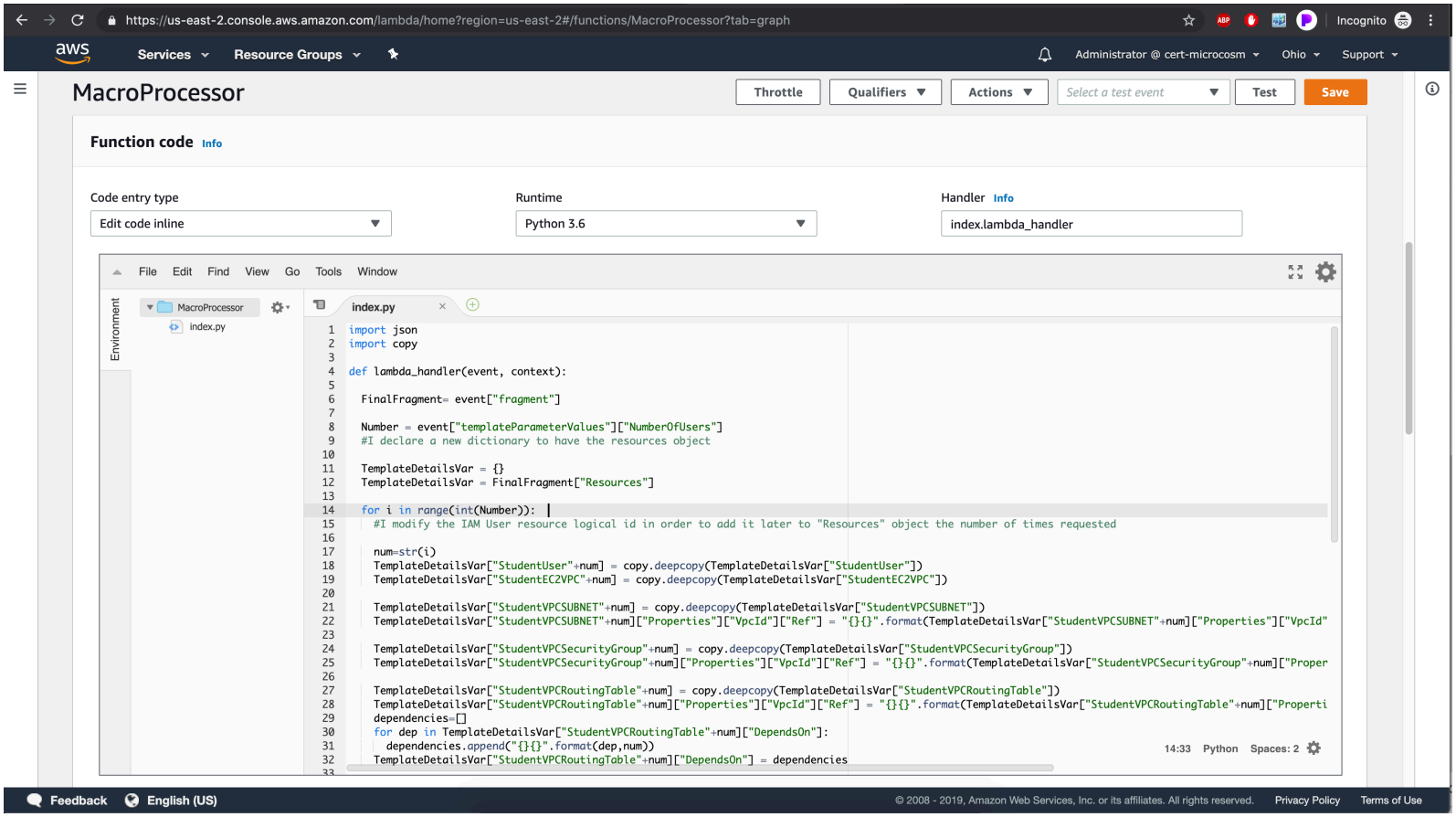
- Once the Change Set has been generated, select the Execute button



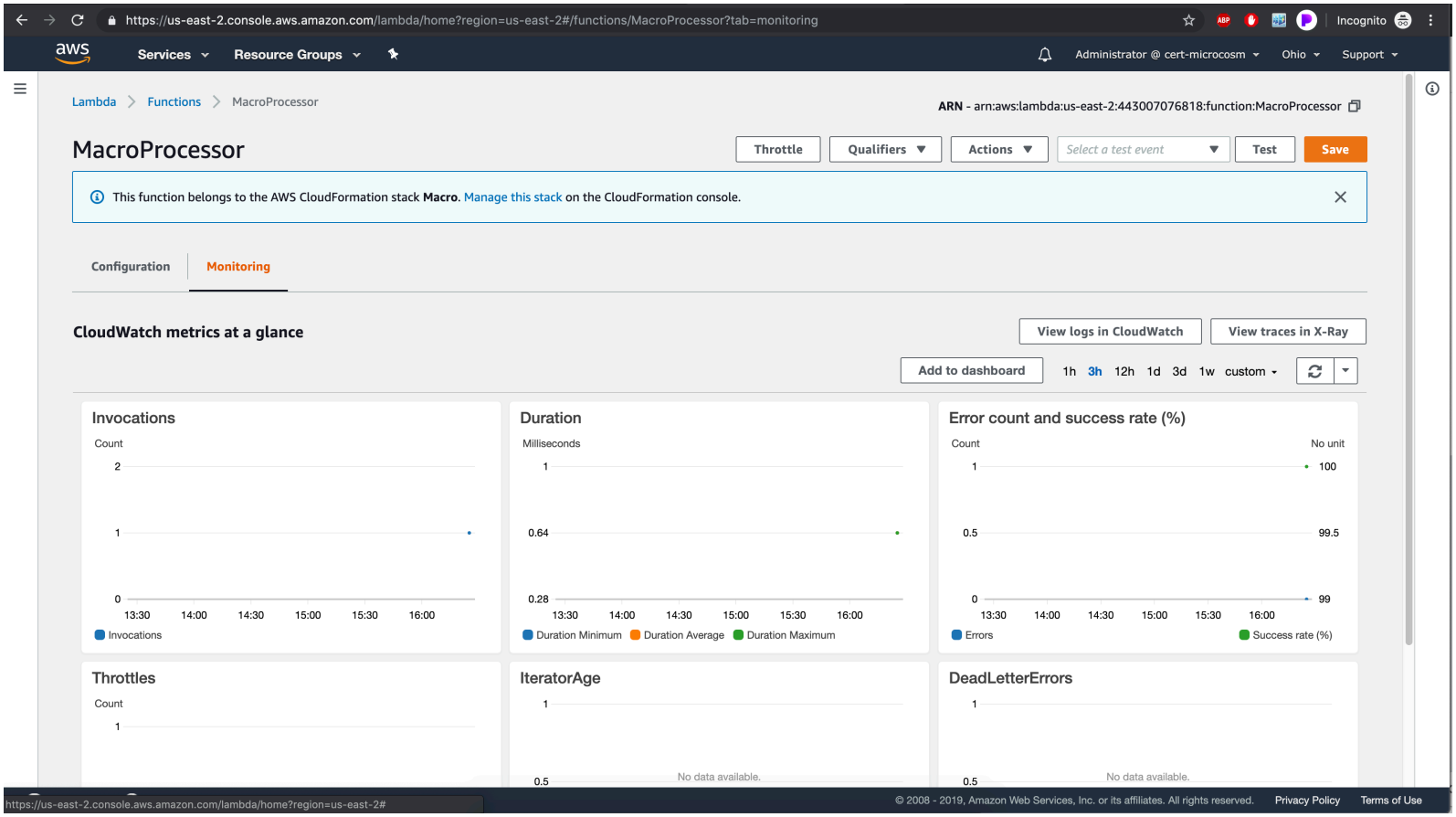
- When the creation is complete, observe the various elemts created by both of the stack creation processes
  - **AWS Lambda Function Deployed**
    - From the top services menu select Lambda
    - Select the MacroProcessor function



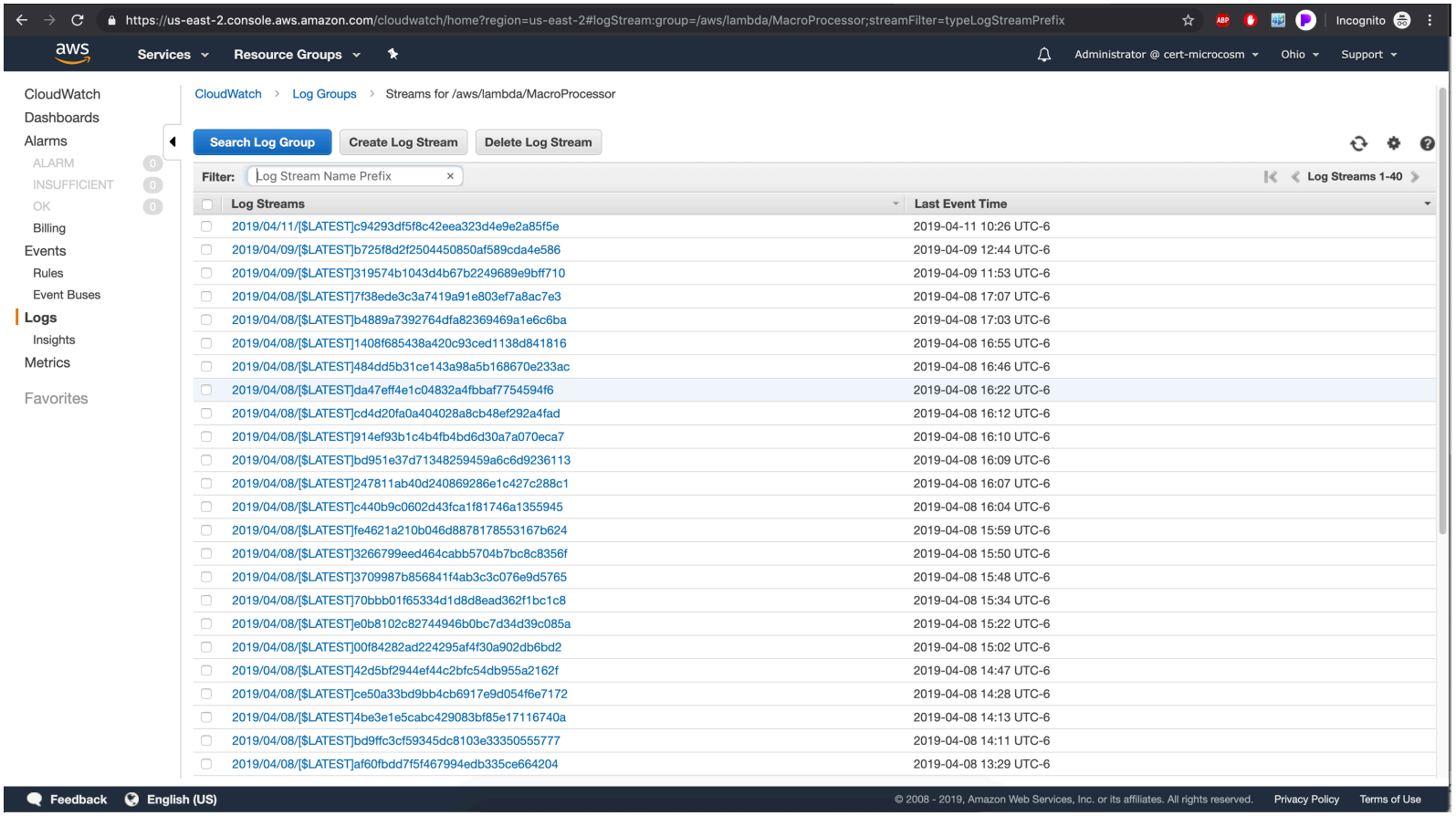
- Scroll down and observe the Lambda function code. This code is editable and when save is pressed and a change set is created for the Dynamic stack, the new version of the function will be run against the template (whether its the same template re-uploaded or the dynamic template which has been changed)



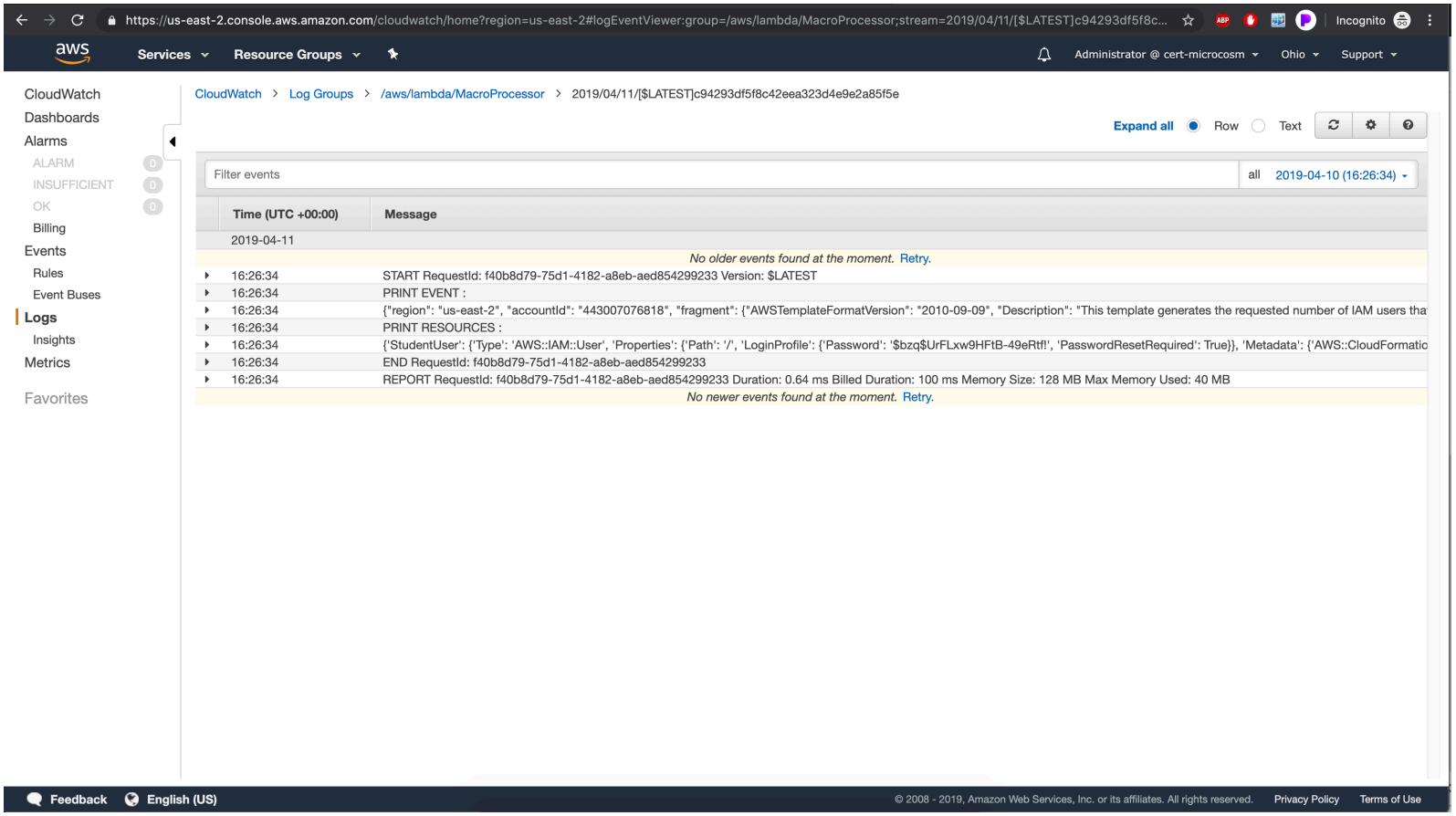
- Scrolling back up and selecting the Monitoring tab will allow you to see the lambda functions behaviours in graphical form, or you can select to view the CloudWatch logs



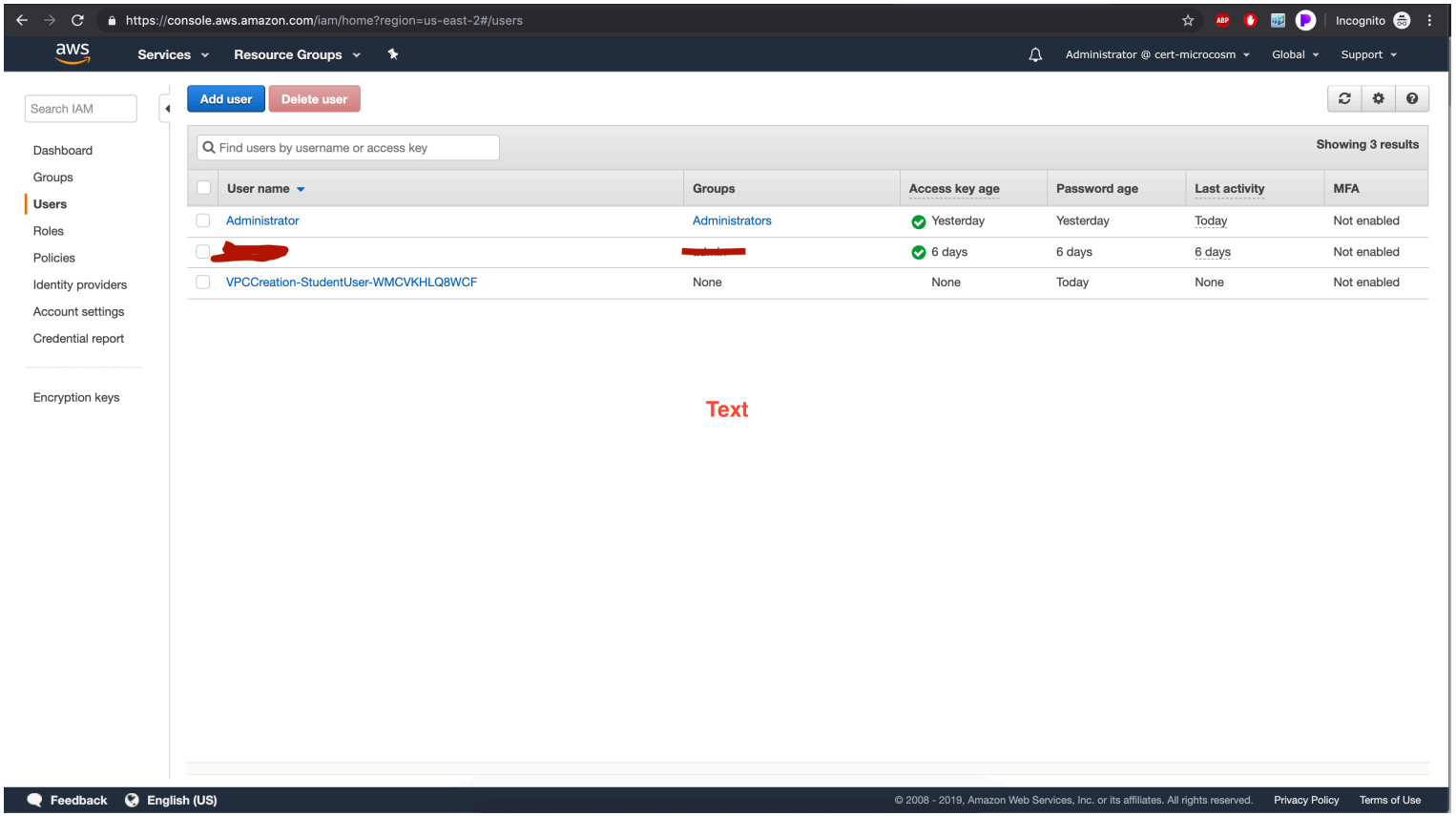
- After Selecting CloudWatch, select a LogStream representing the most recent code run



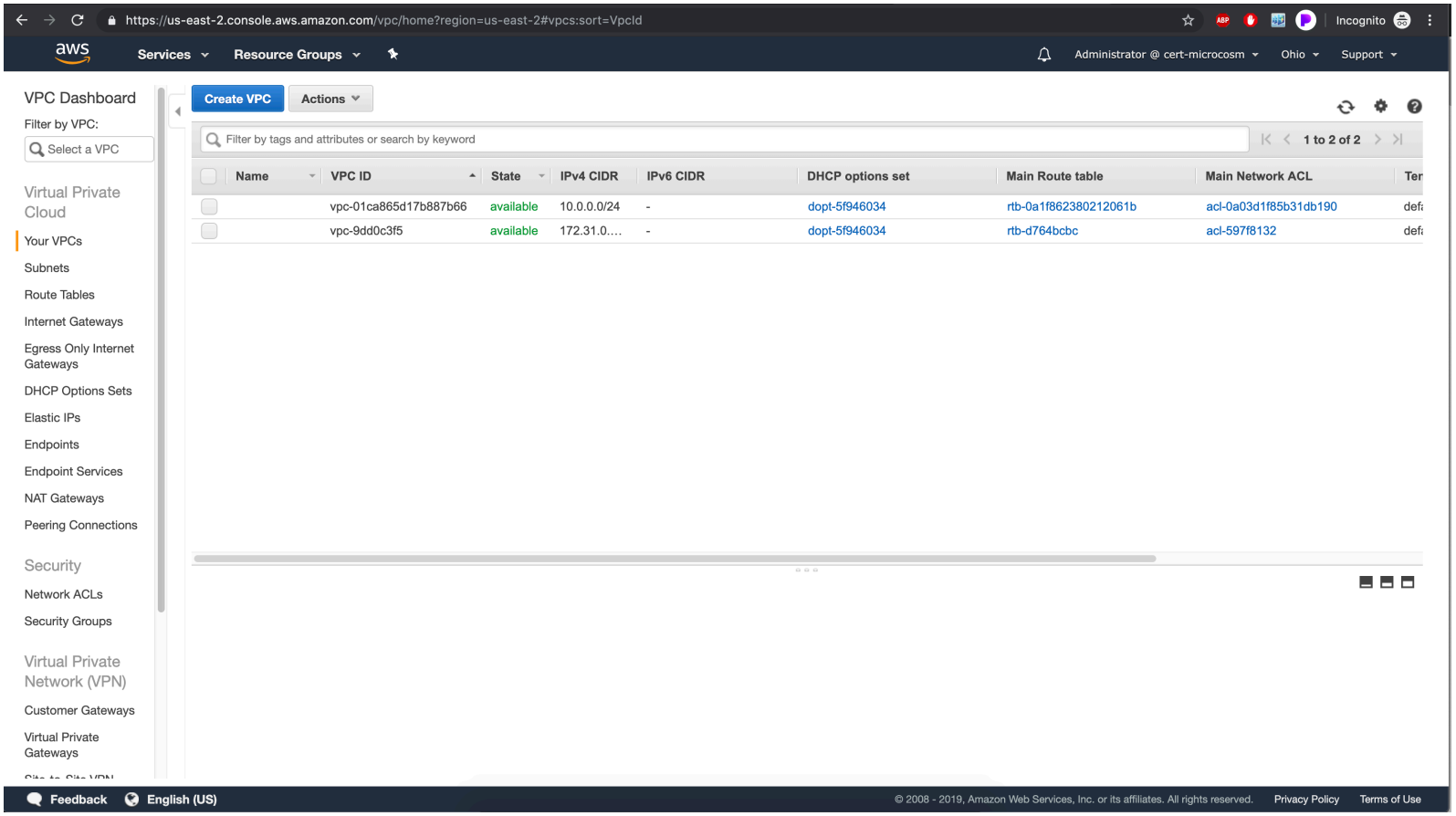
- View the logs generated by the lambda function



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



- **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



Return to Table of Contents

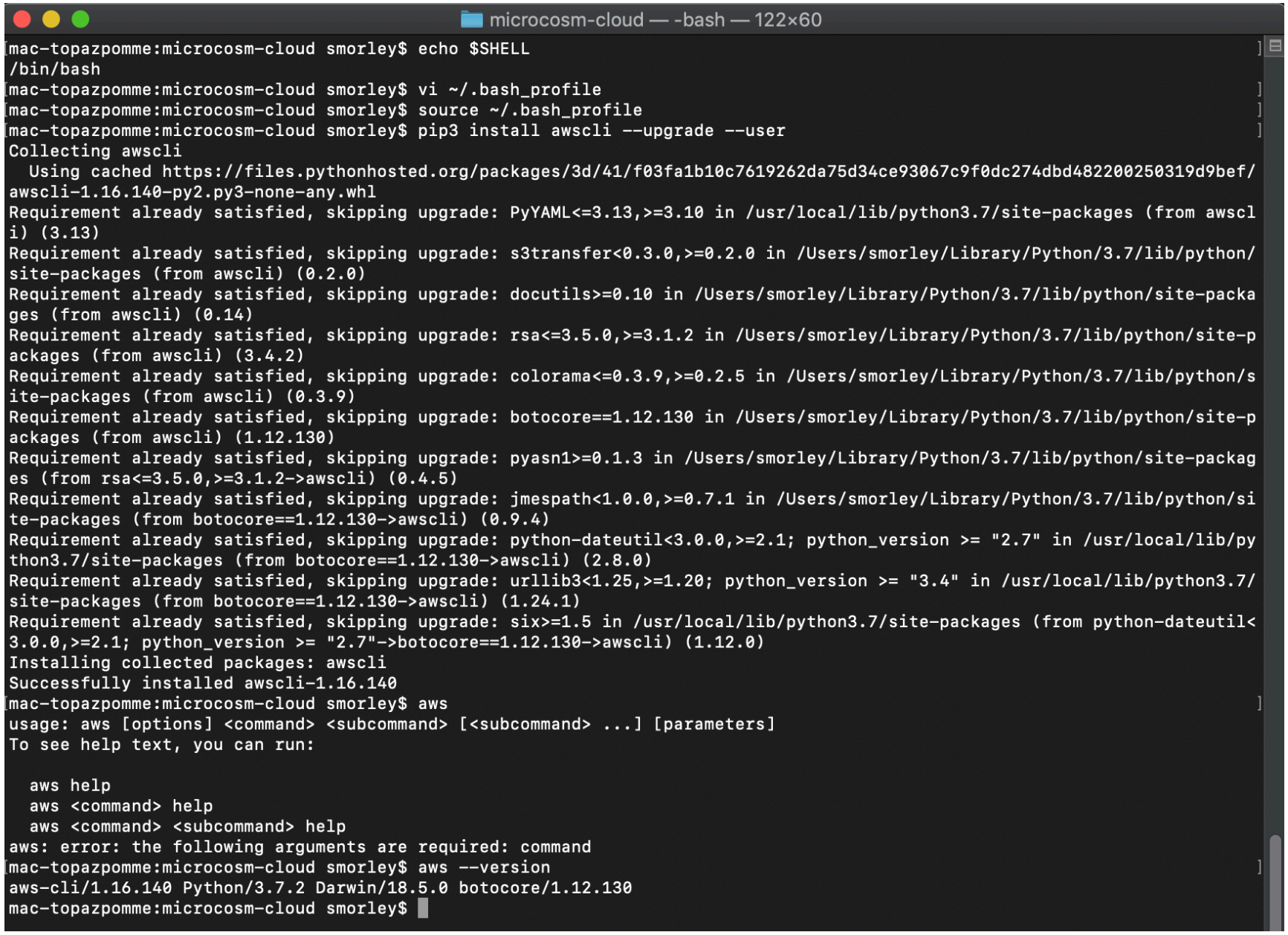
APPENDIX B

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

- If you haven't already, install python3 making use of these instructions.

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

- Install the AWS CLI



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

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

- Get Temporary AWS ECR Login credentials to use with Docker

```
mac-topazpomme:devops-microcosm smorley$ aws ecr get-login --region us-east-2  --no-include-email
docker login -u AWS -p eyJwYXlsb2FkIjoiKzhsTWtCL2krRjFoL3plVzFVcjBlajNvbWhPUGpMWW5OK01XNVJLUUpXR1pHTGFDQTMzQ2VBY3hYZ29XT0R
SS1YzYkpIdGdvZ0kvaWxHZzZ4Y2lEanJTQXFCaVZJemdXWVNScEx4Ymg3NDBvbW1JQTFHQS8wWWdmRnZJN1dBblFQa0pwS1Nza0hPZkR5N0VuSWtVYVBJWXorZ
1EzWDNBclcydnF5bjY0WDhaU014b3F2bkdoV09hKzM4bzlPNWdBVWtMWmVsMXhweWpBQncyNFBXVDdlcldKeG5SbGt2enVrSXhmQjFhZ20zek5sMkZhRTFydnh
ibEw3NnI2ZGQvNzNGeTU4c3FRa1g1SGlMdFlGY0ozTlRmMkJKckQ1cjlUUkR6TGlxZ2ZMRFl4eGlPMVVIdjAwaUovb2hEa1VtOFJMcDM3SGhabzJ1MnpjTkRVc
3gvbVNROW5EM2dvZ1NtL1YwUFhZcm1pdkkvNFJERERxYnA5eXZmcHVSazV2OUF6MGZucld3NnJUSS9IZHJCZzFxeVVnc3hnSmo0RXY5TUtaOWZzb0pvRFlXSlh
6N2ZndWpIRXM0Y3AyM2dUSEhodUJrN09FVkllNVgvV3NIMEFMaHBhYWI2UXMzbUJVa1RvdlpMd1M0Uk5zL0lWMThXb0hRRk4vUU1iSW9qaWxNRGxnMHFCQkorV
GxaYTNYOFVrckZMNE5BU0lmdFNSL0NNc2hTSUd3aVlKemZIZWcyVDZZVWZiekowT3VtNDRPWTY3RDRzb3JaMTRuRWZ3bkJ1dUo5SUNPVEFyK0pJTms0aEVDUnF
JQ2x3SFU1S0p5REZOYXlyRFEzeGQ5cU5GUEJxR3l4bURxeFYwd0o0MkorUy8vSE1DTTc3QWErL0UzNFBNaEt1N2E5a3k5YzVPYXQzUGNxeUFEUkJaaFYzYzQyd
TRkQ1VQUWtDZUUyUDV4WEFLUnJNOUZ4RFJwT3l5by9IUXVHNG9iWEQxZ0htUVhnRXc3OUh1WnN5aEVXdnp3bnVHaC9tajVUUUlSVjNRL1YzbW9LMGZ0VW9HWm9
FWmIxRHRjUjBHSU9SUTZoZGMzWGxrZS9zNU9uRFllSzJvSkk3bE1yVzdCM09jMXdoNG1oUHpVNXNpYUhjK0loNFpOb252ZlVrMTN4OGxMRVRweHFLN0I1Q1g5N
zdDeERvTjdnbksvZS9DdlJHVVhHS0NIOUsxbW1LblVSL1R5VlZzeklMVHU3OTk5MXBGaU5CK2pmeVVQU0o4ZXVmNkQwaHc2cmw5bkhkbWlBYUZhNHFuUW55bUQ
xbmcxUWVwa0liUUh6aURZZm5tNFZGOWNmaVZSY21DWGJXM0FMWWpaZXBJUFJWM0FKcW9VUElkd3NTd2NycHNsZHBCU0x5YkJ1UVdvRmZxanRvT2YzWGpObzVke
G8zU3RZbmxoSkVpMU9yUGkxTVh2bUFLciszMkd4anlCMWY3K29DMzRiL3orVkUwVUdFNmJTR2ZFR0pTVkxMckJ3PT0iLCJkYXRha2V5IjoiQVFFQkFIakI3L2l
nd01nNE5Qd2F1cnhTSVl4NEhmbnh1R2MvNDhiRHd2d0RwTllXWmdBQUFINHdmQVlKS29aSWh2Y05BUWNHb0c4d2JRSUJBREJvQmdrcWhraUc5dzBCQndFd0hnW
UpZSVpJQVdVREJBRXVNQkVFRExRYkViQzNwY0htR3c2bk5BSUJFSUE3eUUveWI1MkpOOXl5MnMzeTRUd3JiNm9nRGV0dlArNCswOVIwcm1kM0dPL1lvai96TU9
BNEpiNVprUHM4QWFqWnRNM09vL1IxY1FLNmJsdz0iLCJ2ZXJzaW9uIjoiMiIsInR5cGUiOiJEQVRBX0tFWSIsImV4cGlyYXRpb24iOjE1NTQ5NjQ4ODF9 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 eyJwYXlsb2FkIjoiKzhsTWtCL2krRjFoL3plVzFVcjBlajNvbWhPUGpMWW
5OK01XNVJLUUpXR1pHTGFDQTMzQ2VBY3hYZ29XT0RSS1YzYkpIdGdvZ0kvaWxHZzZ4Y2lEanJTQXFCaVZJemdXWVNScEx4Ymg3NDBvbW1JQTFHQS8wWWdmRnZJ
N1dBblFQa0pwS1Nza0hPZkR5N0VuSWtVYVBJWXorZ1EzWDNBclcydnF5bjY0WDhaU014b3F2bkdoV09hKzM4bzlPNWdBVWtMWmVsMXhweWpBQncyNFBXVDdlcl
dKeG5SbGt2enVrSXhmQjFhZ20zek5sMkZhRTFydnhibEw3NnI2ZGQvNzNGeTU4c3FRa1g1SGlMdFlGY0ozTlRmMkJKckQ1cjlUUkR6TGlxZ2ZMRFl4eGlPMVVI
djAwaUovb2hEa1VtOFJMcDM3SGhabzJ1MnpjTkRVc3gvbVNROW5EM2dvZ1NtL1YwUFhZcm1pdkkvNFJERERxYnA5eXZmcHVSazV2OUF6MGZucld3NnJUSS9IZH
JCZzFxeVVnc3hnSmo0RXY5TUtaOWZzb0pvRFlXSlh6N2ZndWpIRXM0Y3AyM2dUSEhodUJrN09FVkllNVgvV3NIMEFMaHBhYWI2UXMzbUJVa1RvdlpMd1M0Uk5z
L0lWMThXb0hRRk4vUU1iSW9qaWxNRGxnMHFCQkorVGxaYTNYOFVrckZMNE5BU0lmdFNSL0NNc2hTSUd3aVlKemZIZWcyVDZZVWZiekowT3VtNDRPWTY3RDRzb3
JaMTRuRWZ3bkJ1dUo5SUNPVEFyK0pJTms0aEVDUnFJQ2x3SFU1S0p5REZOYXlyRFEzeGQ5cU5GUEJxR3l4bURxeFYwd0o0MkorUy8vSE1DTTc3QWErL0UzNFBN
aEt1N2E5a3k5YzVPYXQzUGNxeUFEUkJaaFYzYzQydTRkQ1VQUWtDZUUyUDV4WEFLUnJNOUZ4RFJwT3l5by9IUXVHNG9iWEQxZ0htUVhnRXc3OUh1WnN5aEVXdn
p3bnVHaC9tajVUUUlSVjNRL1YzbW9LMGZ0VW9HWm9FWmIxRHRjUjBHSU9SUTZoZGMzWGxrZS9zNU9uRFllSzJvSkk3bE1yVzdCM09jMXdoNG1oUHpVNXNpYUhj
K0loNFpOb252ZlVrMTN4OGxMRVRweHFLN0I1Q1g5NzdDeERvTjdnbksvZS9DdlJHVVhHS0NIOUsxbW1LblVSL1R5VlZzeklMVHU3OTk5MXBGaU5CK2pmeVVQU0
o4ZXVmNkQwaHc2cmw5bkhkbWlBYUZhNHFuUW55bUQxbmcxUWVwa0liUUh6aURZZm5tNFZGOWNmaVZSY21DWGJXM0FMWWpaZXBJUFJWM0FKcW9VUElkd3NTd2Ny
cHNsZHBCU0x5YkJ1UVdvRmZxanRvT2YzWGpObzVkeG8zU3RZbmxoSkVpMU9yUGkxTVh2bUFLciszMkd4anlCMWY3K29DMzRiL3orVkUwVUdFNmJTR2ZFR0pTVk
xMckJ3PT0iLCJkYXRha2V5IjoiQVFFQkFIakI3L2lnd01nNE5Qd2F1cnhTSVl4NEhmbnh1R2MvNDhiRHd2d0RwTllXWmdBQUFINHdmQVlKS29aSWh2Y05BUWNH
b0c4d2JRSUJBREJvQmdrcWhraUc5dzBCQndFd0hnWUpZSVpJQVdVREJBRXVNQkVFRExRYkViQzNwY0htR3c2bk5BSUJFSUE3eUUveWI1MkpOOXl5MnMzeTRUd3
JiNm9nRGV0dlArNCswOVIwcm1kM0dPL1lvai96TU9BNEpiNVprUHM4QWFqWnRNM09vL1IxY1FLNmJsdz0iLCJ2ZXJzaW9uIjoiMiIsInR5cGUiOiJEQVRBX0tF
WSIsImV4cGlyYXRpb24iOjE1NTQ5NjQ4ODF9 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               NA
MES
[mac-topazpomme:devops-microcosm smorley$ docker container ls --all                                                              ]
CONTAINER ID        IMAGE                         COMMAND                  CREATED             STATUS
  PORTS               NAMES
b57f36821ce5        sonarqube:lts                 "./bin/run.sh"           7 days ago          Exited (0) 42 minutes ago
                      sonarqube
73a4061cd9f0        gillax/hubot-slack-jenkins    "./bin/hubot -a slack"   7 days ago          Exited (0) 37 minutes ago
                      hubot
ba06c7de2e42        gitlab/gitlab-ce              "/assets/wrapper"        7 days ago          Exited (0) 37 minutes ago
                      gitlab
3193c8f8470b        sonatype/nexus                "/bin/sh -c 'java   …"   7 days ago          Exited (143) 37 minutes ago
                      sonatype_nexus
013b8b77e566        h1kkan/jenkins-docker:lts     "/sbin/tini -- /usr/…"   7 days ago          Exited (143) 37 minutes ago
                      jenkins
3fda643668ec        owasp/zap2docker-stable       "zap-webswing.sh"        7 days ago          Exited (0) 44 minutes ago
                      owaspzap
[mac-topazpomme:devops-microcosm smorley$ docker tag sonarqube:lts sonarqube:lts0419                                             ]
```

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

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

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

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

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

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

- Verify tag is correct by relisting images

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

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

```
mac-topazpomme:devops-microcosm smorley$ aws ecr create-repository --repository-name sonar
qube
{
    "repository": {
        "repositoryArn": "arn:aws:ecr:us-east-2:443007076818:repository/sonarqube",
        "registryId": "443007076818",
        "repositoryName": "sonarqube",
        "repositoryUri": "443007076818.dkr.ecr.us-east-2.amazonaws.com/sonarqube",
        "createdAt": 1554929452.0
    }
}
mac-topazpomme:devops-microcosm smorley$ docker tag sonarqube:lts 443007076818.dkr.ecr.us-
east-2.amazonaws.com/sonarqube:lts
mac-topazpomme:devops-microcosm smorley$ docker push 443007076818.dkr.ecr.us-east-2.amazon
aws.com/sonarqube:lts
The push refers to repository [443007076818.dkr.ecr.us-east-2.amazonaws.com/sonarqube]
68e5087b6ffc: Pushed
237afe62393f: Pushed
079758d1bbb7: Pushed
e11f2ab9e2f4: Pushed
f7d12d471667: Pushed
f350d0146bb3: Pushed
e38df31d449c: Pushed
af5ae4841776: Pushed
b17cc31e431b: Pushed
12cb127eee44: Pushed
604829a174eb: Pushed
fbb641a8b943: Pushed
lts: digest: sha256:8cbd208b264ab1404bce2ab16bcd6ba9a31a35e5b3e800c0d72295d719c087e8 size:
 2839
```

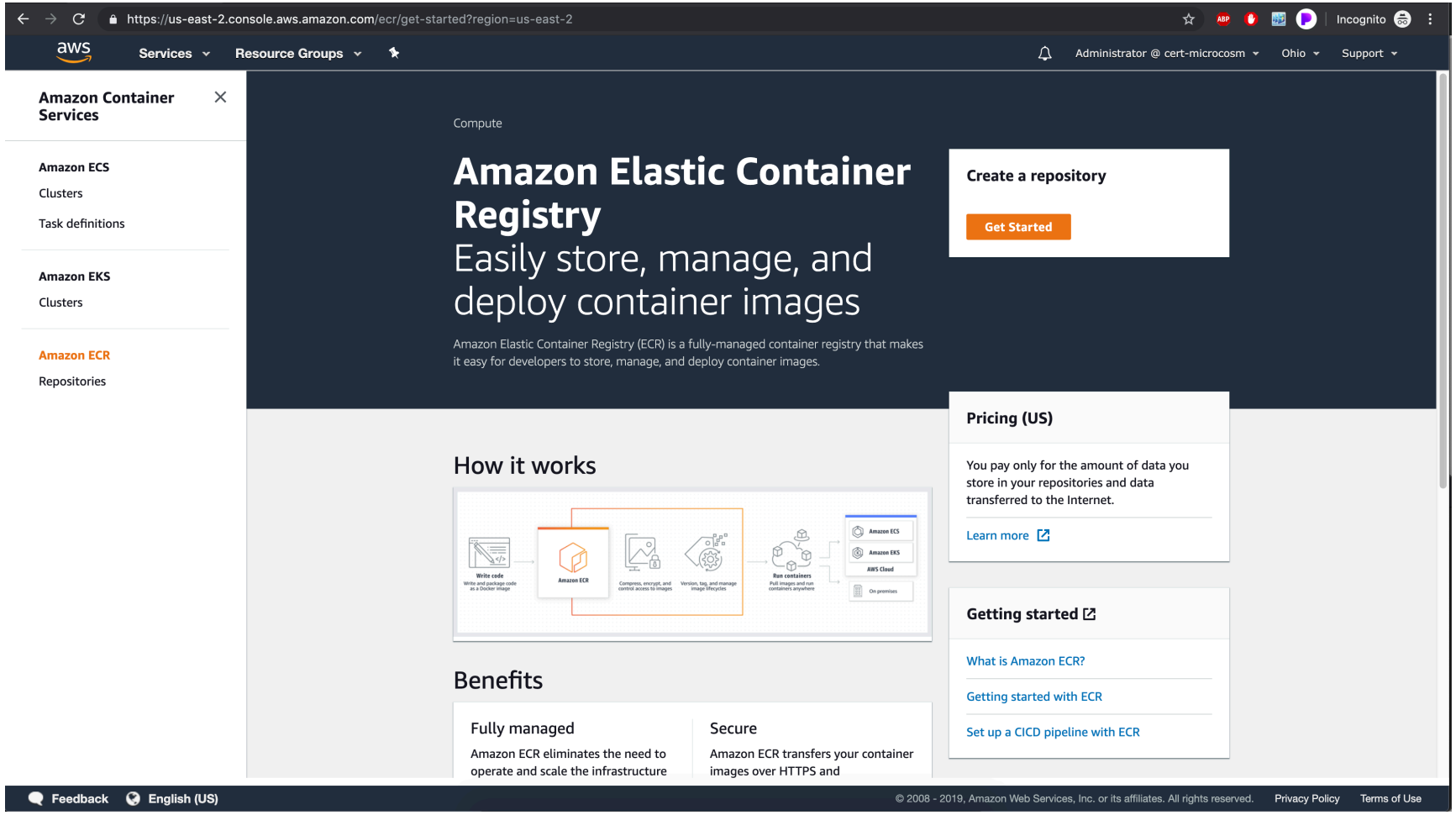
- Repeat for as many images as desired to add to the container registry

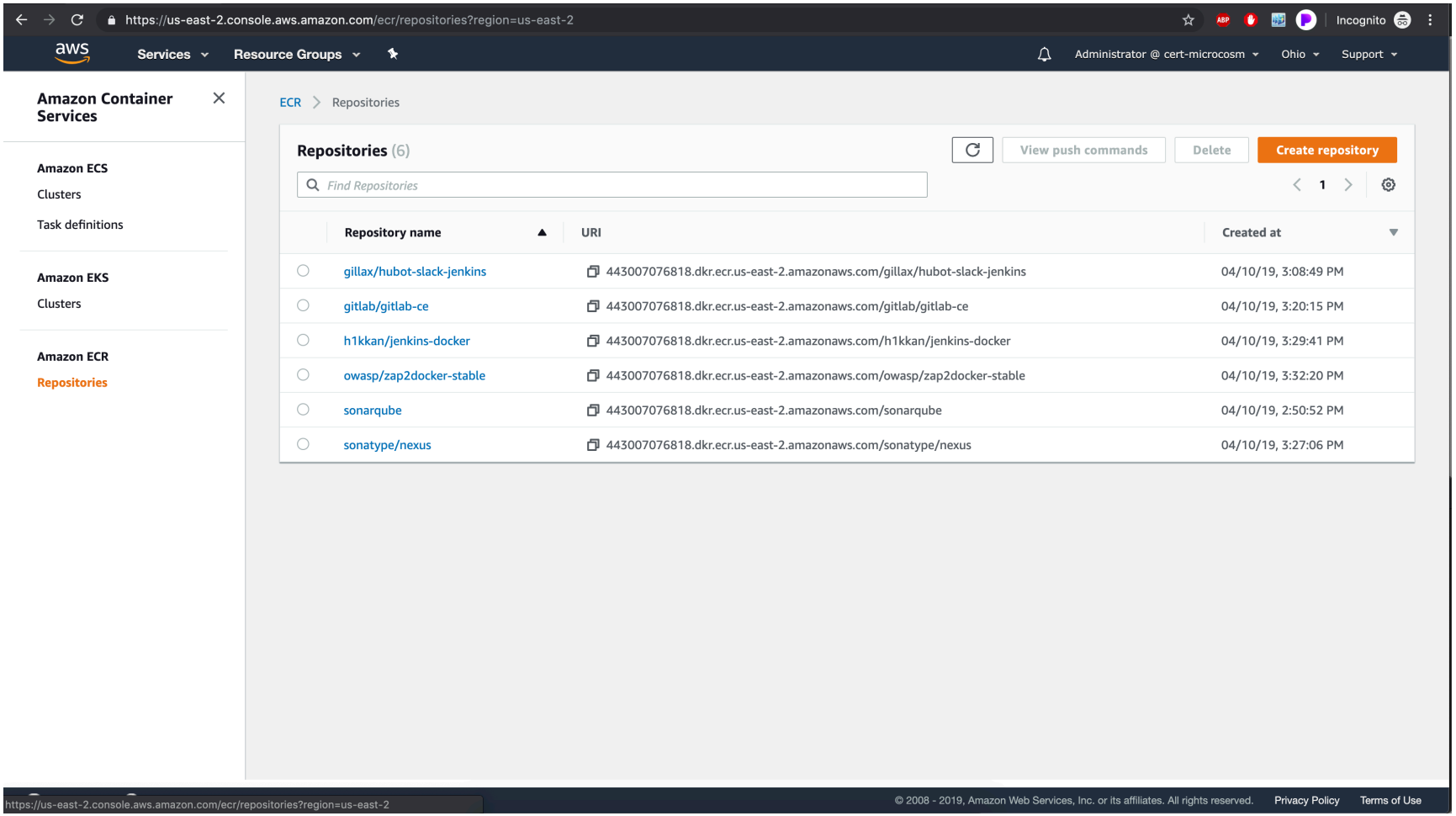
```
mac-topazpomme:devops-microcosm smorley$ aws ecr create-repository --repository-name h
1kkan/jenkins-docker
{
```

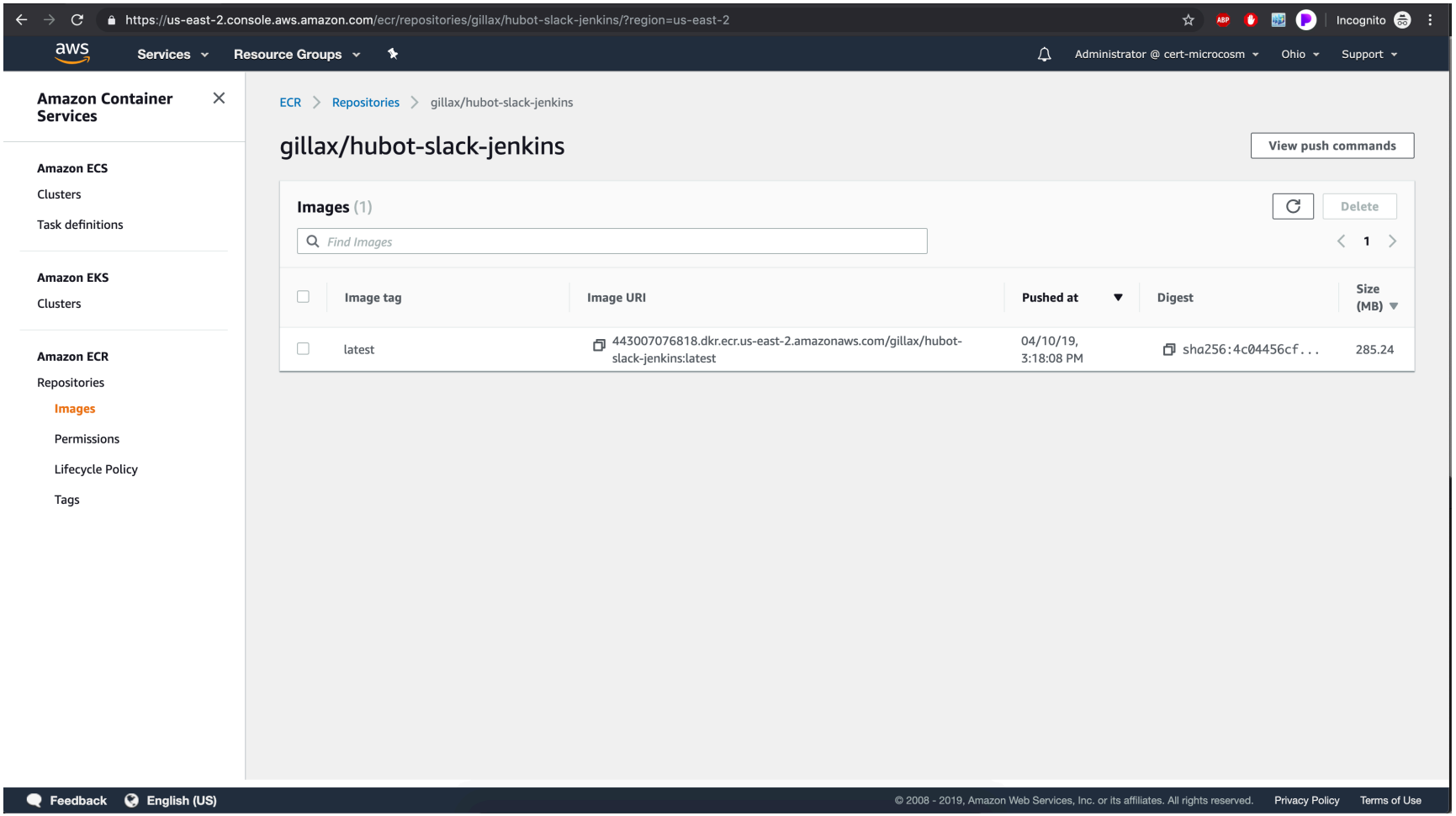
```
    "repository": {
        "repositoryArn": "arn:aws:ecr:us-east-2:443007076818:repository/h1kkan/jenkins
-docker",
        "registryId": "443007076818",
        "repositoryName": "h1kkan/jenkins-docker",
        "repositoryUri": "443007076818.dkr.ecr.us-east-2.amazonaws.com/h1kkan/jenkins-
docker",
        "createdAt": 1554931781.0
    }
}
[mac-topazpomme:devops-microcosm smorley$ docker tag h1kkan/jenkins-docker:lts 44300707]
6818.dkr.ecr.us-east-2.amazonaws.com/h1kkan/jenkins-docker:lts
[mac-topazpomme:devops-microcosm smorley$ docker push 443007076818.dkr.ecr.us-east-2.am]
azonaws.com/h1kkan/jenkins-docker:lts
The push refers to repository [443007076818.dkr.ecr.us-east-2.amazonaws.com/h1kkan/jen
kins-docker]
4717473633af: Pushed
01e56e46ef42: Pushed
0e21db749a1b: Pushed
1abb109c6333: Pushed
ea9254dfb470: Pushed
781f3534b8f0: Pushed
69ed8c046be0: Pushed
1cf9d4808e4e: Pushed
2b138287e104: Pushed
e674c0445366: Pushed
86cea145f8b3: Pushed
653304fe4cf3: Pushed
04b4fa9a597a: Pushed
0d790265fcc4: Pushed
d97f26ef1623: Pushed
e87475e5811f: Pushed
51e6d0a07219: Pushed
ba57bc494b22: Pushed
3443d6cf0f1f: Pushed
```

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

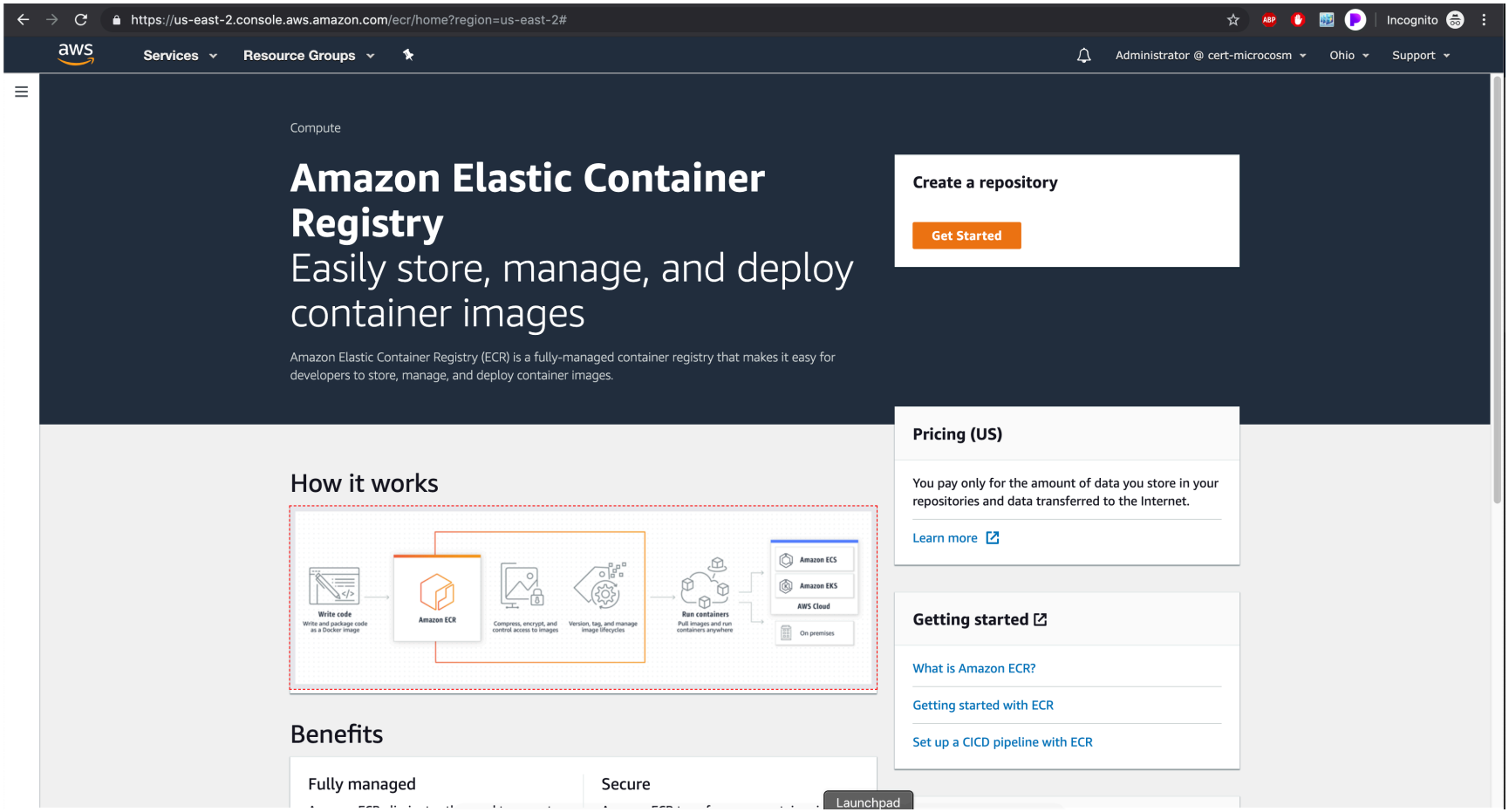
- View repositories and their details in teh AWS ECR Console

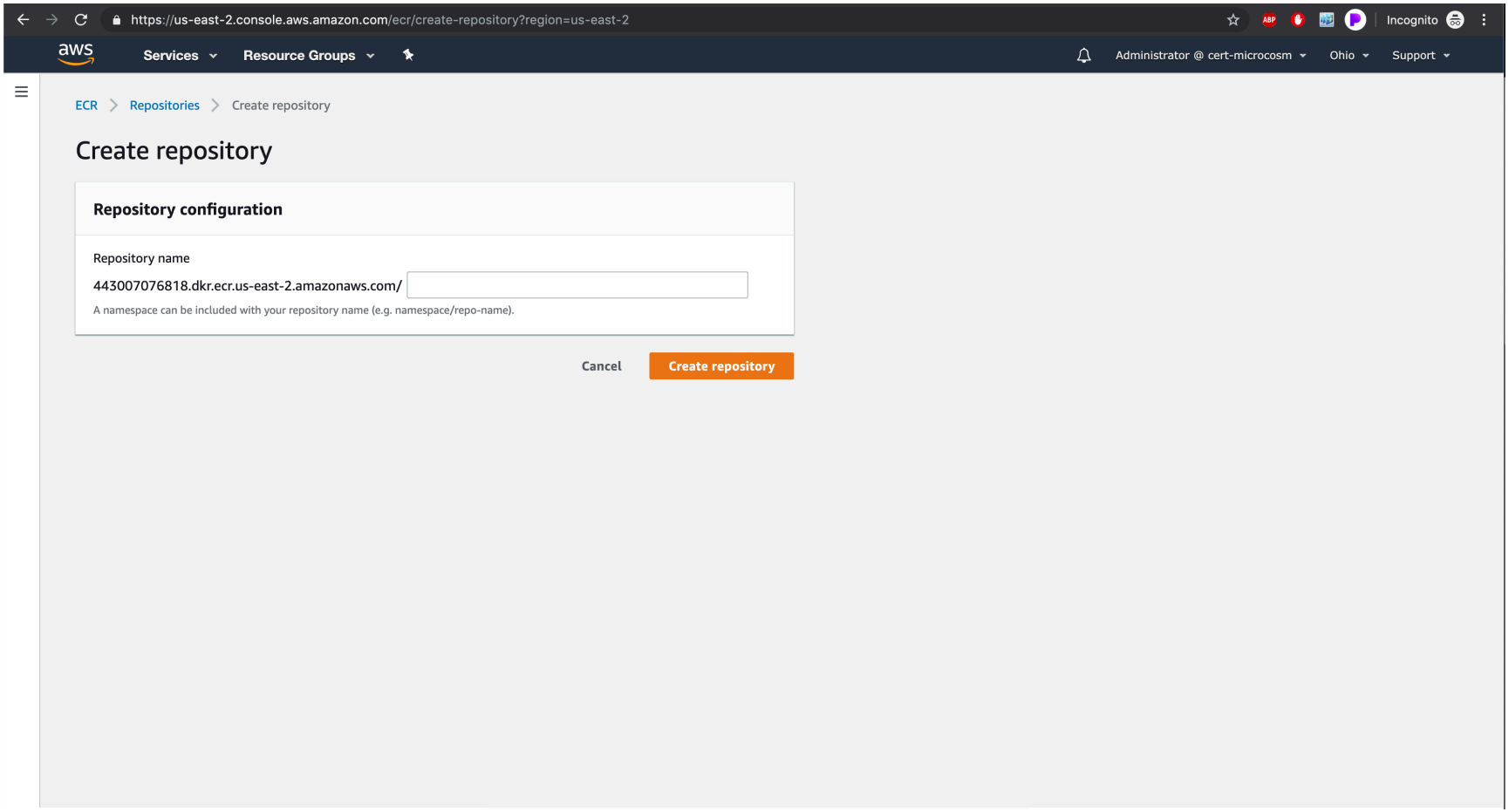


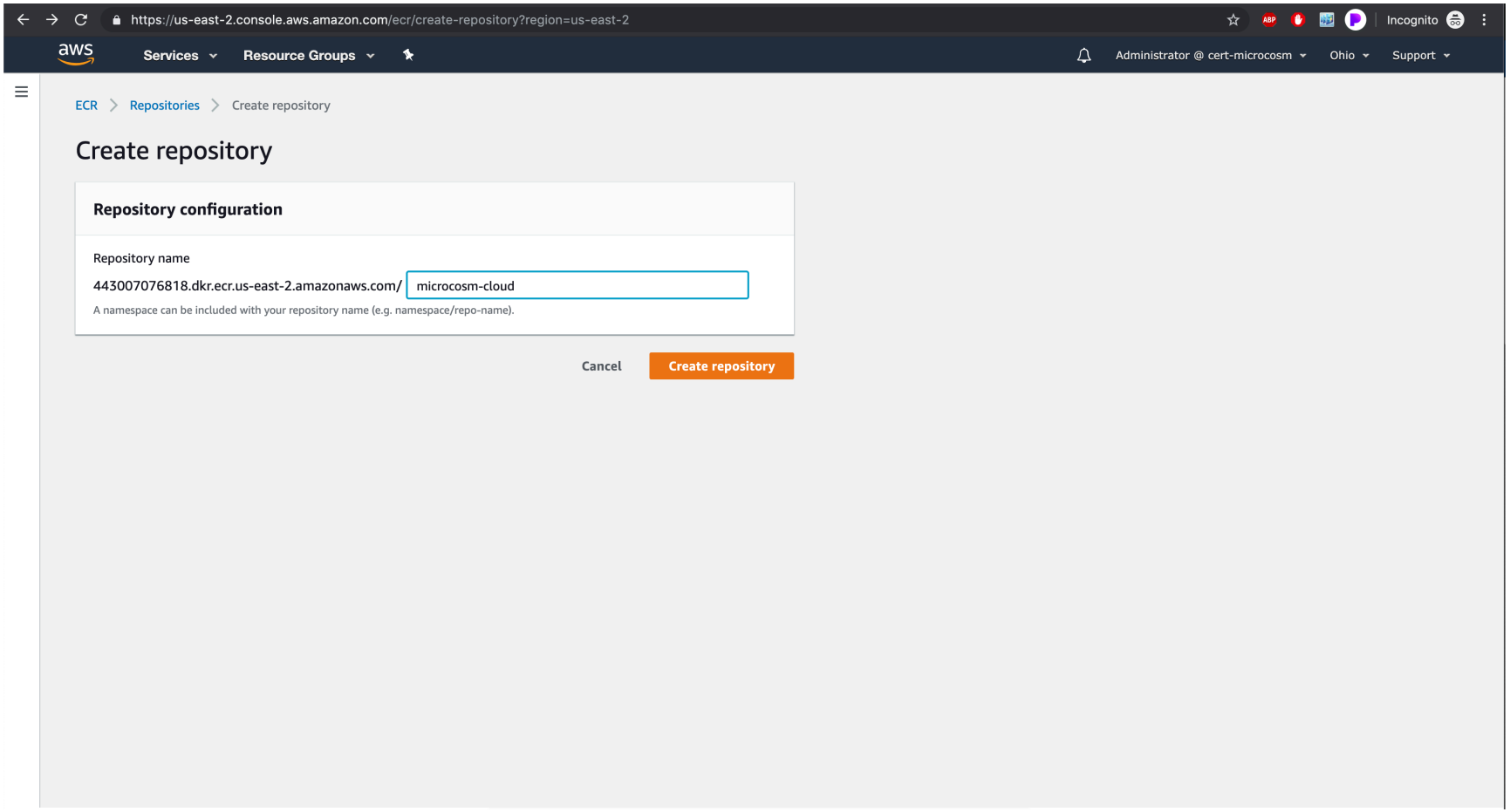

https://us-east-2.console.aws.amazon.com/ecr/repositories?region=us-east-2
Incognito
aws
Services
Resource Groups
Administrator @ cert-microcosm
Ohio
Support
Amazon Container Services
Amazon ECS
Clusters
Task definitions
Amazon EKS
Clusters
Amazon ECR
Repositories
ECR > Repositories
Repositories (6)
View push commands
Delete
Create repository
Find Repositories
1
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
https://us-east-2.console.aws.amazon.com/ecr/repositories?region=us-east-2
© 2008 - 2019, Amazon Web Services, Inc. or its affiliates. All rights reserved.
Privacy Policy
Terms of Use

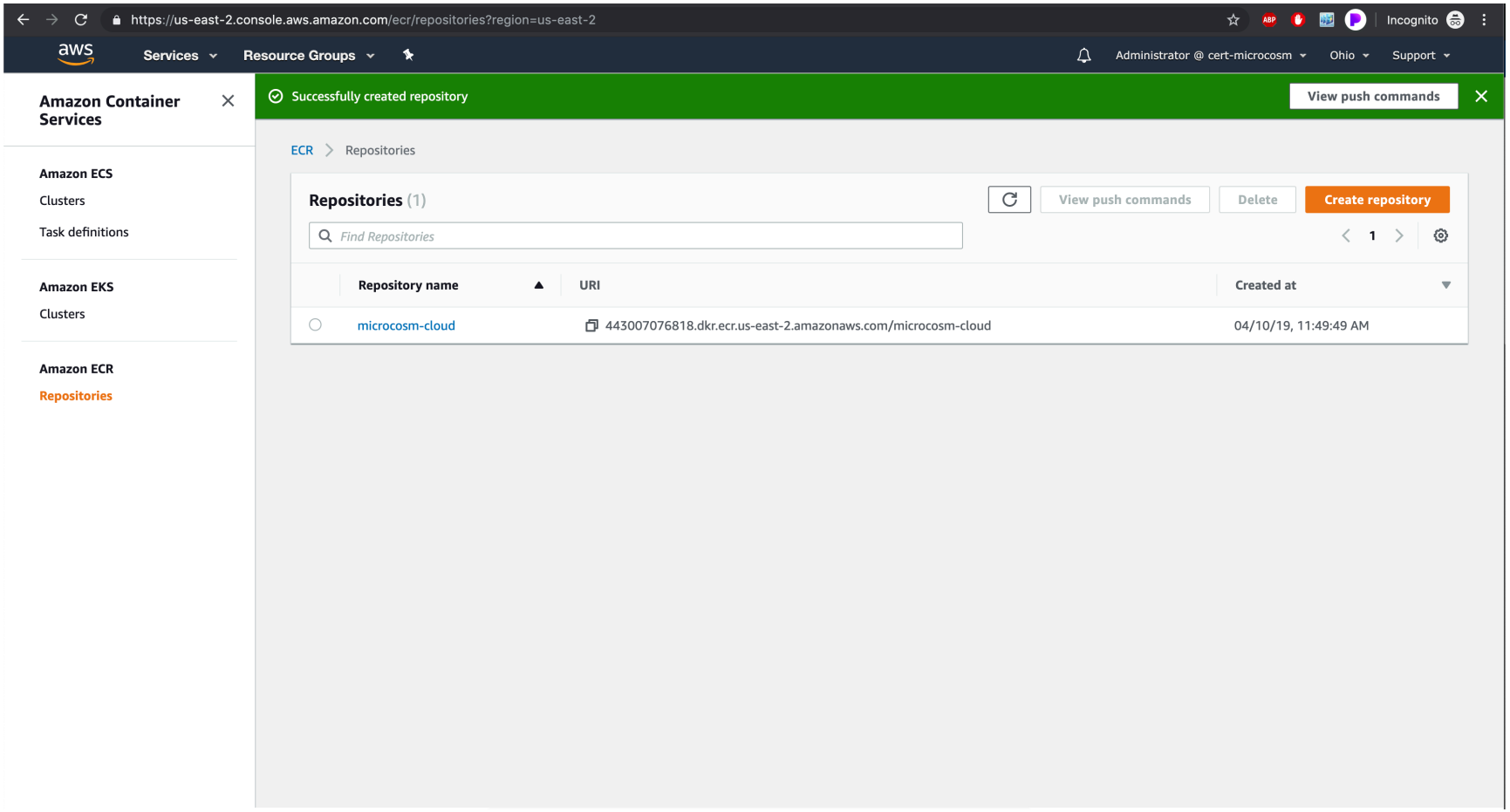


- NOTE: it is possible to create repositories using the AWS ECR Console, but it is faster to do using the CLI and the terminal.


https://us-east-2.console.aws.amazon.com/ecr/home?region=us-east-2#
Incognito
aws
Services
Resource Groups
Administrator @ cert-microcosm
Ohio
Support
Compute
Amazon Elastic Container Registry
Easily store, manage, and deploy container images
Amazon Elastic Container Registry (ECR) is a fully-managed container registry that makes it easy for developers to store, manage, and deploy container images.
Create a repository
Get Started
Pricing (US)
You pay only for the amount of data you store in your repositories and data transferred to the Internet.
Learn more
How it works
Write code
Write and package code as a Docker image
Amazon ECR
Compress, encrypt, and control access to images
Version, tag, and manage image lifecycles
Run containers
Pull images and run containers anywhere
Amazon ECS
Amazon EKS
AWS Cloud
On premises
Getting started
What is Amazon ECR?
Getting started with ECR
Set up a CICD pipeline with ECR
Benefits
Fully managed
Secure
Launchpad


https://us-east-2.console.aws.amazon.com/ecr/create-repository?region=us-east-2
Incognito
aws
Services
Resource Groups
Administrator @ cert-microcosm
Ohio
Support
ECR
Repositories
Create repository
Create repository
Repository configuration
Repository name
443007076818.dkr.ecr.us-east-2.amazonaws.com/
A namespace can be included with your repository name (e.g. namespace/repo-name).
Cancel
Create repository


https://us-east-2.console.aws.amazon.com/ecr/create-repository?region=us-east-2
Incognito
aws
Services
Resource Groups
Administrator @ cert-microcosm
Ohio
Support
ECR
Repositories
Create repository
Create repository
Repository configuration
Repository name
443007076818.dkr.ecr.us-east-2.amazonaws.com/
microcosm-cloud
A namespace can be included with your repository name (e.g. namespace/repo-name).
Cancel
Create repository

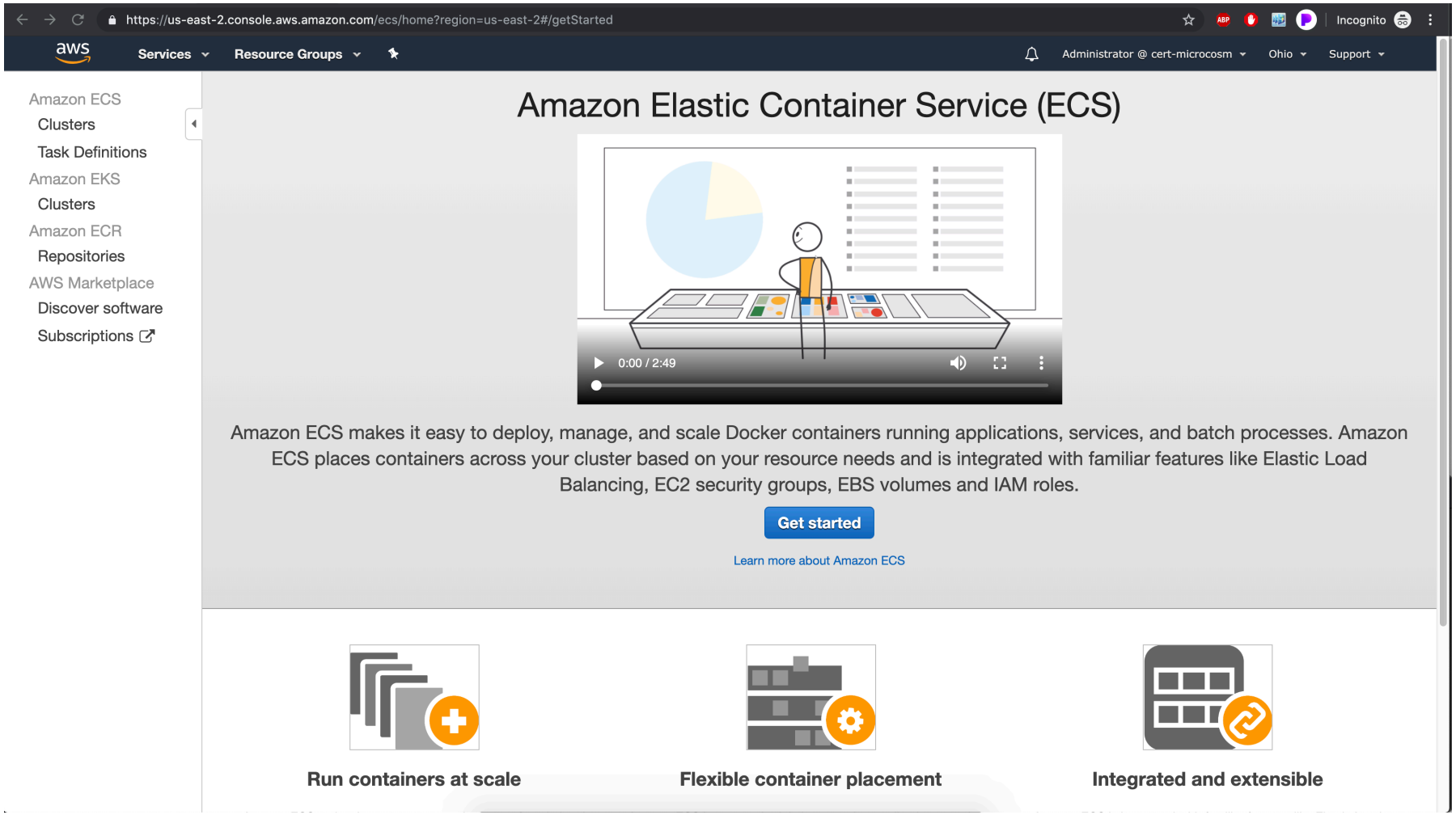


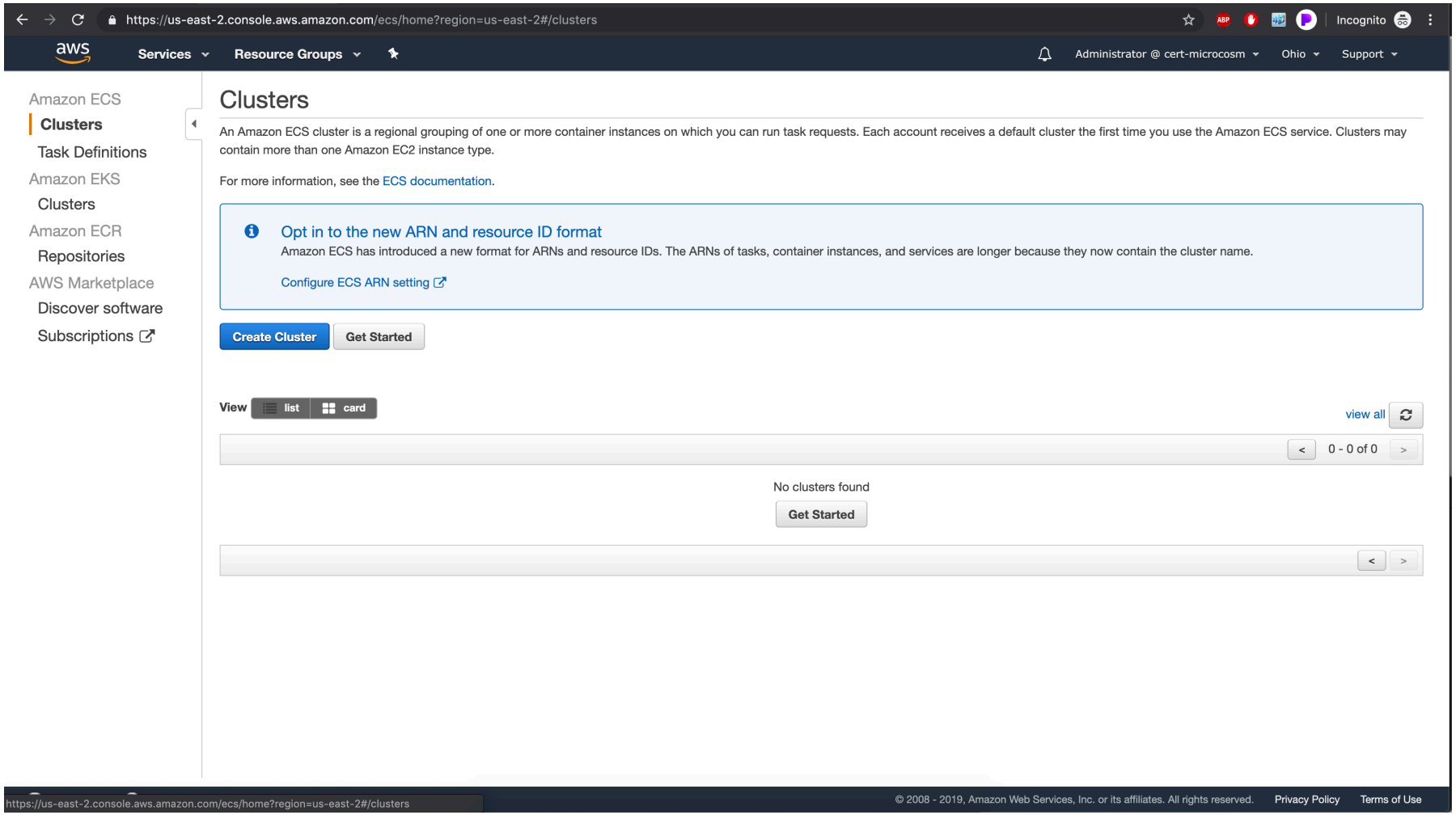
Return to Table of Contents

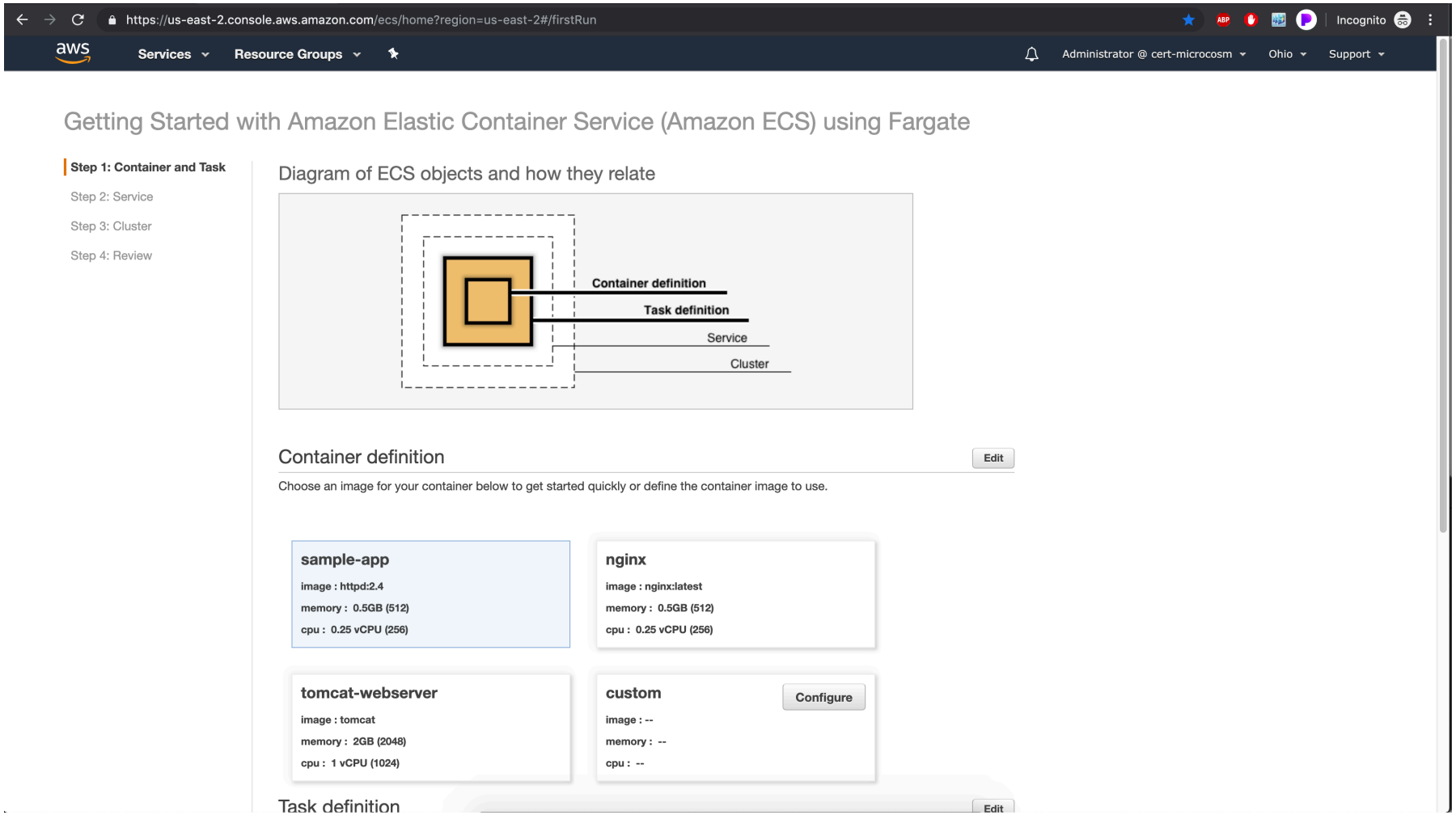
ECS Get Started Wizard

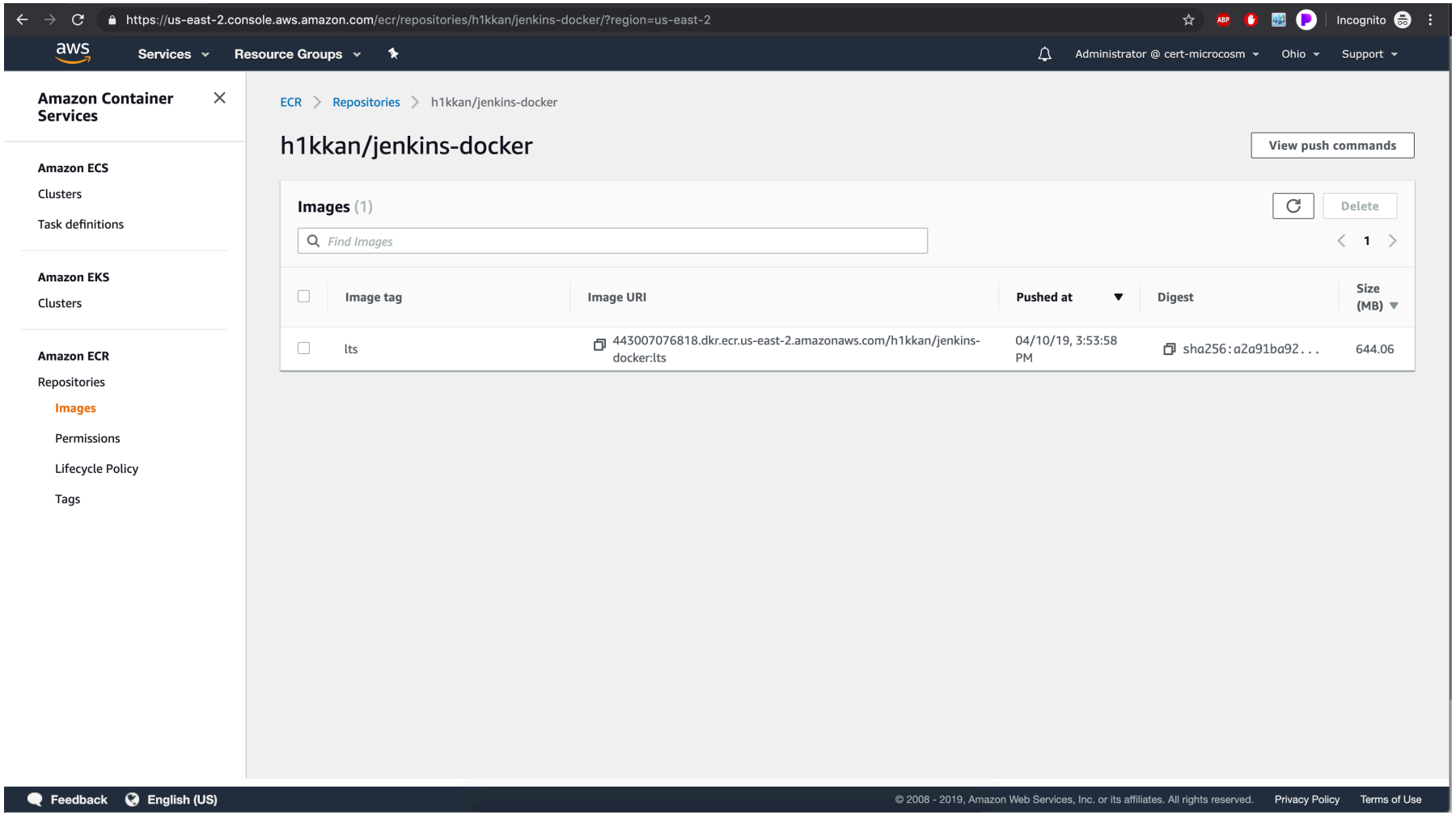
Using Get Started Wizard

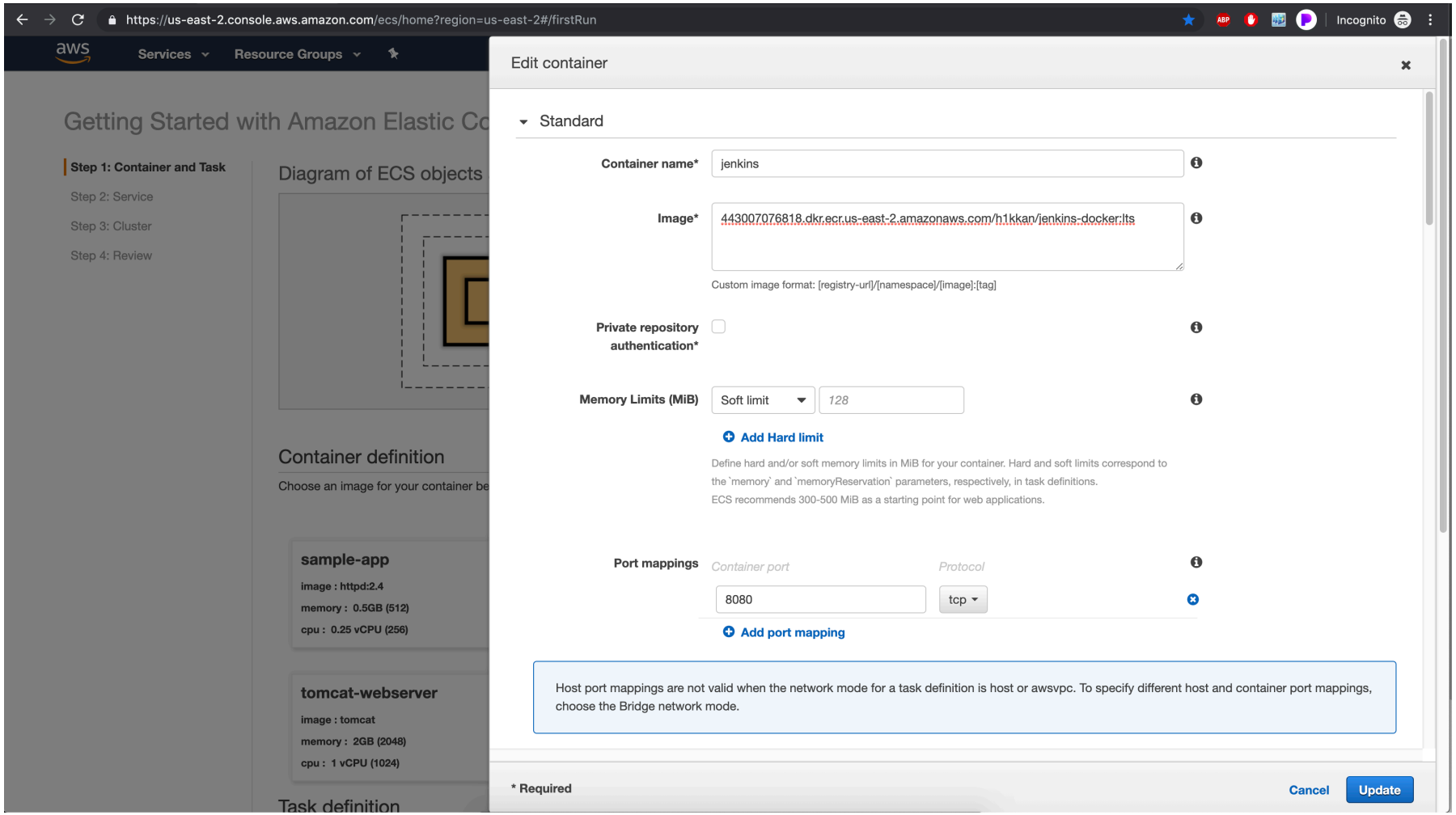
Follow the screen shots for reference

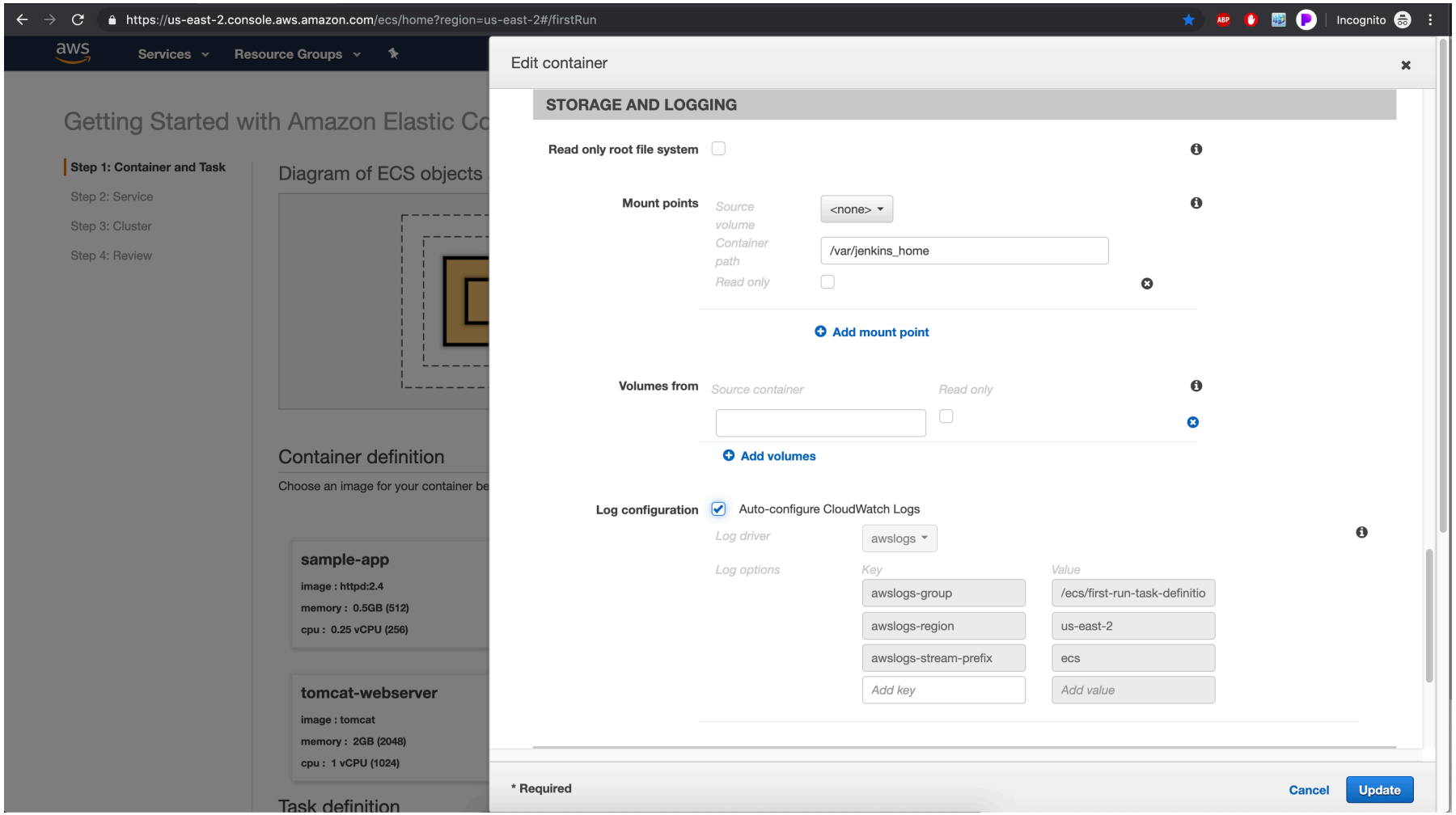

https://us-east-2.console.aws.amazon.com/ecs/home?region=us-east-2#/getStarted
Incognito
aws
Services
Resource Groups
Administrator @ cert-microcosm
Ohio
Support
Amazon ECS
Clusters
Task Definitions
Amazon EKS
Clusters
Amazon ECR
Repositories
AWS Marketplace
Discover software
Subscriptions
Amazon Elastic Container Service (ECS)
0:00 / 2:49
Amazon ECS makes it easy to deploy, manage, and scale Docker containers running applications, services, and batch processes. Amazon ECS places containers across your cluster based on your resource needs and is integrated with familiar features like Elastic Load Balancing, EC2 security groups, EBS volumes and IAM roles.
Get started
Learn more about Amazon ECS
Run containers at scale
Flexible container placement
Integrated and extensible

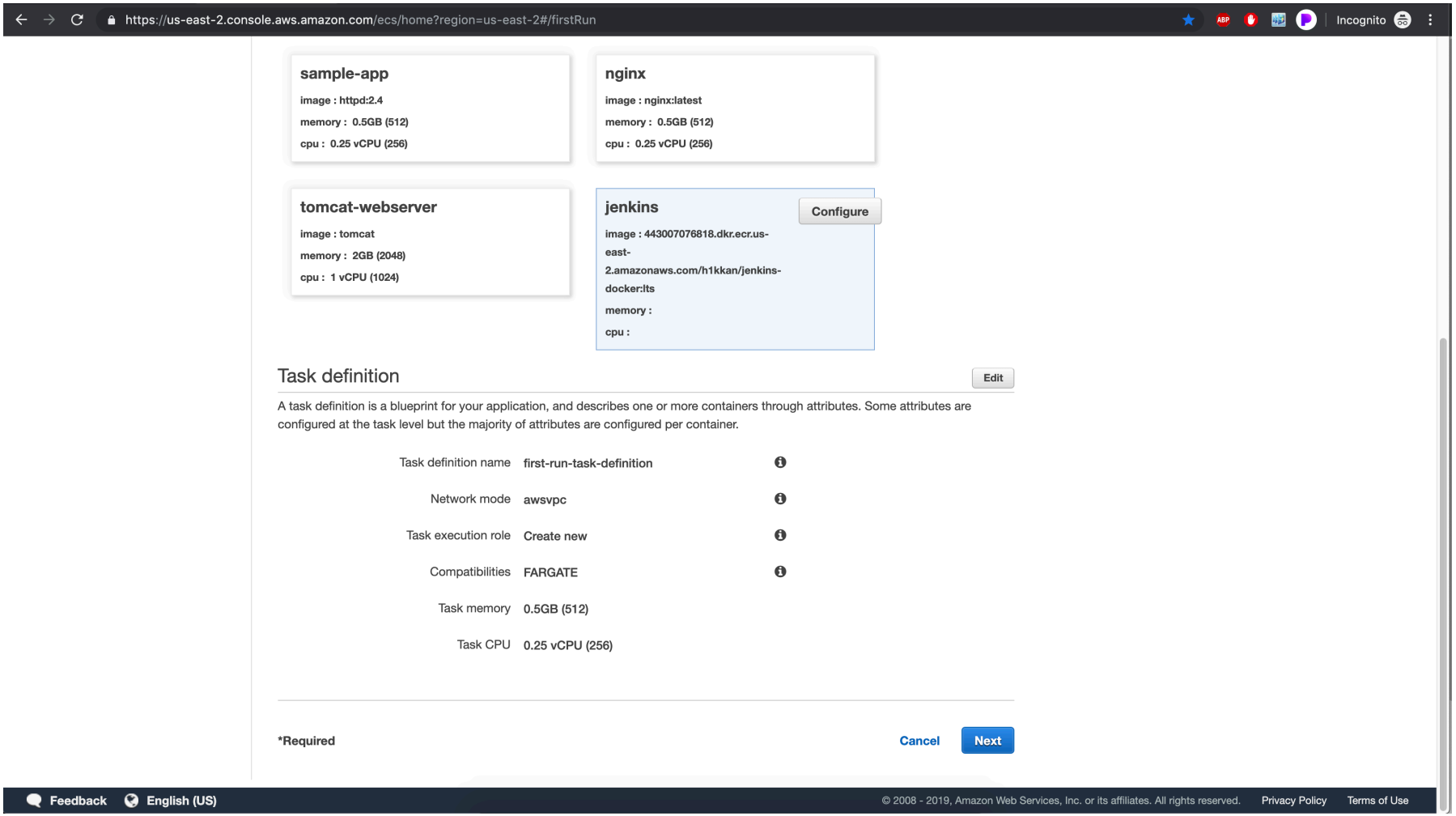

https://us-east-2.console.aws.amazon.com/ecs/home?region=us-east-2#/clusters
Incognito
aws
Services
Resource Groups
Administrator @ cert-microcosm
Ohio
Support
Amazon ECS
Clusters
Task Definitions
Amazon EKS
Clusters
Amazon ECR
Repositories
AWS Marketplace
Discover software
Subscriptions
Clusters
An 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
view all
0 - 0 of 0
No clusters found
Get Started
https://us-east-2.console.aws.amazon.com/ecs/home?region=us-east-2#/clusters
© 2008 - 2019, Amazon Web Services, Inc. or its affiliates. All rights reserved.
Privacy Policy
Terms of Use


https://us-east-2.console.aws.amazon.com/ecs/home?region=us-east-2#/firstRun
Incognito
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
Task definition
Service
Cluster
Container definition
Edit
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
Configure
image : --
memory : --
cpu : --
Task definition
Edit


https://us-east-2.console.aws.amazon.com/ecr/repositories/h1kkan/jenkins-docker/?region=us-east-2
Incognito
aws
Services
Resource Groups
Administrator @ cert-microcosm
Ohio
Support
Amazon Container Services
Amazon ECS
Clusters
Task definitions
Amazon EKS
Clusters
Amazon ECR
Repositories
Images
Permissions
Lifecycle Policy
Tags
ECR > Repositories > h1kkan/jenkins-docker
h1kkan/jenkins-docker
View push commands
Images (1)
Delete
Find 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:a2a91ba92...
644.06
Feedback
English (US)
© 2008 - 2019, Amazon Web Services, Inc. or its affiliates. All rights reserved.
Privacy Policy
Terms of Use


https://us-east-2.console.aws.amazon.com/ecs/home?region=us-east-2#/firstRun
Incognito
aws
Services
Resource Groups
Getting Started with Amazon Elastic Co
Step 1: Container and Task
Step 2: Service
Step 3: Cluster
Step 4: Review
Diagram of ECS objects
Container definition
Choose an image for your container be
sample-app
image : httpd:2.4
memory : 0.5GB (512)
cpu : 0.25 vCPU (256)
tomcat-webserver
image : tomcat
memory : 2GB (2048)
cpu : 1 vCPU (1024)
Task definition
Edit container
Standard
Container name*
jenkins
Image*
443007076818.dkr.ecr.us-east-2.amazonaws.com/h1kkan/jenkins-docker:lts
Custom image format: [registry-url]/[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


https://us-east-2.console.aws.amazon.com/ecs/home?region=us-east-2#/firstRun
Incognito
aws
Services
Resource Groups
Getting Started with Amazon Elastic Co
Step 1: Container and Task
Step 2: Service
Step 3: Cluster
Step 4: Review
Diagram of ECS objects
Container definition
Choose an image for your container be
sample-app
image : httpd:2.4
memory : 0.5GB (512)
cpu : 0.25 vCPU (256)
tomcat-webserver
image : tomcat
memory : 2GB (2048)
cpu : 1 vCPU (1024)
Task definition
Edit container
STORAGE AND LOGGING
Read only root file system
Mount points
Source volume
Container path
Read only
<none>
/var/jenkins_home
Add mount point
Volumes from
Source container
Read only
Add volumes
Log configuration
Auto-configure CloudWatch Logs
Log driver
awslogs
Log options
Key
Value
awslogs-group
/ecs/first-run-task-definitio
awslogs-region
us-east-2
awslogs-stream-prefix
ecs
Add key
Add value
* Required
Cancel
Update

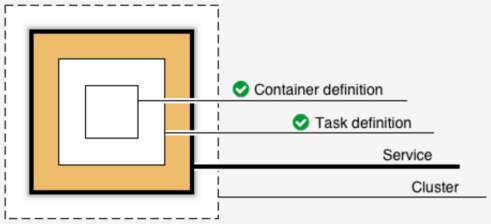

https://us-east-2.console.aws.amazon.com/ecs/home?region=us-east-2#/firstRun
Incognito
sample-app
image : httpd:2.4
memory : 0.5GB (512)
cpu : 0.25 vCPU (256)
nginx
image : nginx:latest
memory : 0.5GB (512)
cpu : 0.25 vCPU (256)
tomcat-webserver
image : tomcat
memory : 2GB (2048)
cpu : 1 vCPU (1024)
jenkins
Configure
image : 443007076818.dkr.ecr.us-east-2.amazonaws.com/h1kkan/jenkins-docker:lts
memory :
cpu :
Task definition
Edit
A task definition is a blueprint for your application, and describes one or more containers through attributes. Some attributes are configured at the task level but the majority of attributes are configured per container.
Task definition name first-run-task-definition
Network mode awsvpc
Task execution role Create new
Compatibilities FARGATE
Task memory 0.5GB (512)
Task CPU 0.25 vCPU (256)
*Required
Cancel
Next
Feedback
English (US)
© 2008 - 2019, Amazon Web Services, Inc. or its affiliates. All rights reserved.
Privacy Policy
Terms of Use

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

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

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

Diagram of ECS objects and how they relate



Define your service

Edit

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)

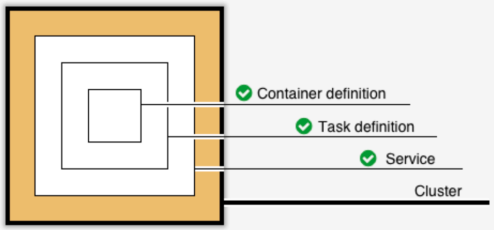
 Privacy Policy Terms of Use

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

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

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

Diagram of ECS objects and how they relate



Configure your cluster

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

To see key differences between Fargate and standard ECS clusters, see the Amazon ECS documentation.

Cluster name: microcosm

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

VPC ID: Automatically create new

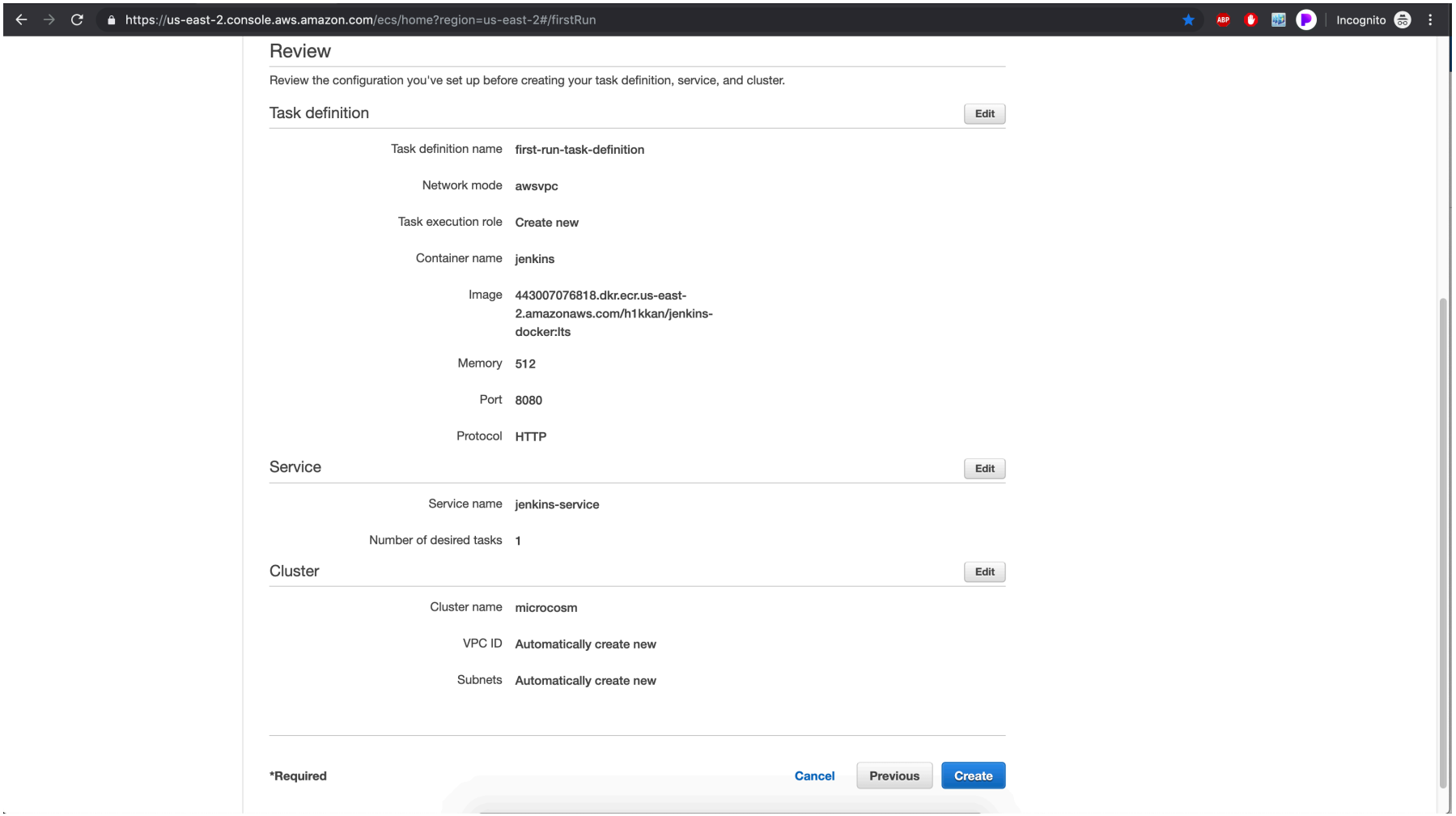
Subnets: Automatically create new

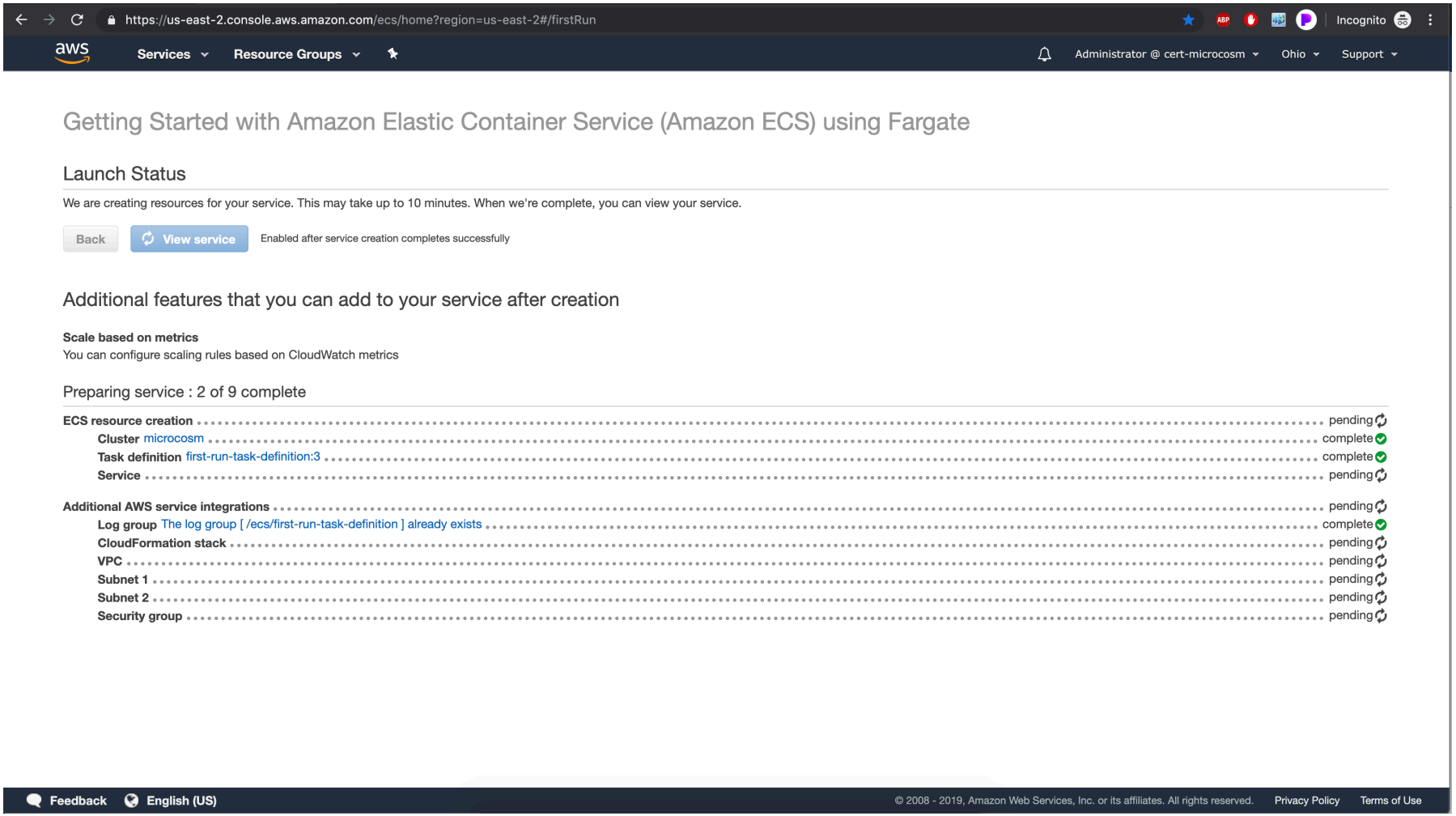
*Required

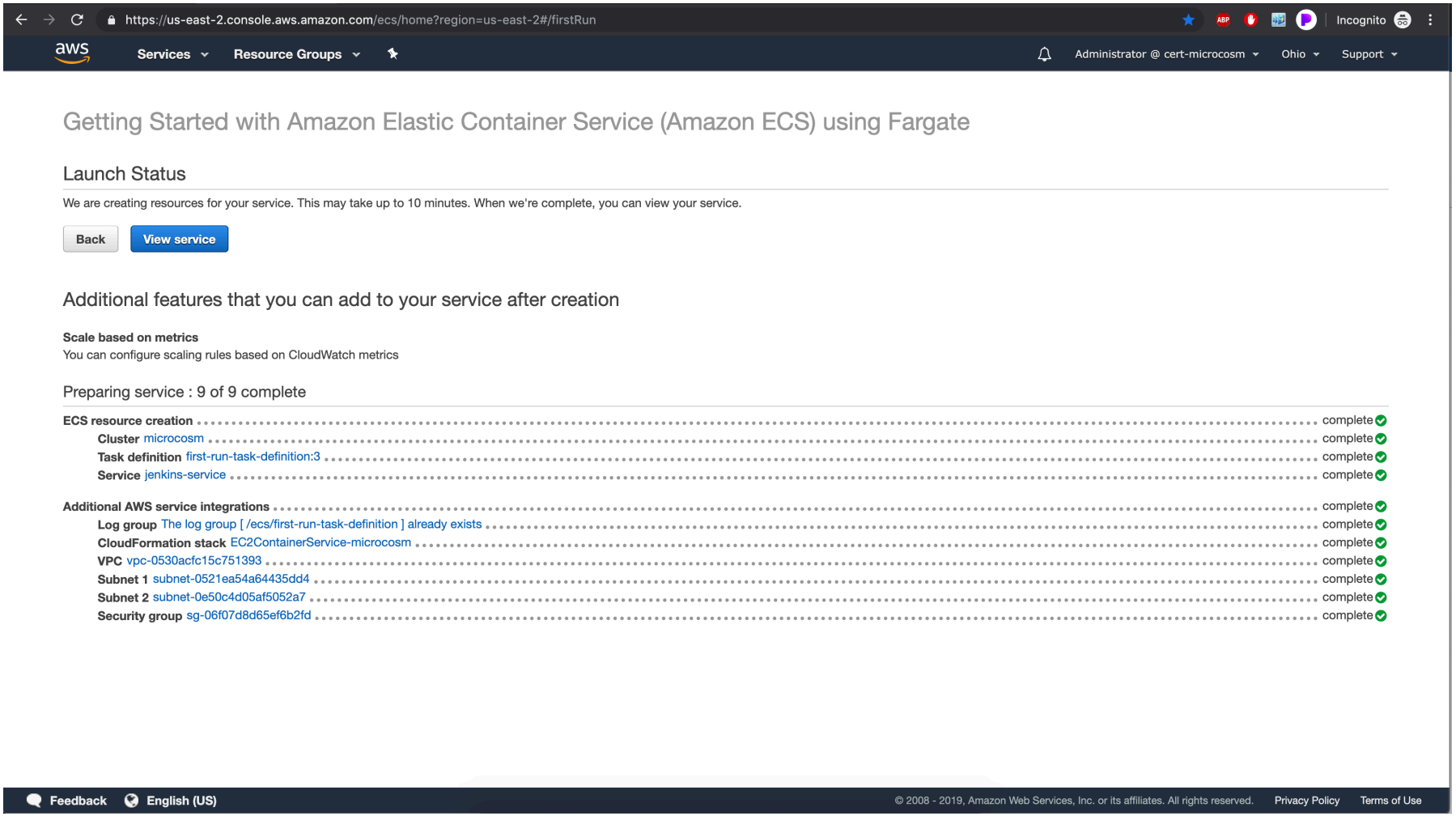
Cancel | Previous | Next

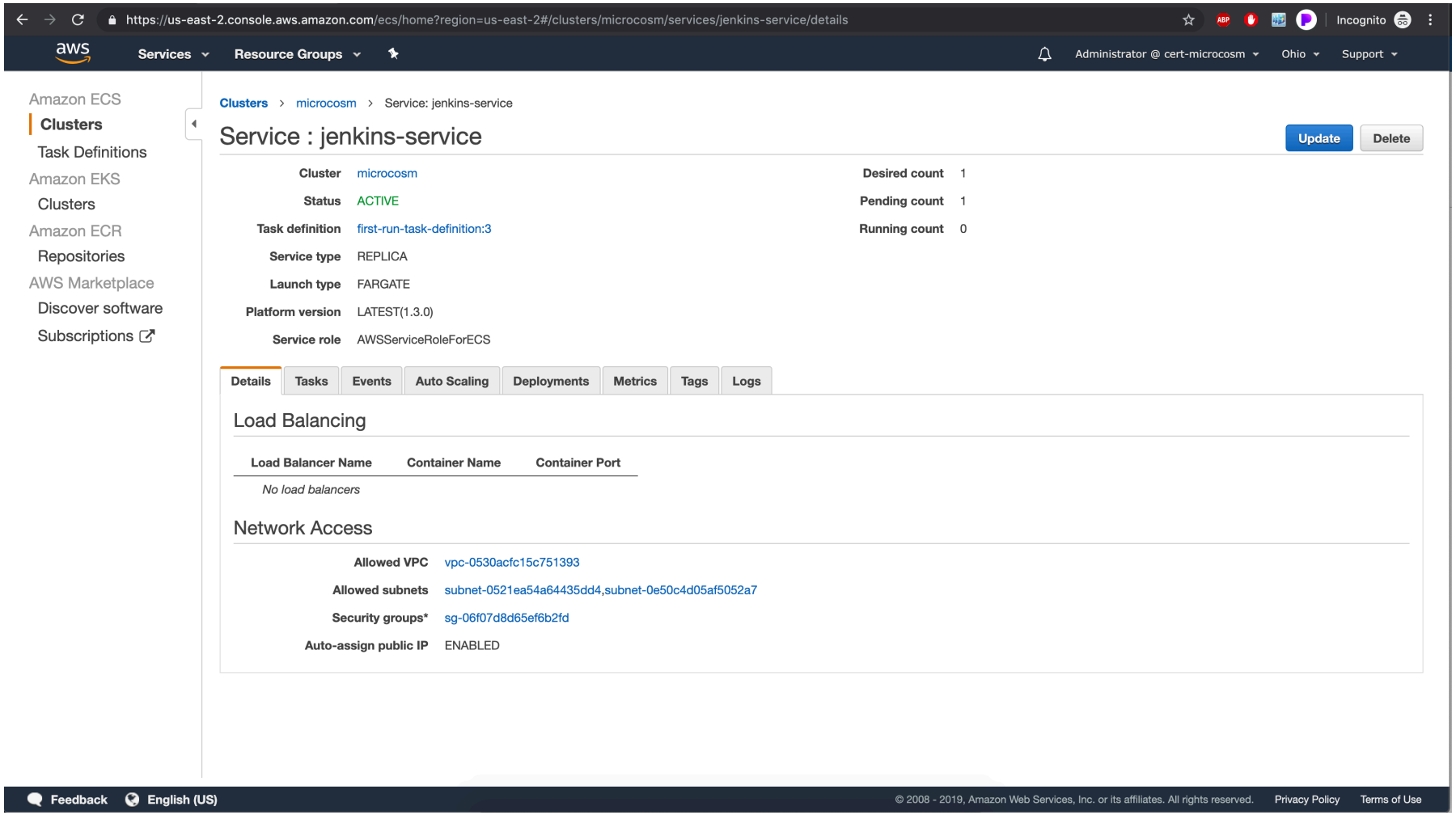
Feedback | English (US)

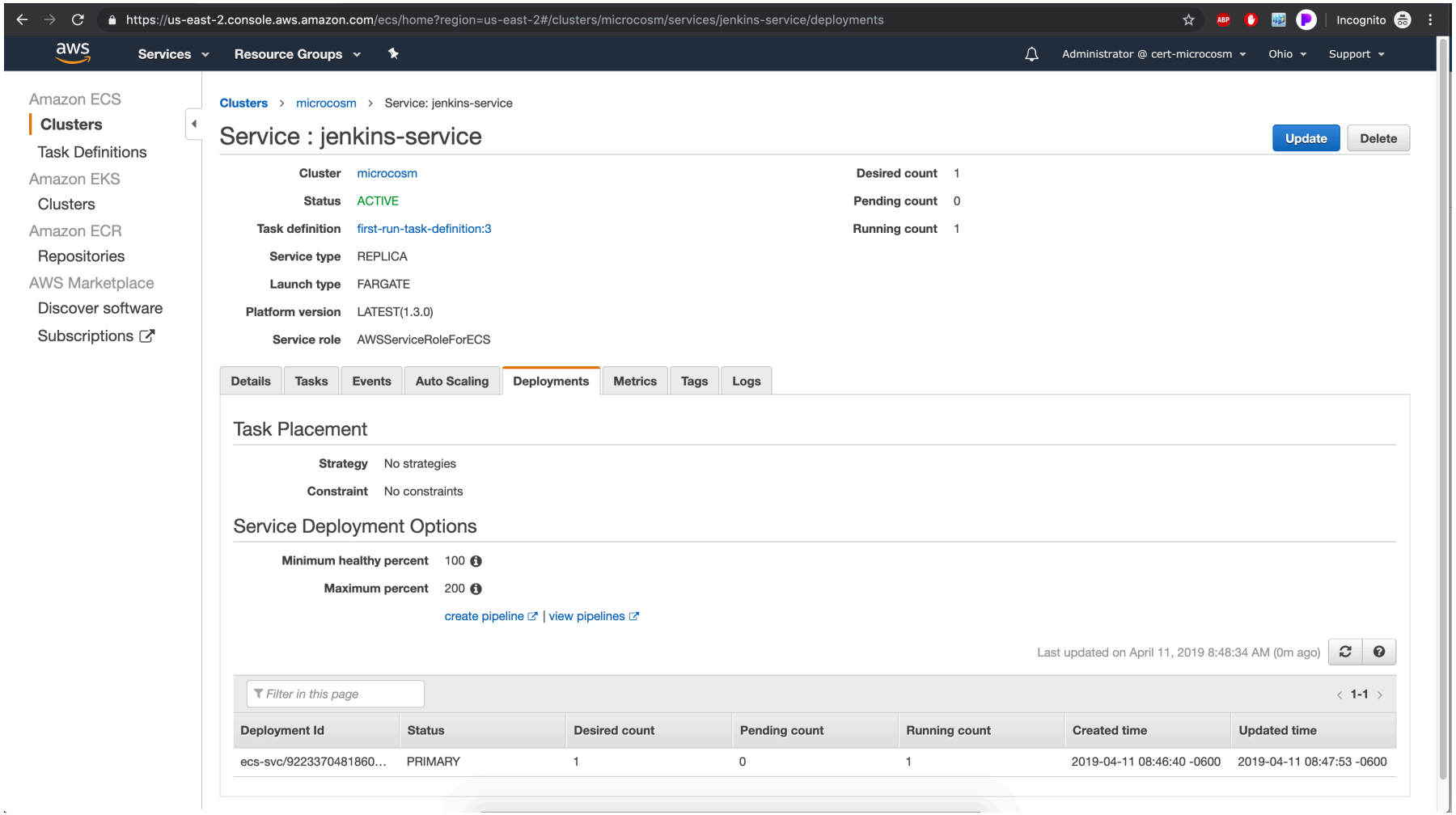
 Privacy Policy | Terms of Use

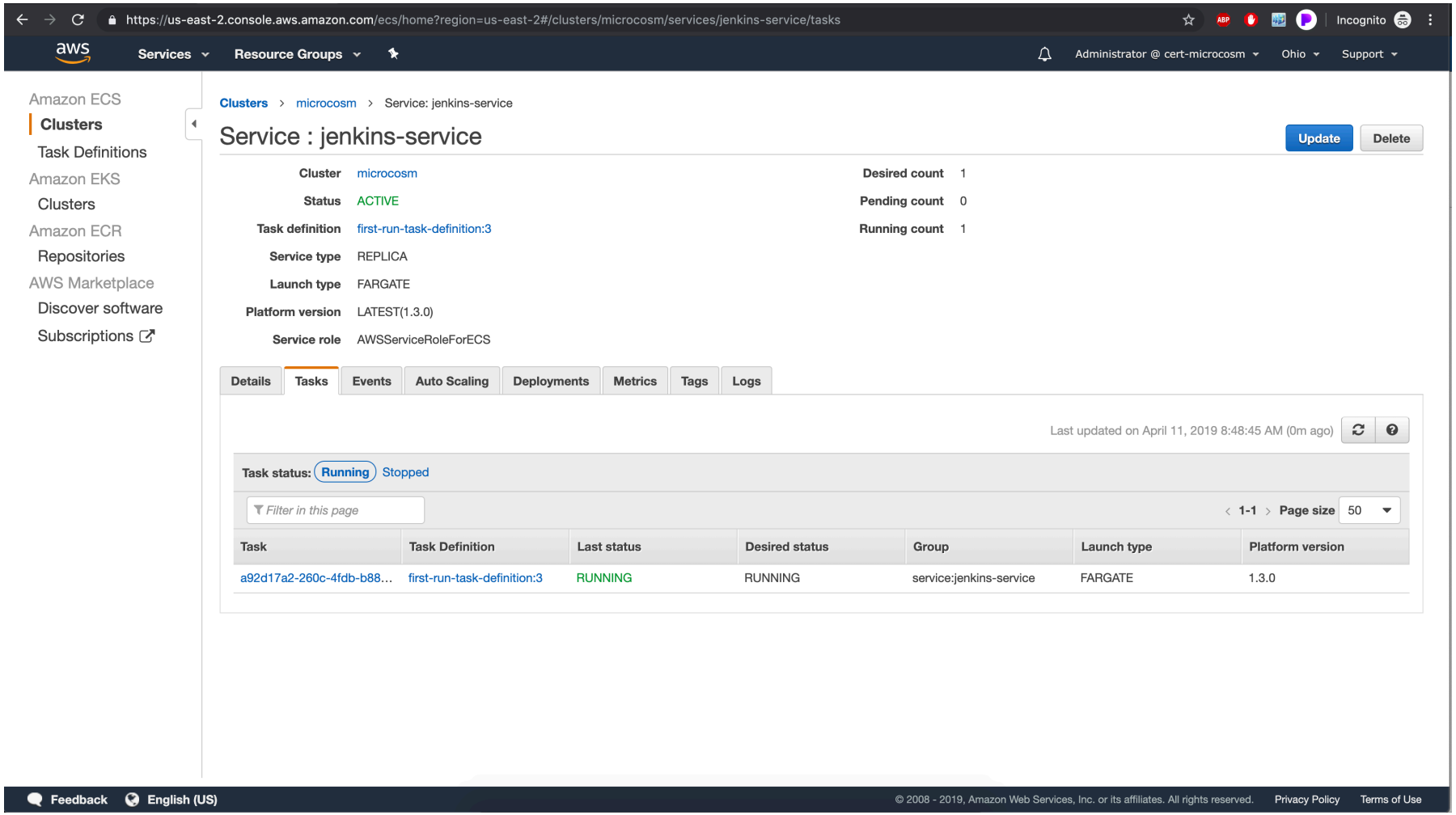

https://us-east-2.console.aws.amazon.com/ecs/home?region=us-east-2#/firstRun
Incognito
Review
Review the configuration you've set up before creating your task definition, service, and cluster.
Task definition
Edit
Task definition name first-run-task-definition
Network mode awsvpc
Task execution role Create new
Container name jenkins
Image 443007076818.dkr.ecr.us-east-2.amazonaws.com/h1kkan/jenkins-docker:lts
Memory 512
Port 8080
Protocol HTTP
Service
Edit
Service name jenkins-service
Number of desired tasks 1
Cluster
Edit
Cluster name microcosm
VPC ID Automatically create new
Subnets Automatically create new
*Required
Cancel
Previous
Create

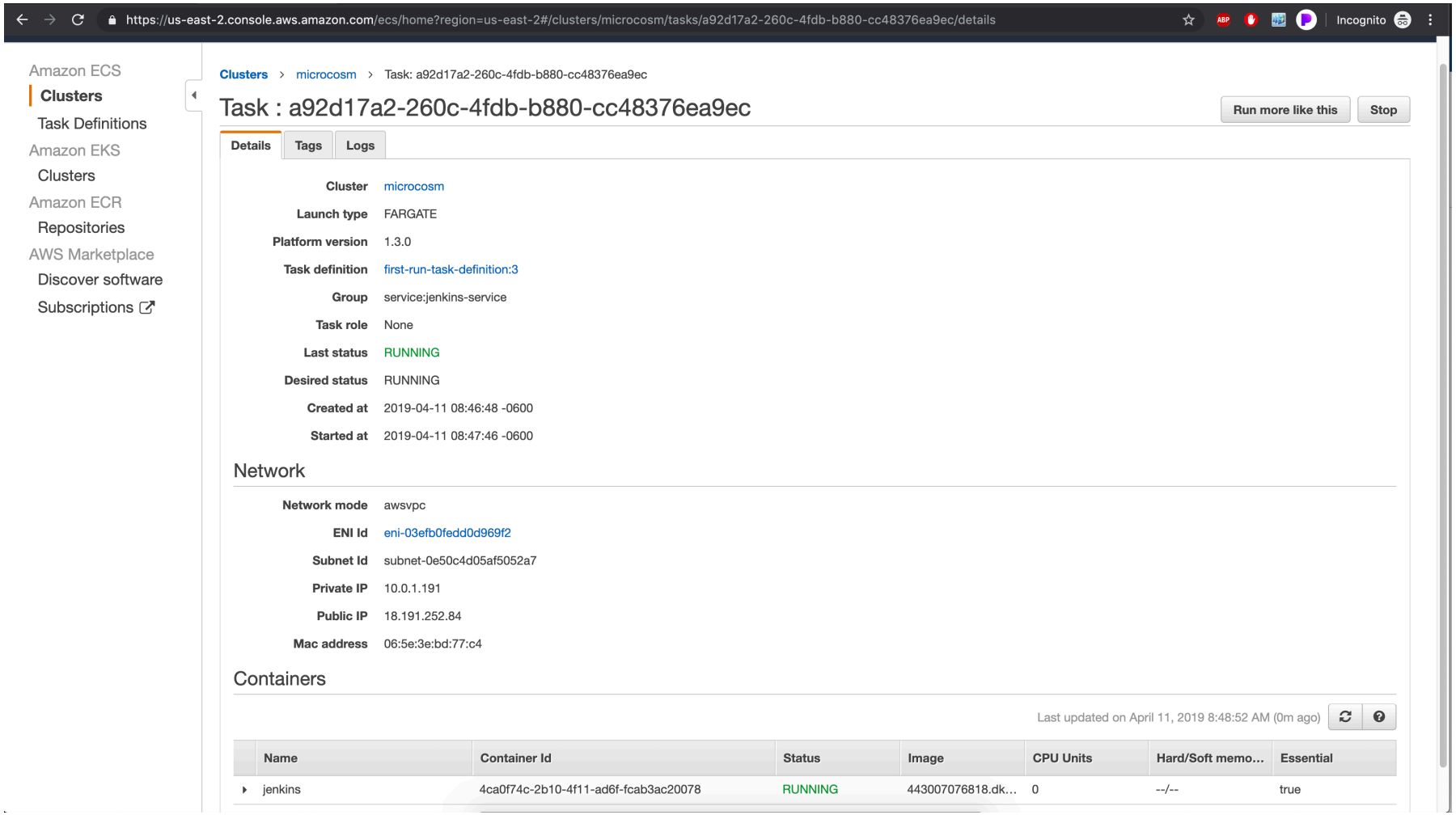

https://us-east-2.console.aws.amazon.com/ecs/home?region=us-east-2#/firstRun
Incognito
aws
Services
Resource Groups
Administrator @ cert-microcosm
Ohio
Support
Getting Started with Amazon Elastic Container Service (Amazon ECS) using Fargate
Launch Status
We are creating resources for your service. This 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 pending
Additional AWS service integrations pending
Log group The log group [/ecs/first-run-task-definition] already exists complete
CloudFormation stack pending
VPC pending
Subnet 1 pending
Subnet 2 pending
Security group pending
Feedback
English (US)
© 2008 - 2019, Amazon Web Services, Inc. or its affiliates. All rights reserved.
Privacy Policy
Terms of Use

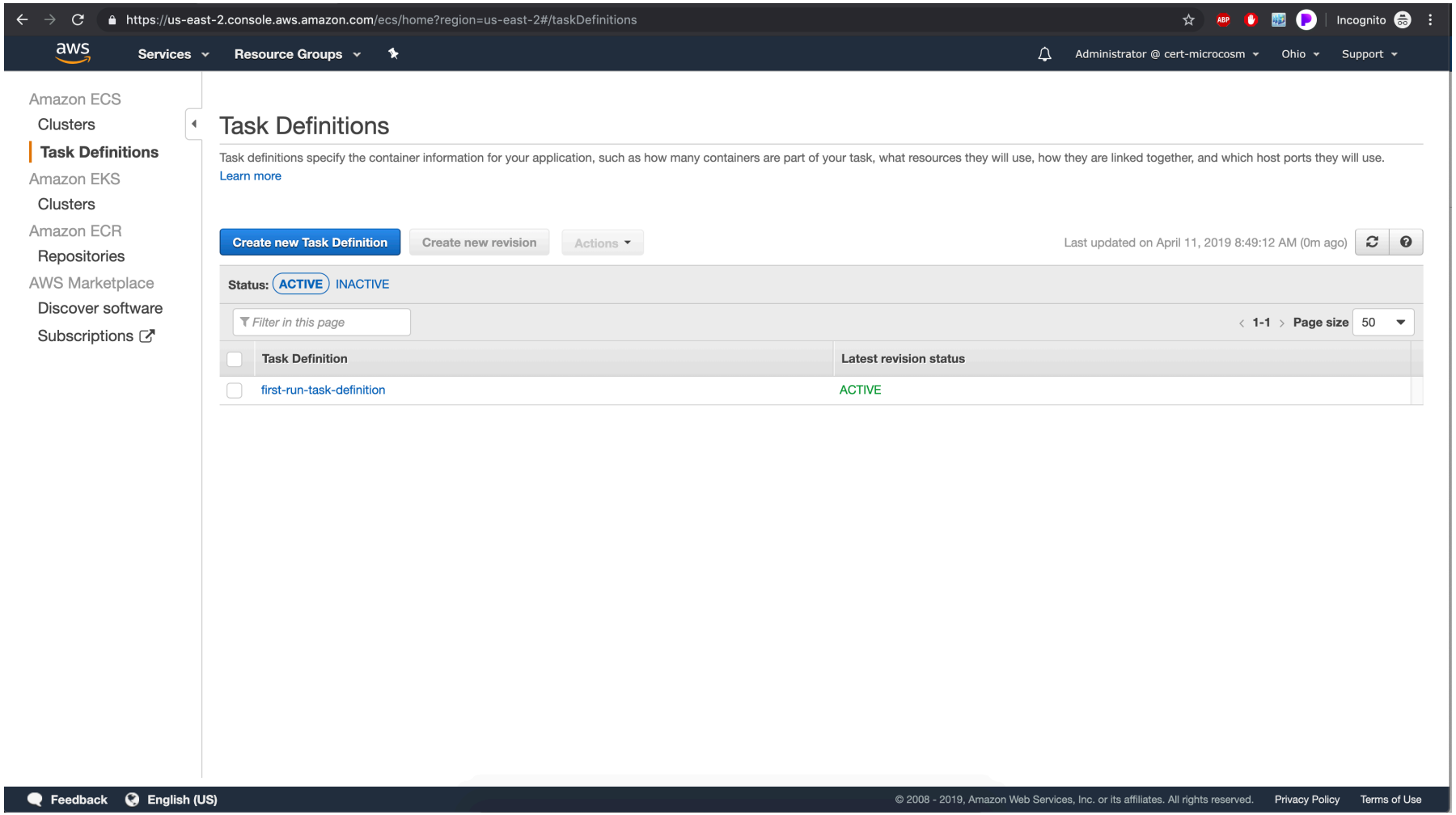

https://us-east-2.console.aws.amazon.com/ecs/home?region=us-east-2#/firstRun
Incognito
aws
Services
Resource Groups
Administrator @ cert-microcosm
Ohio
Support
Getting Started with Amazon Elastic Container Service (Amazon ECS) using Fargate
Launch Status
We are creating resources for your service. This 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-0530acfc15c751393 complete
Subnet 1 subnet-0521ea54a64435dd4 complete
Subnet 2 subnet-0e50c4d05af5052a7 complete
Security group sg-06f07d8d65ef6b2fd complete
Feedback
English (US)
© 2008 - 2019, Amazon Web Services, Inc. or its affiliates. All rights reserved.
Privacy Policy
Terms of Use

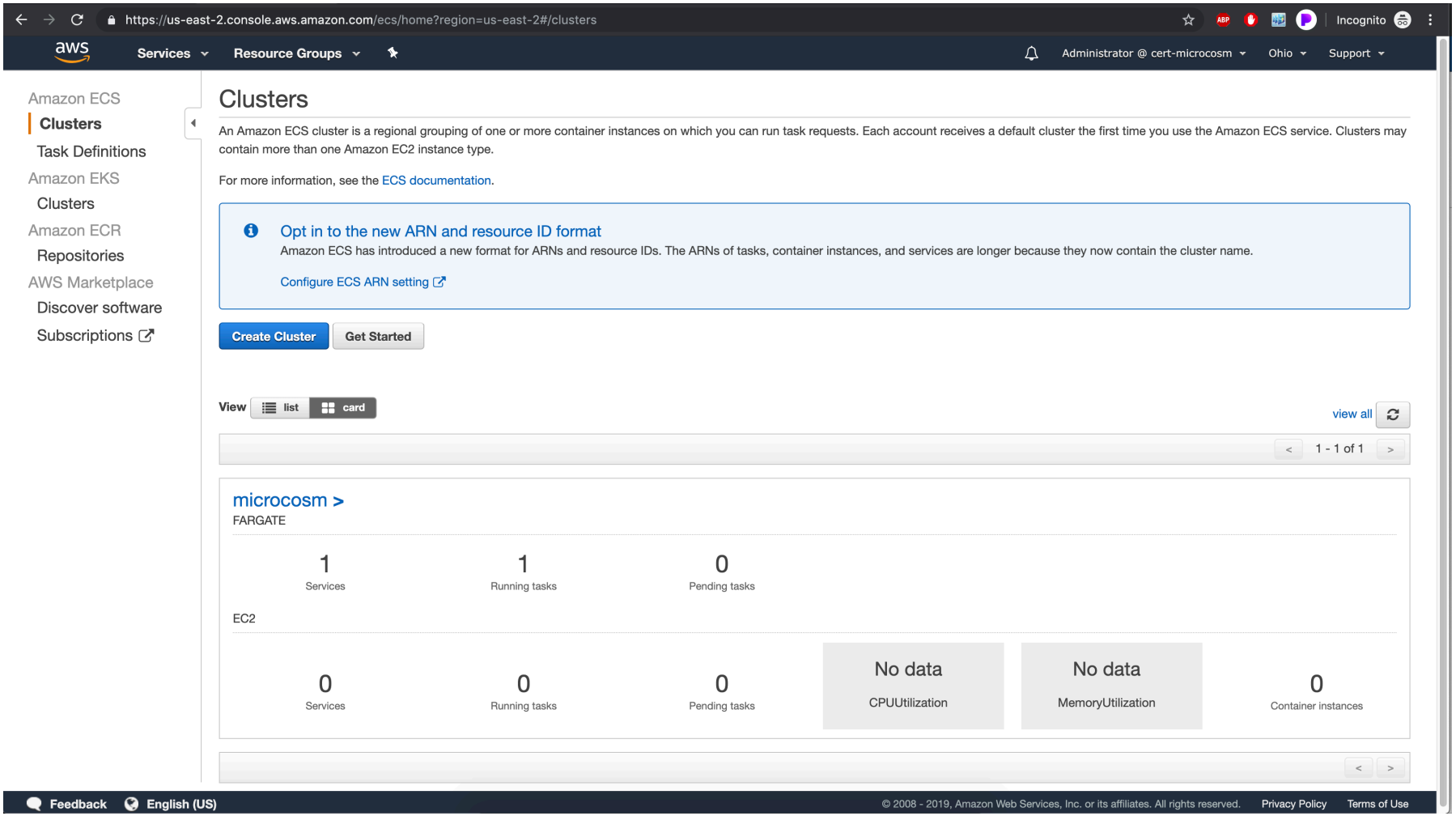

https://us-east-2.console.aws.amazon.com/ecs/home?region=us-east-2#/clusters/microcosm/services/jenkins-service/details
Incognito
aws
Services
Resource Groups
Administrator @ cert-microcosm
Ohio
Support
Amazon ECS
Clusters
Task Definitions
Amazon EKS
Clusters
Amazon ECR
Repositories
AWS Marketplace
Discover software
Subscriptions
Clusters > microcosm > Service: jenkins-service
Service : jenkins-service
Update
Delete
Cluster microcosm
Status ACTIVE
Task definition first-run-task-definition:3
Service type REPLICA
Launch type FARGATE
Platform version LATEST(1.3.0)
Service role AWSServiceRoleForECS
Desired count 1
Pending count 1
Running count 0
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-0530acfc15c751393
Allowed subnets subnet-0521ea54a64435dd4,subnet-0e50c4d05af5052a7
Security groups* sg-06f07d8d65ef6b2fd
Auto-assign public IP ENABLED
Feedback
English (US)
© 2008 - 2019, Amazon Web Services, Inc. or its affiliates. All rights reserved.
Privacy Policy
Terms of Use


https://us-east-2.console.aws.amazon.com/ecs/home?region=us-east-2#/clusters/microcosm/services/jenkins-service/deployments
Incognito
aws
Services
Resource Groups
Administrator @ cert-microcosm
Ohio
Support
Amazon ECS
Clusters
Task Definitions
Amazon EKS
Clusters
Amazon ECR
Repositories
AWS Marketplace
Discover software
Subscriptions
Clusters > microcosm > Service: jenkins-service
Service : jenkins-service
Update
Delete
Cluster microcosm
Status ACTIVE
Task definition first-run-task-definition:3
Service type REPLICA
Launch type FARGATE
Platform version LATEST(1.3.0)
Service role AWSServiceRoleForECS
Desired count 1
Pending count 0
Running count 1
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)
Filter in this page
1-1
Deployment Id
Status
Desired count
Pending count
Running count
Created time
Updated time
ecs-svc/9223370481860...
PRIMARY
1
0
1
2019-04-11 08:46:40 -0600
2019-04-11 08:47:53 -0600


https://us-east-2.console.aws.amazon.com/ecs/home?region=us-east-2#/clusters/microcosm/services/jenkins-service/tasks
Incognito
aws
Services
Resource Groups
Administrator @ cert-microcosm
Ohio
Support
Amazon ECS
Clusters
Task Definitions
Amazon EKS
Clusters
Amazon ECR
Repositories
AWS Marketplace
Discover software
Subscriptions
Clusters > microcosm > Service: jenkins-service
Service : jenkins-service
Update
Delete
Cluster microcosm
Status ACTIVE
Task definition first-run-task-definition:3
Service type REPLICA
Launch type FARGATE
Platform version LATEST(1.3.0)
Service role AWSServiceRoleForECS
Desired count 1
Pending count 0
Running count 1
Details
Tasks
Events
Auto Scaling
Deployments
Metrics
Tags
Logs
Last updated on April 11, 2019 8:48:45 AM (0m ago)
Task status: Running Stopped
Filter in this page
1-1
Page size 50
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
Feedback
English (US)
© 2008 - 2019, Amazon Web Services, Inc. or its affiliates. All rights reserved.
Privacy Policy
Terms of Use

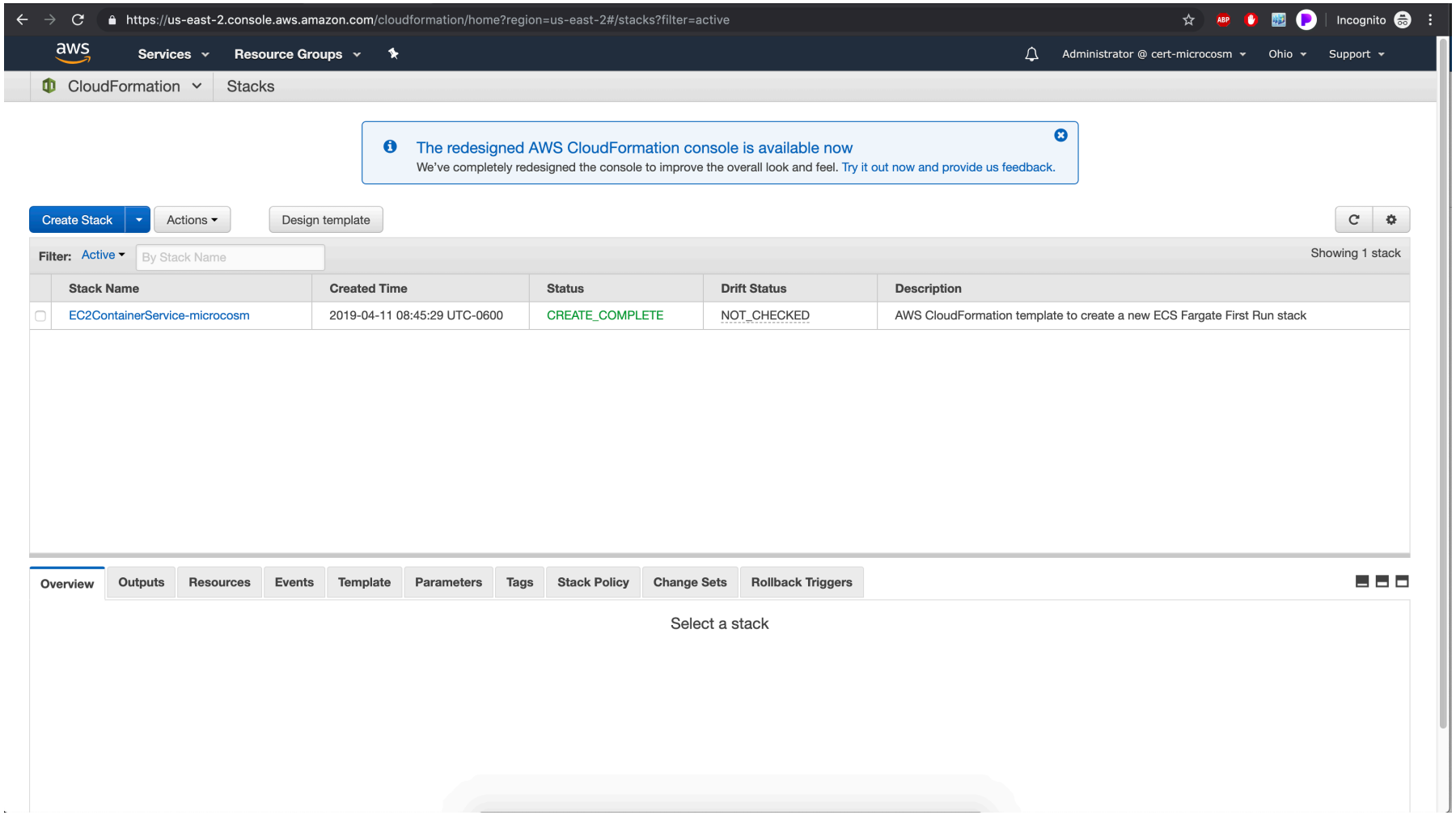

https://us-east-2.console.aws.amazon.com/ecs/home?region=us-east-2#/clusters/microcosm/tasks/a92d17a2-260c-4fdb-b880-cc48376ea9ec/details
Incognito
Amazon ECS
Clusters
Task Definitions
Amazon EKS
Clusters
Amazon ECR
Repositories
AWS Marketplace
Discover software
Subscriptions
Clusters > microcosm > Task: a92d17a2-260c-4fdb-b880-cc48376ea9ec
Task : a92d17a2-260c-4fdb-b880-cc48376ea9ec
Run more like this
Stop
Details
Tags
Logs
Cluster microcosm
Launch type FARGATE
Platform version 1.3.0
Task definition first-run-task-definition:3
Group service:jenkins-service
Task role None
Last status RUNNING
Desired status RUNNING
Created at 2019-04-11 08:46:48 -0600
Started at 2019-04-11 08:47:46 -0600
Network
Network mode awsvpc
ENI Id eni-03efb0fedd0d969f2
Subnet Id subnet-0e50c4d05af5052a7
Private IP 10.0.1.191
Public IP 18.191.252.84
Mac address 06:5e:3e:bd:77:c4
Containers
Last updated on April 11, 2019 8:48:52 AM (0m ago)
Name
Container Id
Status
Image
CPU Units
Hard/Soft memo...
Essential
jenkins
4ca0f74c-2b10-4f11-ad6f-fcab3ac20078
RUNNING
443007076818.dk...
0
--/--
true

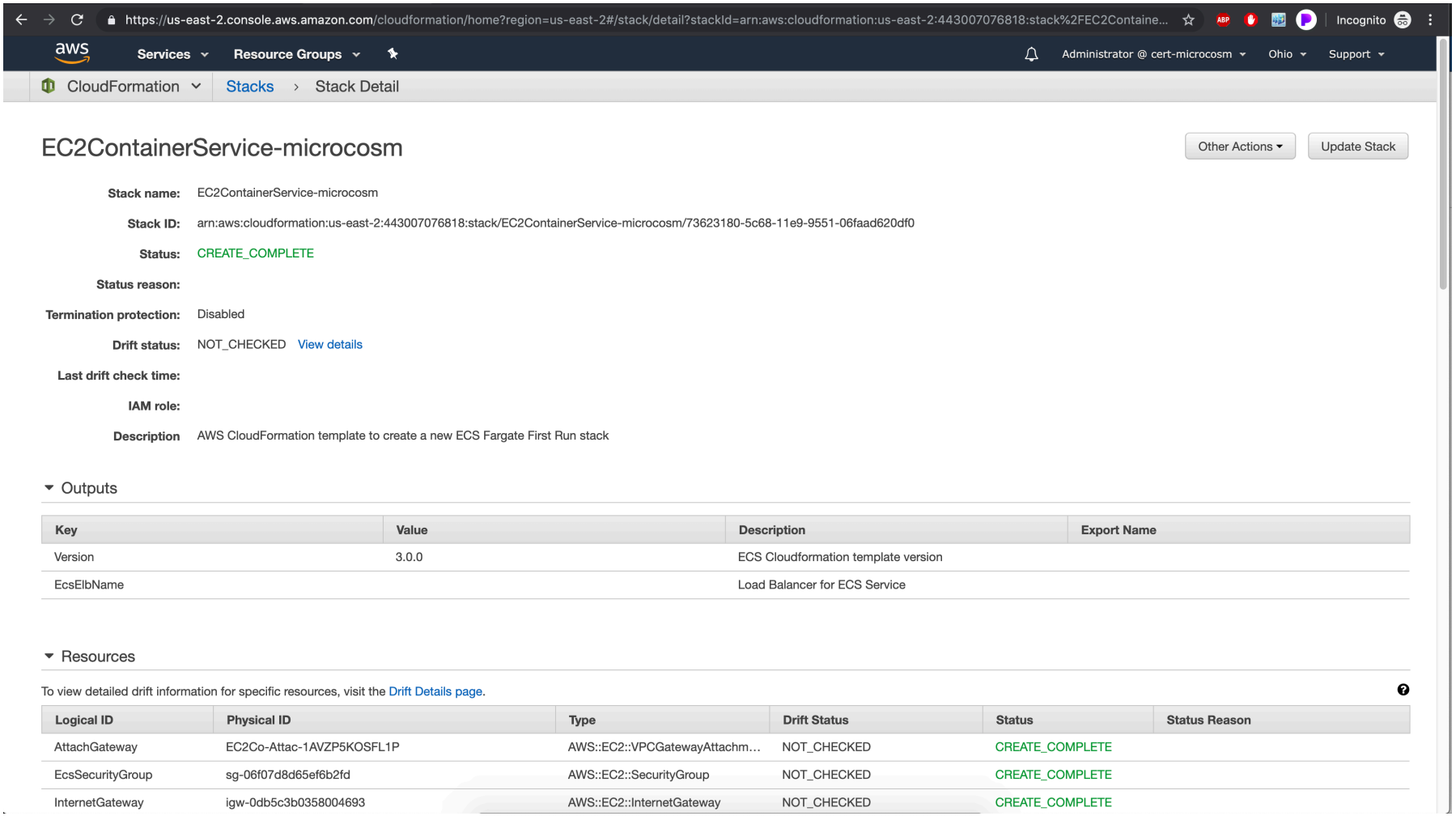

https://us-east-2.console.aws.amazon.com/ecs/home?region=us-east-2#/taskDefinitions
Incognito
aws
Services
Resource Groups
Administrator @ cert-microcosm
Ohio
Support
Amazon ECS
Clusters
Task Definitions
Amazon EKS
Clusters
Amazon ECR
Repositories
AWS Marketplace
Discover software
Subscriptions
Task 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
1-1
Page size 50
Task Definition
Latest revision status
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

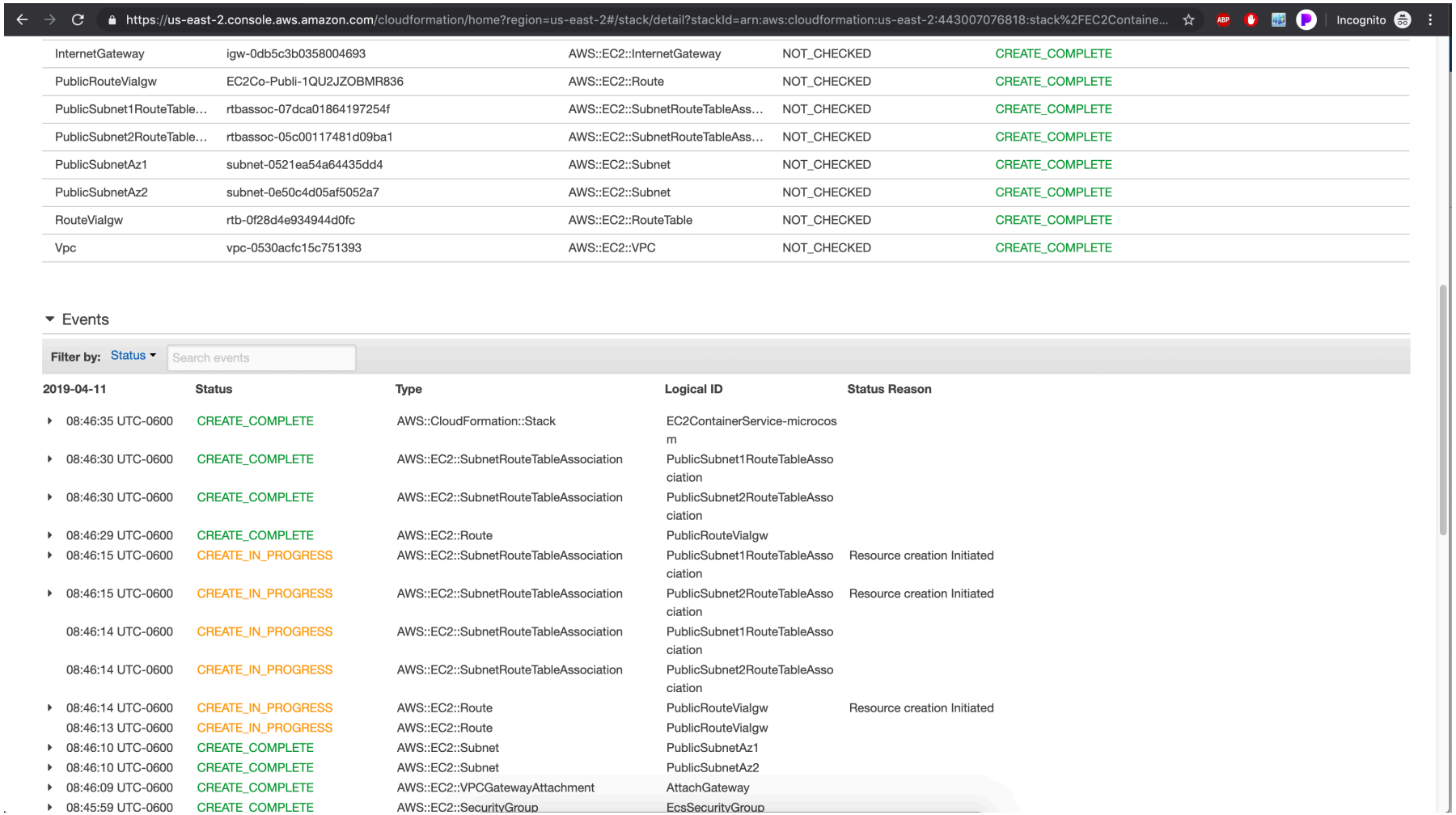


Everything Created By Get Started Wizard

Follow the screen shots for reference


https://us-east-2.console.aws.amazon.com/cloudformation/home?region=us-east-2#/stacks?filter=active
Incognito
aws
Services
Resource Groups
Administrator @ cert-microcosm
Ohio
Support
CloudFormation
Stacks
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 Stack
Actions
Design template
Filter: Active
By Stack Name
Showing 1 stack
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
Overview
Outputs
Resources
Events
Template
Parameters
Tags
Stack Policy
Change Sets
Rollback Triggers
Select a stack


https://us-east-2.console.aws.amazon.com/cloudformation/home?region=us-east-2#/stack/detail?stackId=arn:aws:cloudformation:us-east-2:443007076818:stack%2FEC2Containe...
Incognito
aws
Services
Resource Groups
Administrator @ cert-microcosm
Ohio
Support
CloudFormation
Stacks
Stack Detail
EC2ContainerService-microcosm
Other Actions
Update Stack
Stack name: EC2ContainerService-microcosm
Stack ID: arn:aws:cloudformation:us-east-2:443007076818:stack/EC2ContainerService-microcosm/73623180-5c68-11e9-9551-06faad620df0
Status: CREATE_COMPLETE
Status reason:
Termination protection: Disabled
Drift status: NOT_CHECKED View details
Last drift check time:
IAM role:
Description AWS CloudFormation template to create a new ECS Fargate First Run stack
Outputs
Key Value Description Export Name
Version 3.0.0 ECS Cloudformation template version
EcsElbName Load Balancer for ECS Service
Resources
To view detailed drift information for specific resources, visit the Drift Details page.
Logical ID Physical ID Type Drift Status Status Status Reason
AttachGateway EC2Co-Attac-1AVZP5KOSFL1P AWS::EC2::VPCGatewayAttachm... NOT_CHECKED CREATE_COMPLETE
EcsSecurityGroup sg-06f07d8d65ef6b2fd AWS::EC2::SecurityGroup NOT_CHECKED CREATE_COMPLETE
InternetGateway igw-0db5c3b0358004693 AWS::EC2::InternetGateway NOT_CHECKED CREATE_COMPLETE

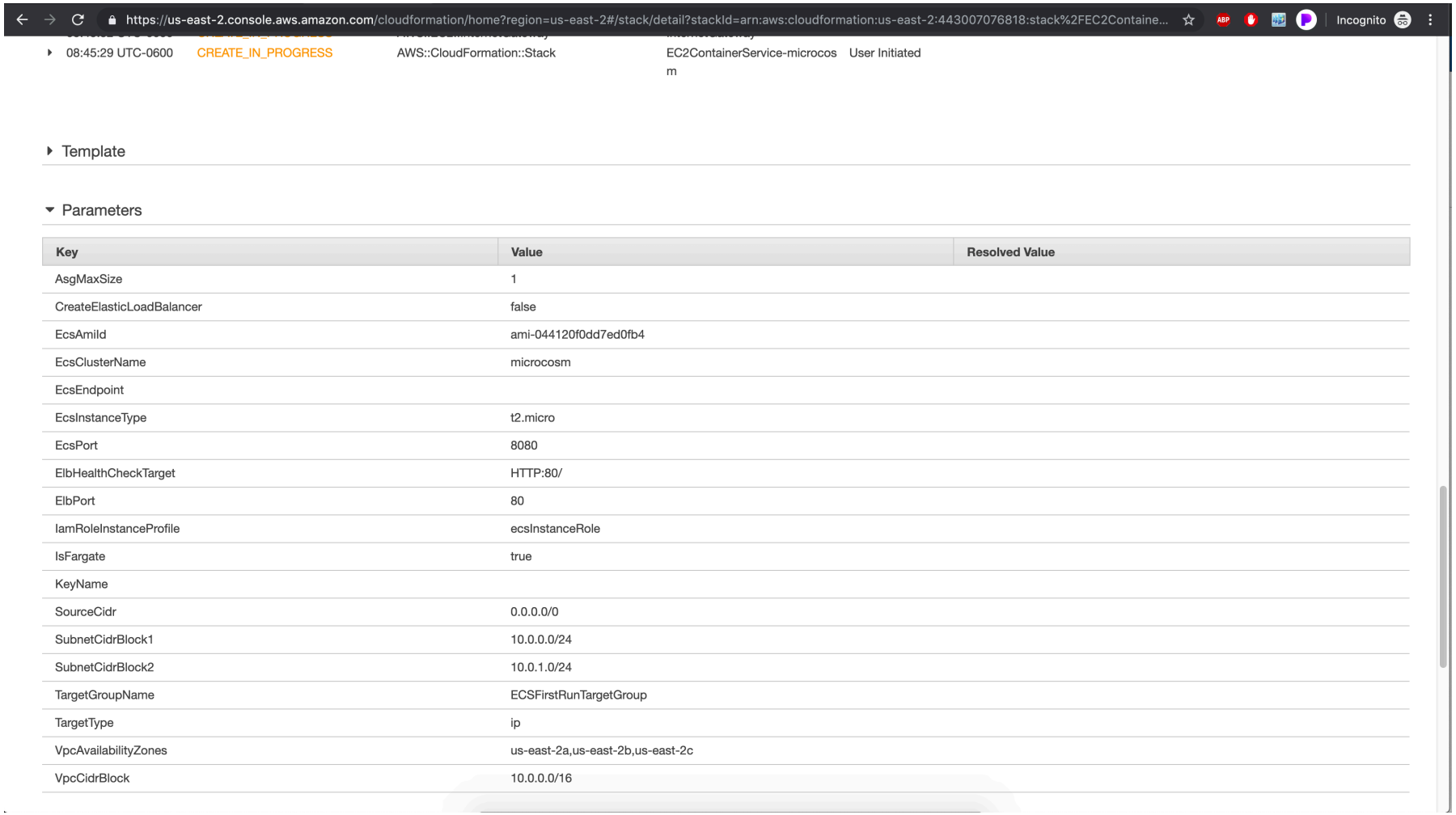


InternetGateway	igw-0db5c3b0358004693	AWS::EC2::InternetGateway	NOT_CHECKED	CREATE_COMPLETE
PublicRouteViaIgw	EC2Co-Publi-1QU2JZOBMR836	AWS::EC2::Route	NOT_CHECKED	CREATE_COMPLETE
PublicSubnet1RouteTable…	rtbassoc-07dca01864197254f	AWS::EC2::SubnetRouteTableAss…	NOT_CHECKED	CREATE_COMPLETE
PublicSubnet2RouteTable…	rtbassoc-05c00117481d09ba1	AWS::EC2::SubnetRouteTableAss…	NOT_CHECKED	CREATE_COMPLETE
PublicSubnetAz1	subnet-0521ea54a64435dd4	AWS::EC2::Subnet	NOT_CHECKED	CREATE_COMPLETE
PublicSubnetAz2	subnet-0e50c4d05af5052a7	AWS::EC2::Subnet	NOT_CHECKED	CREATE_COMPLETE
RouteViaIgw	rtb-0f28d4e934944d0fc	AWS::EC2::RouteTable	NOT_CHECKED	CREATE_COMPLETE
Vpc	vpc-0530acfc15c751393	AWS::EC2::VPC	NOT_CHECKED	CREATE_COMPLETE

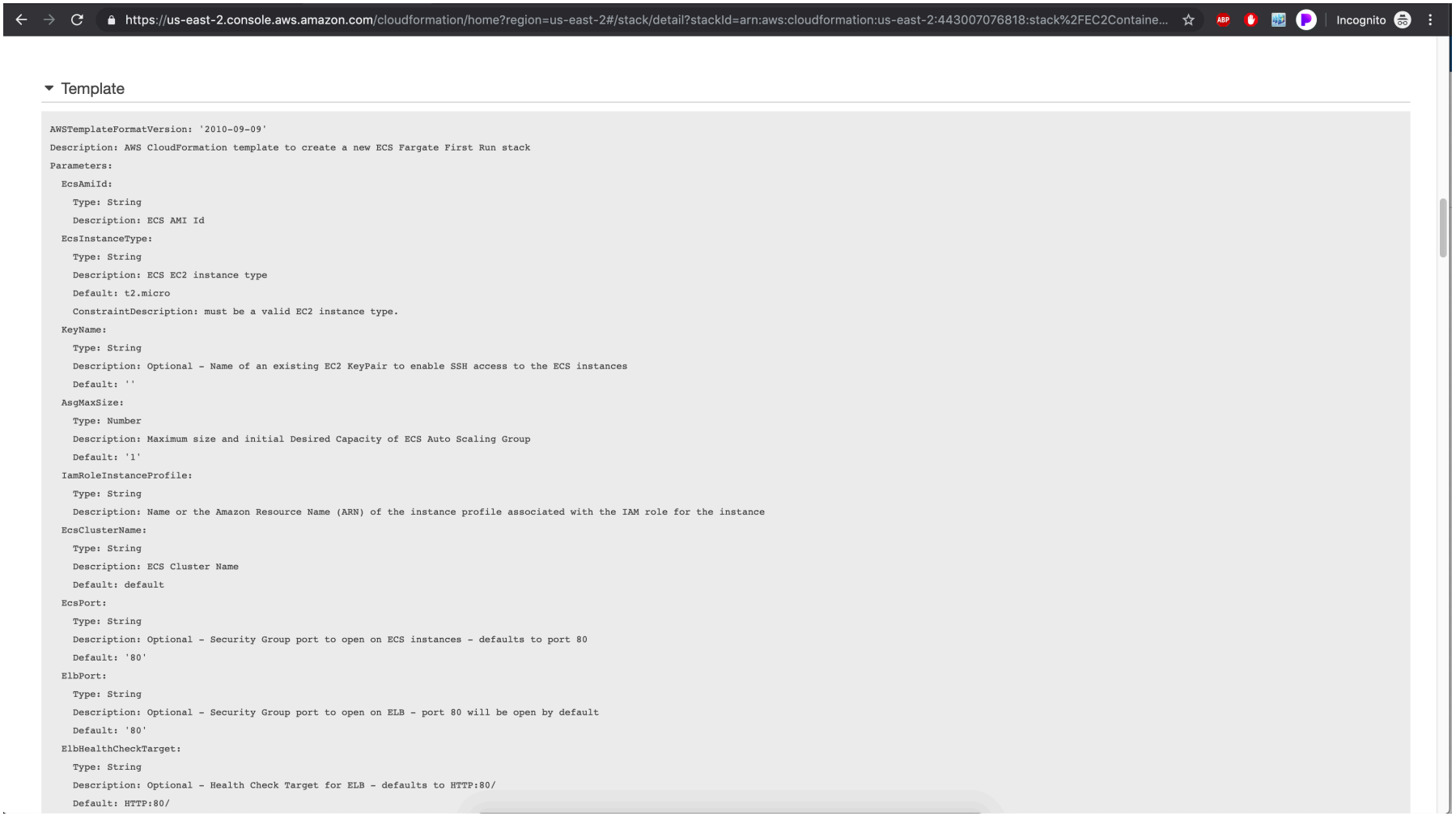
Events

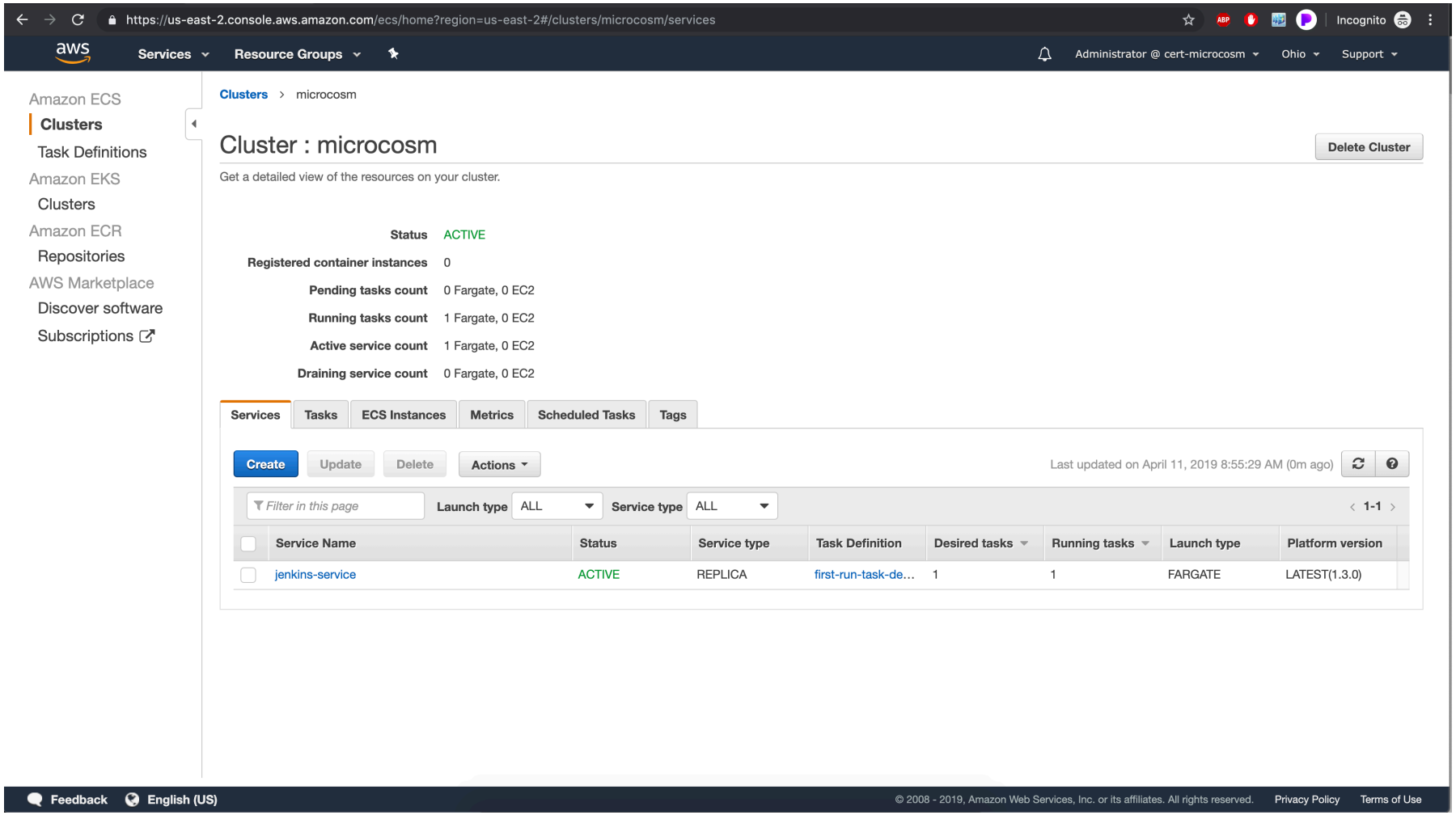
Filter by: Status Search events

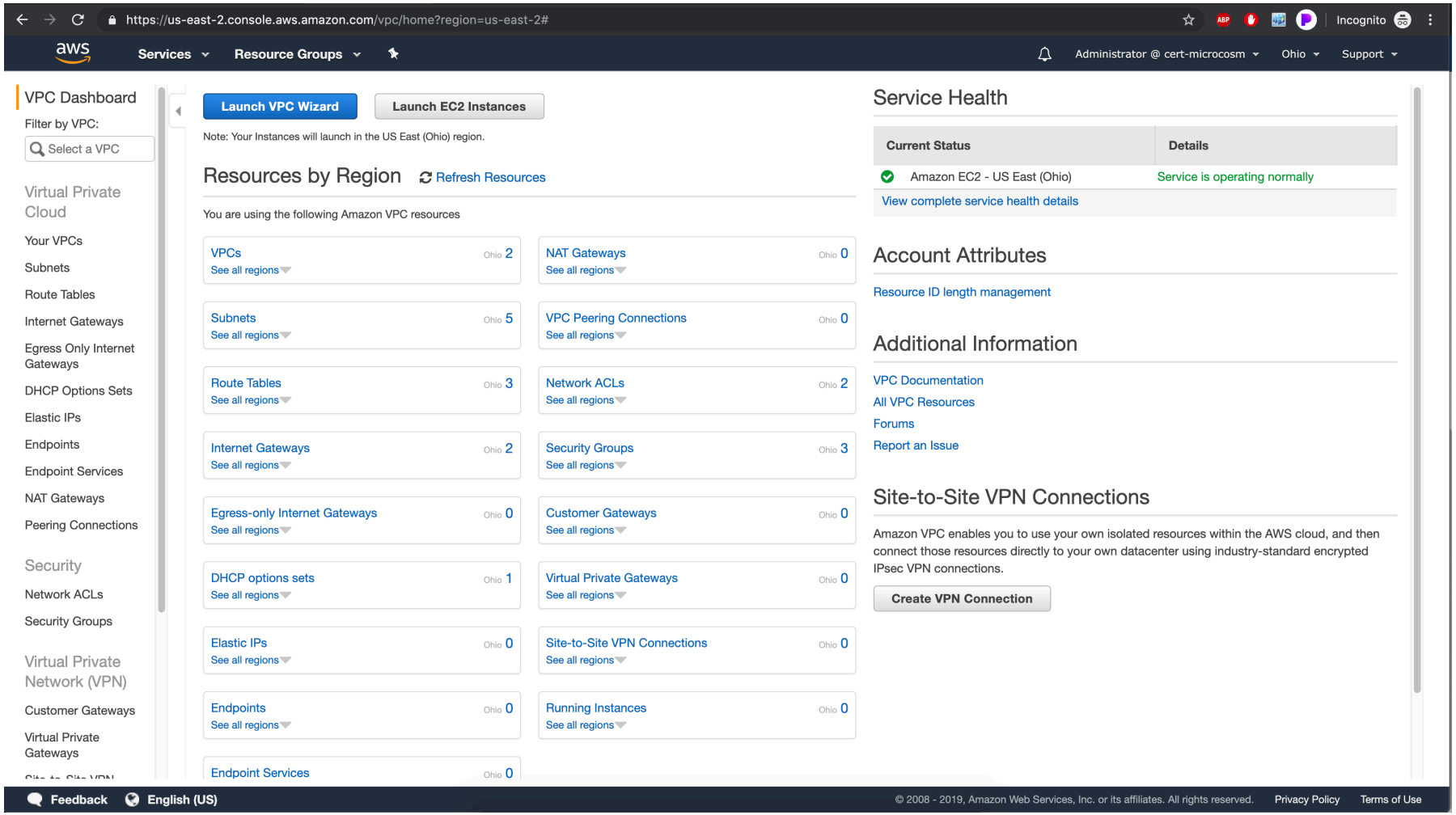
2019-04-11	Status	Type	Logical ID	Status Reason
08:46:35 UTC-0600	CREATE_COMPLETE	AWS::CloudFormation::Stack	EC2ContainerService-microcosm	
08:46:30 UTC-0600	CREATE_COMPLETE	AWS::EC2::SubnetRouteTableAssociation	PublicSubnet1RouteTableAssociation	
08:46:30 UTC-0600	CREATE_COMPLETE	AWS::EC2::SubnetRouteTableAssociation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

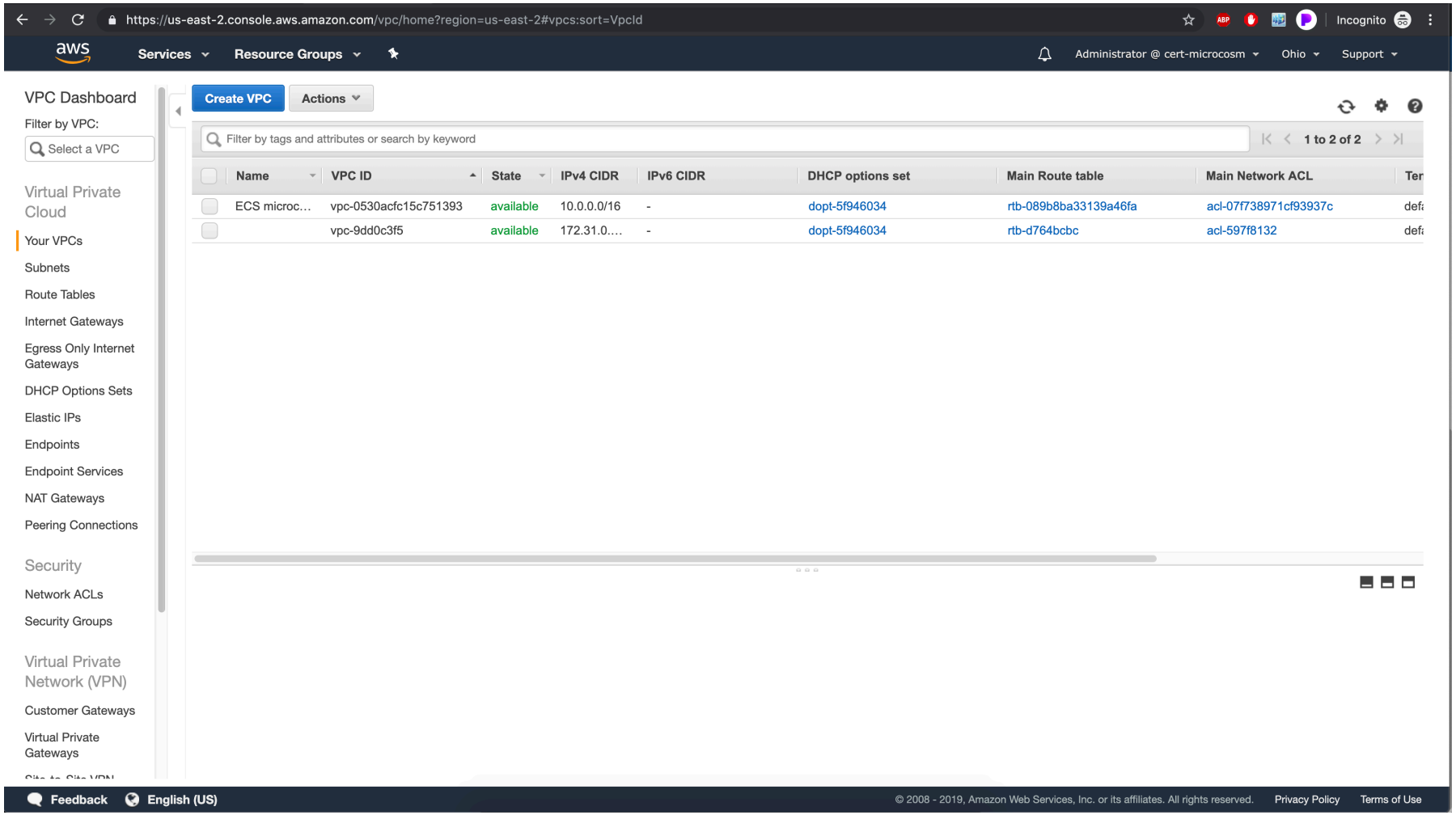


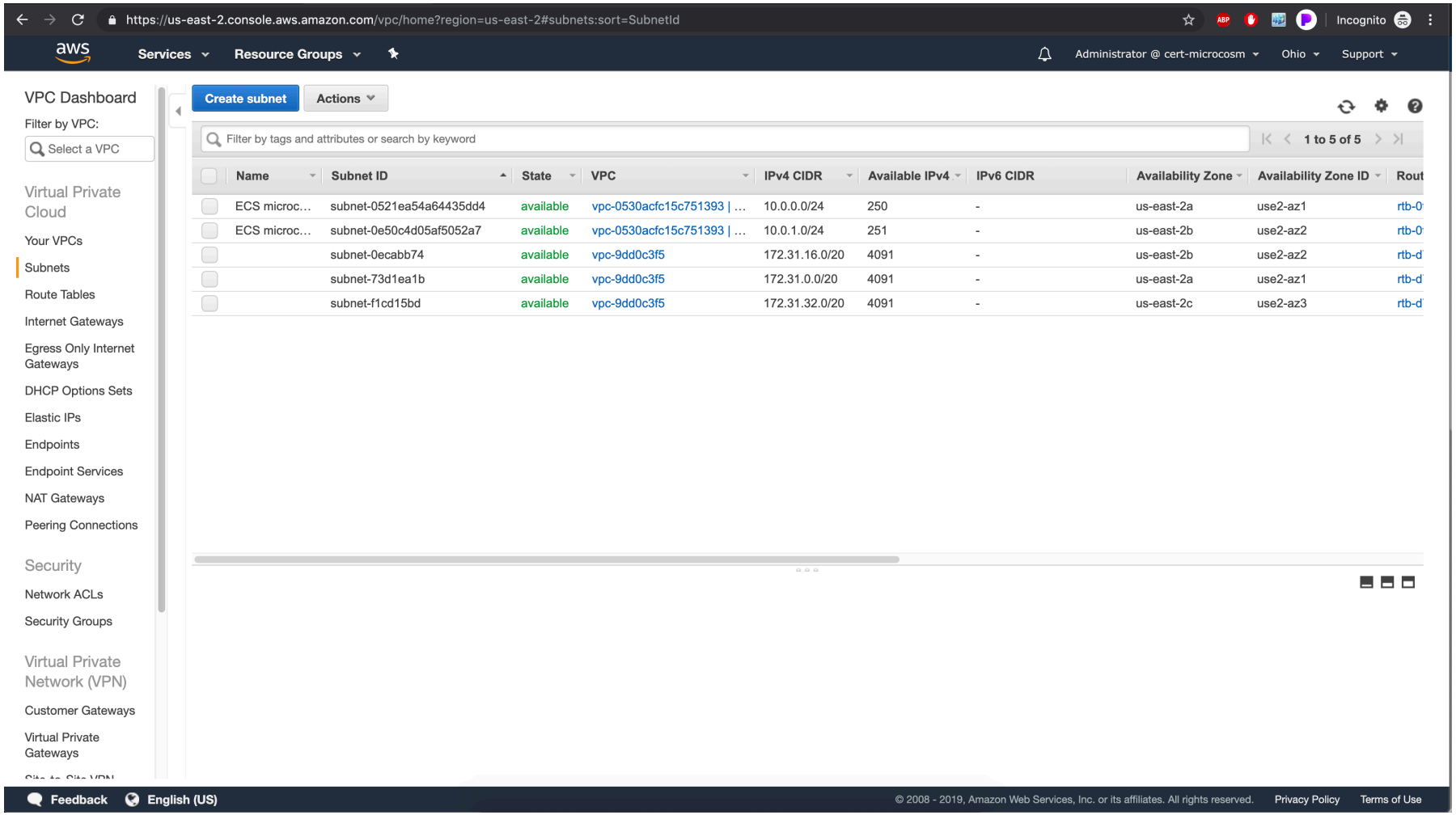
Key	Value	Resolved Value
AsgMaxSize	1	
CreateElasticLoadBalancer	false	
EcsAmiId	ami-044120f0dd7ed0fb4	
EcsClusterName	microcosm	
EcsEndpoint		
EcsInstanceType	t2.micro	
EcsPort	8080	
ElbHealthCheckTarget	HTTP:80/	
ElbPort	80	
IamRoleInstanceProfile	ecsInstanceRole	
IsFargate	true	
KeyName		
SourceCidr	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	

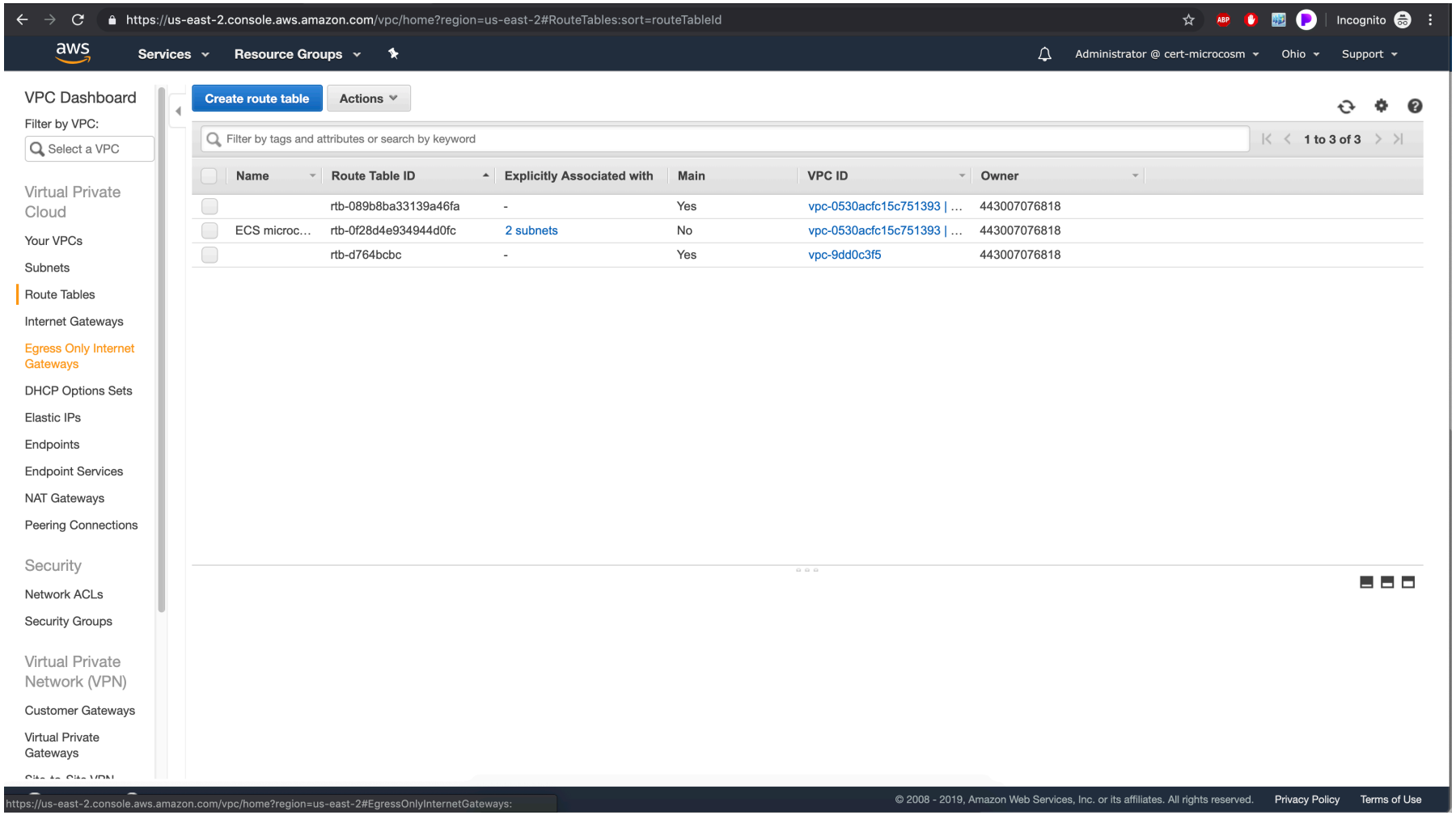

https://us-east-2.console.aws.amazon.com/cloudformation/home?region=us-east-2#/stack/detail?stackId=arn:aws:cloudformation:us-east-2:443007076818:stack%2FEC2Containe...
Incognito
Template
AWSTemplateFormatVersion: '2010-09-09'
Description: AWS CloudFormation template to create a new ECS Fargate First 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/

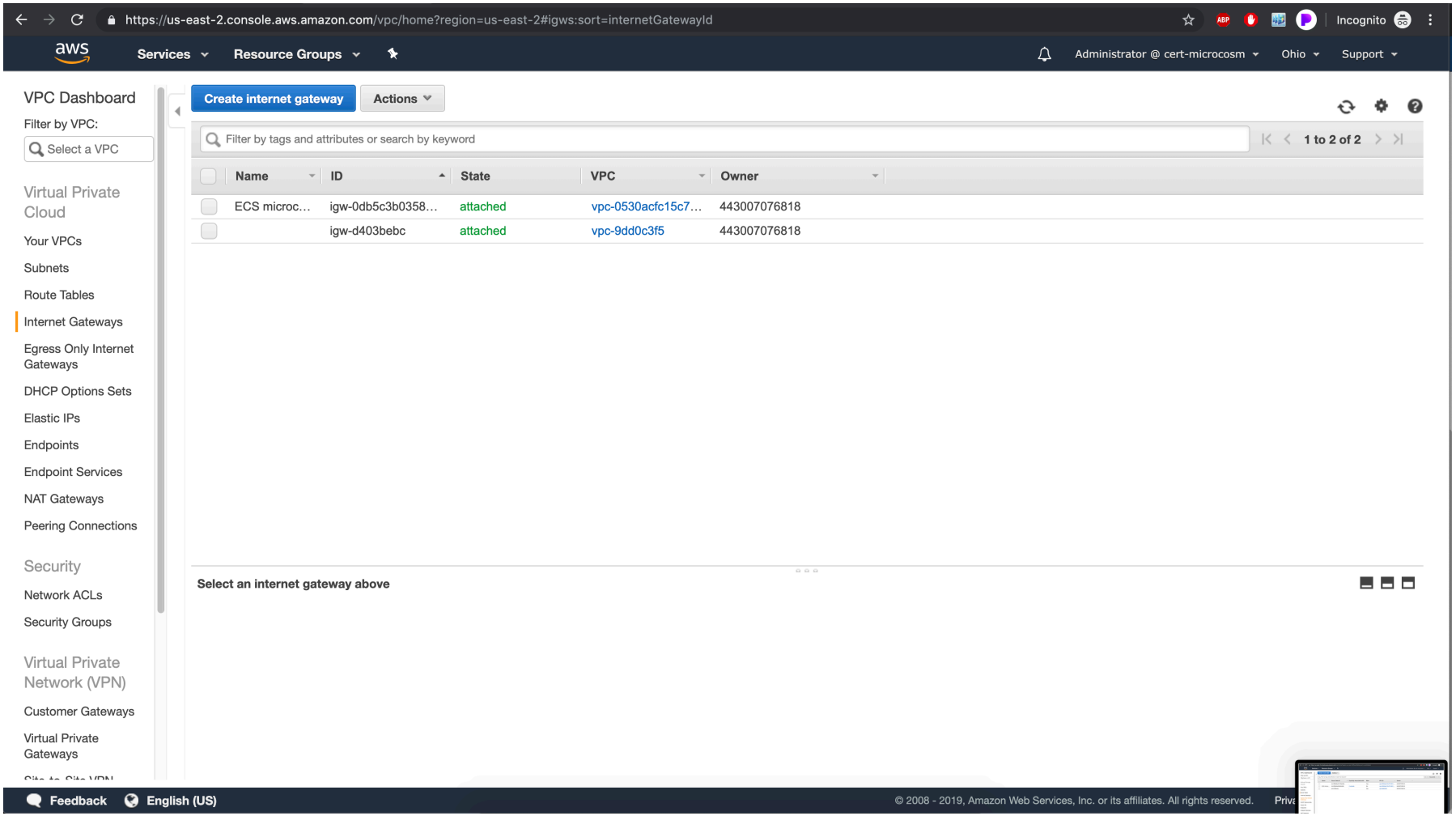

https://us-east-2.console.aws.amazon.com/ecs/home?region=us-east-2#/clusters/microcosm/services
Incognito
aws
Services
Resource Groups
Administrator @ cert-microcosm
Ohio
Support
Amazon ECS
Clusters
Task Definitions
Amazon EKS
Clusters
Amazon ECR
Repositories
AWS Marketplace
Discover software
Subscriptions
Clusters > microcosm
Cluster : microcosm
Delete Cluster
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 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
1-1
Service Name
Status
Service type
Task Definition
Desired tasks
Running tasks
Launch type
Platform version
jenkins-service
ACTIVE
REPLICA
first-run-task-de...
1
1
FARGATE
LATEST(1.3.0)
Feedback
English (US)
© 2008 - 2019, Amazon Web Services, Inc. or its affiliates. All rights reserved.
Privacy Policy
Terms of Use

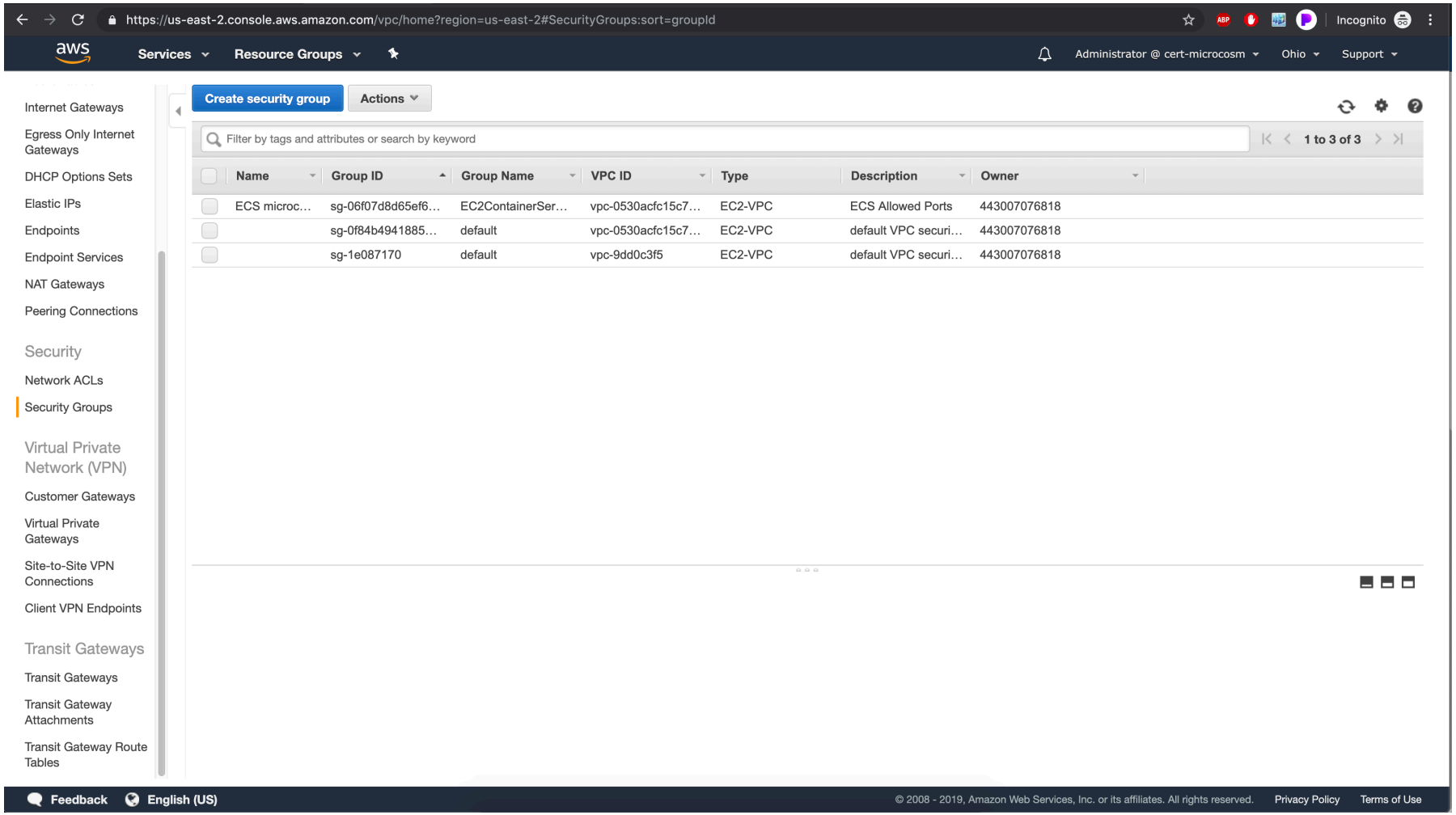

https://us-east-2.console.aws.amazon.com/vpc/home?region=us-east-2#
Incognito
aws
Services
Resource Groups
Administrator @ cert-microcosm
Ohio
Support
VPC Dashboard
Filter by VPC:
Select a VPC
Virtual Private Cloud
Your VPCs
Subnets
Route Tables
Internet Gateways
Egress Only Internet Gateways
DHCP Options Sets
Elastic IPs
Endpoints
Endpoint Services
NAT Gateways
Peering Connections
Security
Network ACLs
Security Groups
Virtual Private Network (VPN)
Customer Gateways
Virtual Private Gateways
Launch VPC Wizard
Launch EC2 Instances
Note: Your Instances will launch in the US East (Ohio) region.
Resources by Region
Refresh Resources
You are using the following Amazon VPC resources
VPCs Ohio 2 See all regions
NAT Gateways Ohio 0 See all regions
Subnets Ohio 5 See all regions
VPC Peering Connections Ohio 0 See all regions
Route Tables Ohio 3 See all regions
Network ACLs Ohio 2 See all regions
Internet Gateways Ohio 2 See all regions
Security Groups Ohio 3 See all regions
Egress-only Internet Gateways Ohio 0 See all regions
Customer Gateways Ohio 0 See all regions
DHCP options sets Ohio 1 See all regions
Virtual Private Gateways Ohio 0 See all regions
Elastic IPs Ohio 0 See all regions
Site-to-Site VPN Connections Ohio 0 See all regions
Endpoints Ohio 0 See all regions
Running Instances Ohio 0 See all regions
Endpoint Services Ohio 0
Service Health
Current Status
Details
Amazon EC2 - US East (Ohio)
Service is operating normally
View complete service health details
Account Attributes
Resource ID length management
Additional Information
VPC Documentation
All VPC Resources
Forums
Report an Issue
Site-to-Site VPN Connections
Amazon VPC enables you to use your own isolated resources within the AWS cloud, and then connect those resources directly to your own datacenter using industry-standard encrypted IPsec VPN connections.
Create VPN Connection
Feedback
English (US)
© 2008 - 2019, Amazon Web Services, Inc. or its affiliates. All rights reserved.
Privacy Policy
Terms of Use

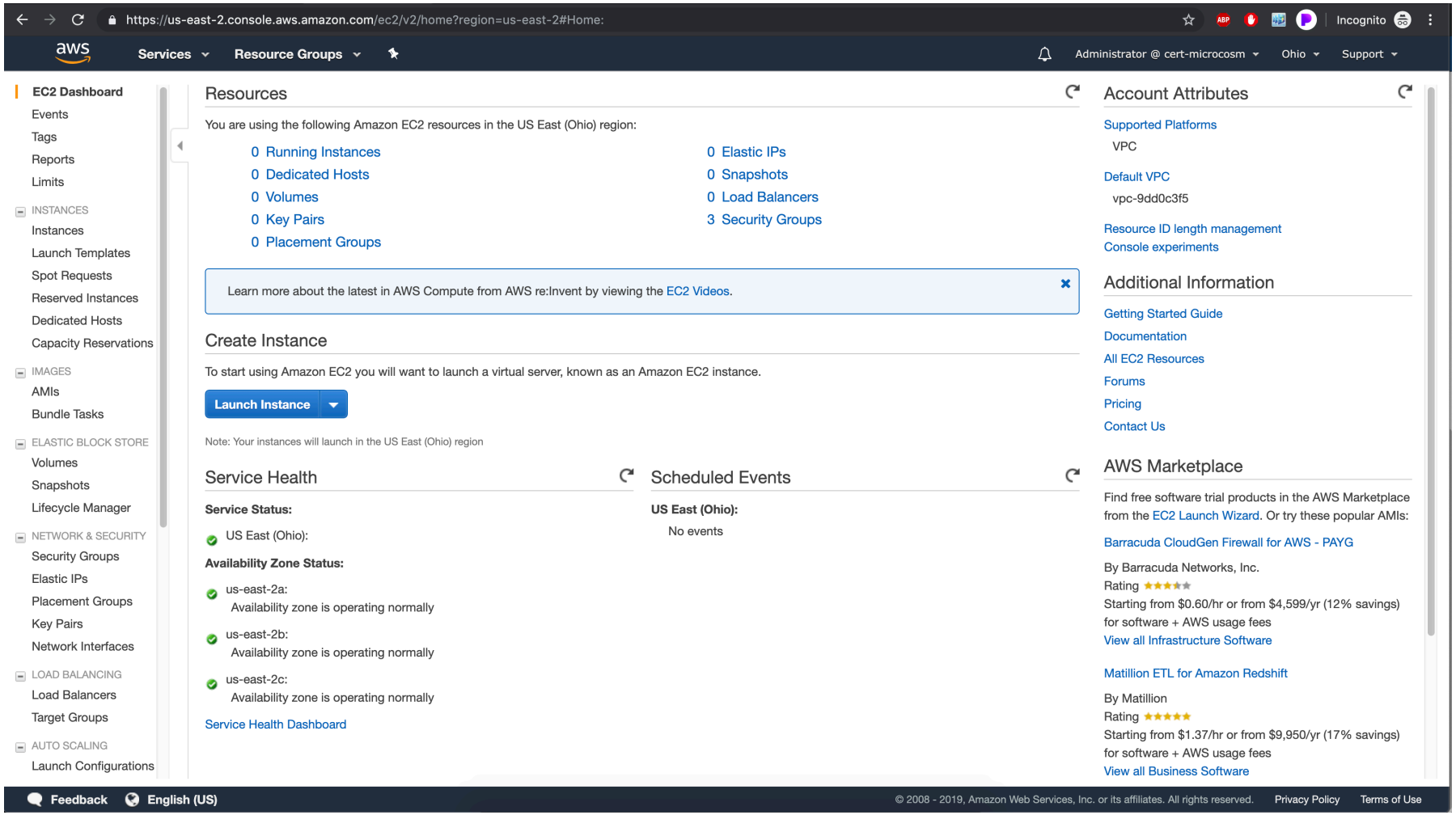

https://us-east-2.console.aws.amazon.com/vpc/home?region=us-east-2#vpcs:sort=VpcId
Incognito
aws
Services
Resource Groups
Administrator @ cert-microcosm
Ohio
Support
VPC Dashboard
Filter by VPC:
Select a VPC
Virtual Private Cloud
Your VPCs
Subnets
Route Tables
Internet Gateways
Egress Only Internet Gateways
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
1 to 2 of 2
Name
VPC ID
State
IPv4 CIDR
IPv6 CIDR
DHCP options set
Main Route table
Main Network ACL
ECS microc...
vpc-0530acfc15c751393
available
10.0.0.0/16
-
dopt-5f946034
rtb-089b8ba33139a46fa
acl-07f738971cf93937c
vpc-9dd0c3f5
available
172.31.0....
-
dopt-5f946034
rtb-d764bcbc
acl-597f8132
Feedback
English (US)
© 2008 - 2019, Amazon Web Services, Inc. or its affiliates. All rights reserved.
Privacy Policy
Terms of Use

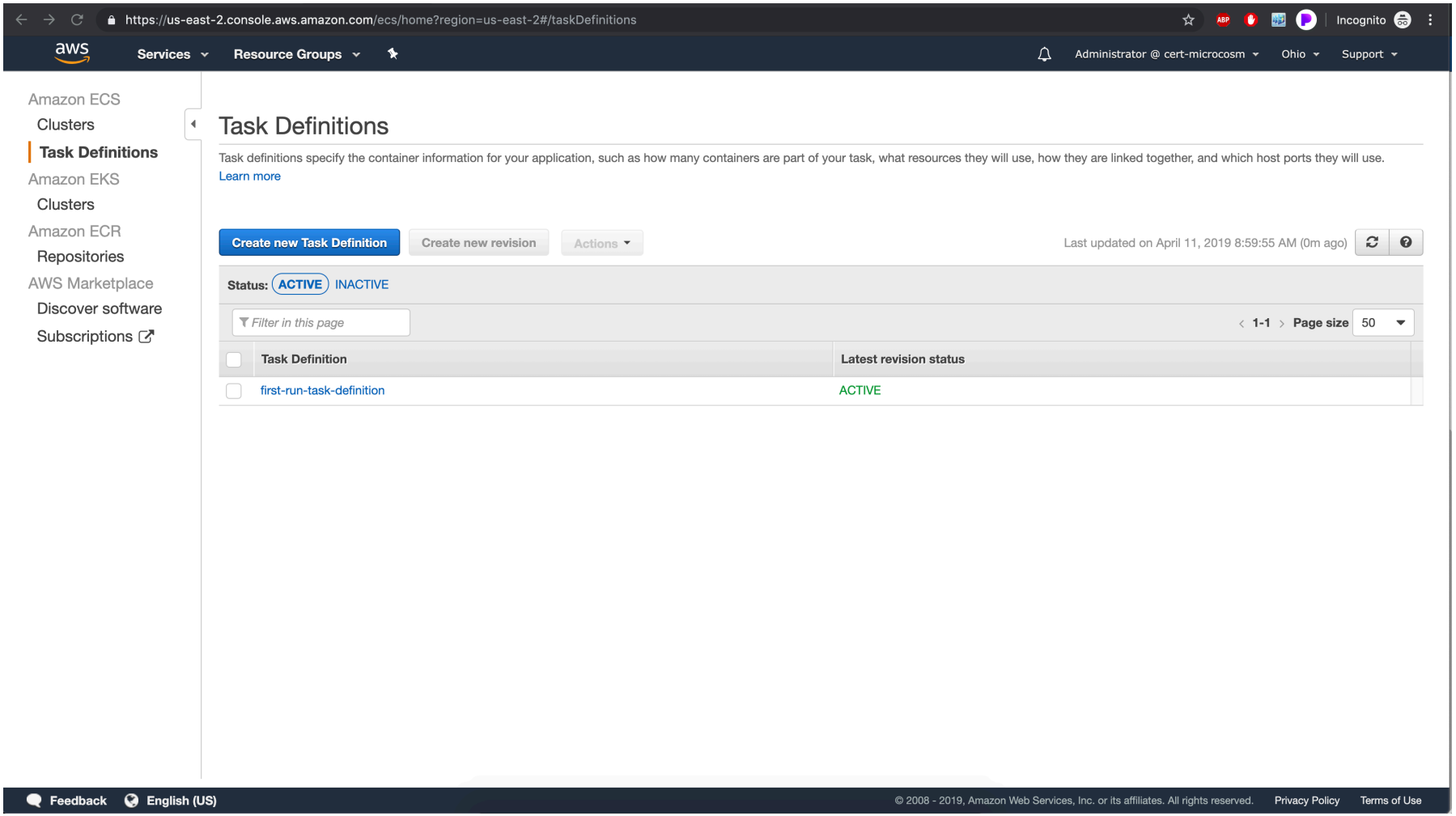

https://us-east-2.console.aws.amazon.com/vpc/home?region=us-east-2#subnets:sort=SubnetId
Incognito
aws
Services
Resource Groups
Administrator @ cert-microcosm
Ohio
Support
VPC Dashboard
Filter by VPC:
Select a VPC
Virtual Private Cloud
Your VPCs
Subnets
Route Tables
Internet Gateways
Egress Only Internet 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
1 to 5 of 5
Name
Subnet ID
State
VPC
IPv4 CIDR
Available IPv4
IPv6 CIDR
Availability Zone
Availability Zone ID
ECS microc... subnet-0521ea54a64435dd4 available vpc-0530acfc15c751393 | ... 10.0.0.0/24 250 - us-east-2a use2-az1
ECS microc... subnet-0e50c4d05af5052a7 available vpc-0530acfc15c751393 | ... 10.0.1.0/24 251 - us-east-2b use2-az2
subnet-0ecabb74 available vpc-9dd0c3f5 172.31.16.0/20 4091 - us-east-2b use2-az2
subnet-73d1ea1b available vpc-9dd0c3f5 172.31.0.0/20 4091 - us-east-2a use2-az1
subnet-f1cd15bd available vpc-9dd0c3f5 172.31.32.0/20 4091 - us-east-2c use2-az3
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
Incognito
aws
Services
Resource Groups
Administrator @ cert-microcosm
Ohio
Support
VPC Dashboard
Filter by VPC:
Select a VPC
Virtual Private Cloud
Your VPCs
Subnets
Route Tables
Internet Gateways
Egress Only Internet Gateways
DHCP Options Sets
Elastic IPs
Endpoints
Endpoint Services
NAT Gateways
Peering Connections
Security
Network ACLs
Security Groups
Virtual Private Network (VPN)
Customer Gateways
Virtual Private Gateways
Create route table
Actions
Filter by tags and attributes or search by keyword
1 to 3 of 3
Name
Route Table ID
Explicitly Associated with
Main
VPC ID
Owner
rtb-089b8ba33139a46fa - Yes vpc-0530acfc15c751393 | ... 443007076818
ECS microc... rtb-0f28d4e934944d0fc 2 subnets No vpc-0530acfc15c751393 | ... 443007076818
rtb-d764bcbc - Yes vpc-9dd0c3f5 443007076818
https://us-east-2.console.aws.amazon.com/vpc/home?region=us-east-2#EgressOnlyInternetGateways:
© 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#igws:sort=internetGatewayId
Incognito
aws
Services
Resource Groups
Administrator @ cert-microcosm
Ohio
Support
VPC Dashboard
Filter by VPC:
Select a VPC
Virtual Private Cloud
Your VPCs
Subnets
Route Tables
Internet Gateways
Egress Only Internet Gateways
DHCP Options Sets
Elastic IPs
Endpoints
Endpoint Services
NAT Gateways
Peering Connections
Security
Network ACLs
Security Groups
Virtual Private Network (VPN)
Customer Gateways
Virtual Private Gateways
Create internet gateway
Actions
Filter by tags and attributes or search by keyword
1 to 2 of 2
Name
ID
State
VPC
Owner
ECS microc…
igw-0db5c3b0358…
attached
vpc-0530acfc15c7…
443007076818
igw-d403bebc
attached
vpc-9dd0c3f5
443007076818
Select an internet gateway above
Feedback
English (US)
© 2008 - 2019, Amazon Web Services, Inc. or its affiliates. All rights reserved.


https://us-east-2.console.aws.amazon.com/vpc/home?region=us-east-2#SecurityGroups:sort=groupId
Incognito
aws
Services
Resource Groups
Administrator @ cert-microcosm
Ohio
Support
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
Site-to-Site VPN Connections
Client VPN Endpoints
Transit Gateways
Transit Gateways
Transit Gateway Attachments
Transit Gateway Route Tables
Create security group
Actions
Filter by tags and attributes or search by keyword
1 to 3 of 3
Name
Group ID
Group Name
VPC ID
Type
Description
Owner
ECS microc…
sg-06f07d8d65ef6…
EC2ContainerSer…
vpc-0530acfc15c7…
EC2-VPC
ECS Allowed Ports
443007076818
sg-0f84b4941885…
default
vpc-0530acfc15c7…
EC2-VPC
default VPC securi…
443007076818
sg-1e087170
default
vpc-9dd0c3f5
EC2-VPC
default VPC securi…
443007076818
Feedback
English (US)
© 2008 - 2019, Amazon Web Services, Inc. or its affiliates. All rights reserved.
Privacy Policy
Terms of Use


https://us-east-2.console.aws.amazon.com/ec2/v2/home?region=us-east-2#Home:
Incognito
aws
Services
Resource Groups
Administrator @ cert-microcosm
Ohio
Support
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
AUTO SCALING
Launch Configurations
Resources
You are using the following Amazon EC2 resources in the US East (Ohio) region:
0 Running Instances
0 Dedicated Hosts
0 Volumes
0 Key Pairs
0 Placement Groups
0 Elastic IPs
0 Snapshots
0 Load Balancers
3 Security Groups
Learn more about the latest in AWS Compute from AWS re:Invent by viewing the EC2 Videos.
Create Instance
To start using Amazon EC2 you will want to launch a virtual server, known as an Amazon EC2 instance.
Launch Instance
Note: Your instances will launch in the US East (Ohio) region
Service Health
Service Status:
US East (Ohio):
Availability Zone Status:
us-east-2a:
Availability zone is operating normally
us-east-2b:
Availability zone is operating normally
us-east-2c:
Availability zone is operating normally
Service Health Dashboard
Scheduled Events
US East (Ohio):
No events
Account Attributes
Supported Platforms
VPC
Default VPC
vpc-9dd0c3f5
Resource ID length management
Console experiments
Additional Information
Getting Started Guide
Documentation
All EC2 Resources
Forums
Pricing
Contact Us
AWS Marketplace
Find free software trial products in the AWS Marketplace from the EC2 Launch Wizard. Or try these popular AMIs:
Barracuda CloudGen Firewall for AWS - PAYG
By Barracuda Networks, Inc.
Rating
Starting from $0.60/hr or from $4,599/yr (12% savings) for software + AWS usage fees
View all Infrastructure Software
Matillion ETL for Amazon Redshift
By Matillion
Rating
Starting from $1.37/hr or from $9,950/yr (17% savings) for software + AWS usage fees
View all Business Software
Feedback
English (US)
© 2008 - 2019, Amazon Web Services, Inc. or its affiliates. All rights reserved.
Privacy Policy
Terms of Use



Deleting an ECS Service and Associated Tasks

- Navigate to the ECS Console and select your cluster

Clusters

An Amazon ECS cluster is a regional grouping of one or more container instances on wh

For more information, see the ECS documentation.

Opt in to the new ARN and resource ID format

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

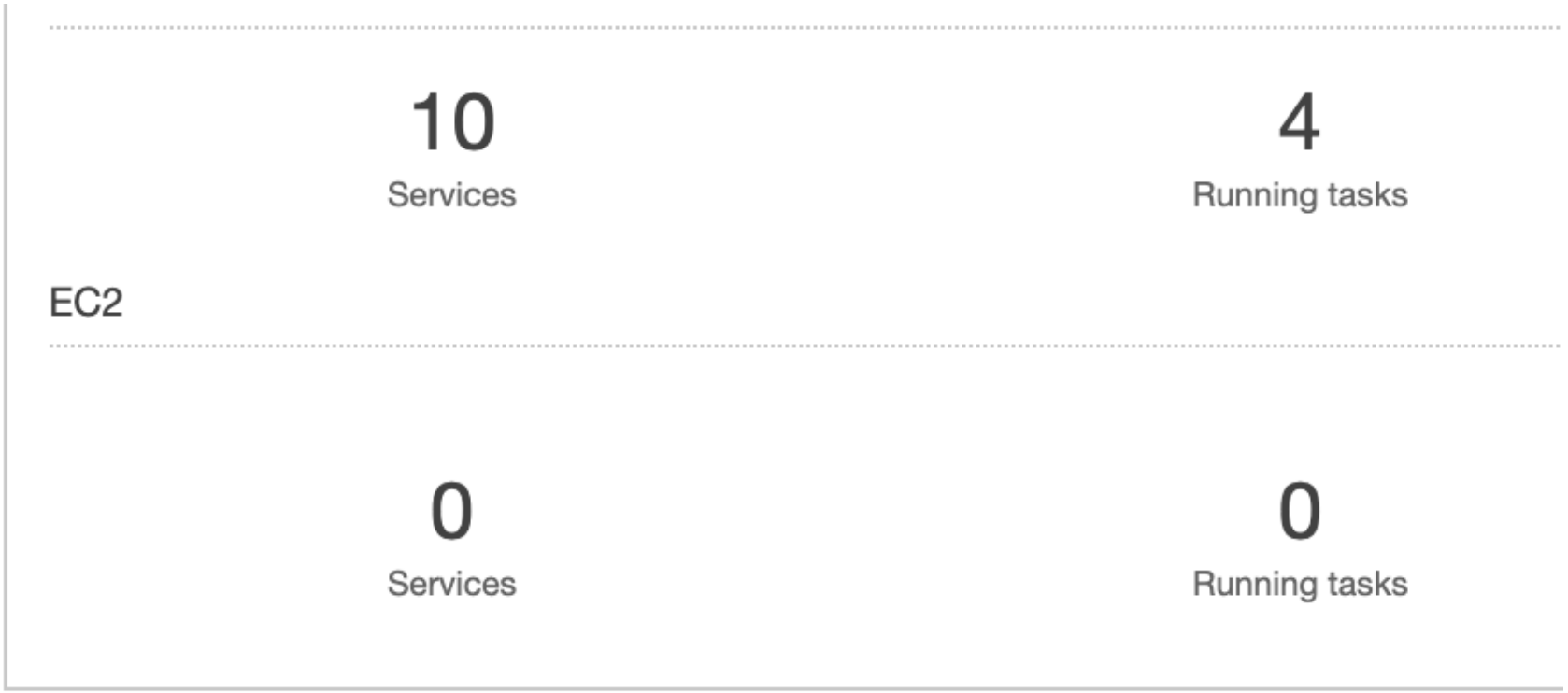
Configure ECS ARN setting

Create Cluster Get Started

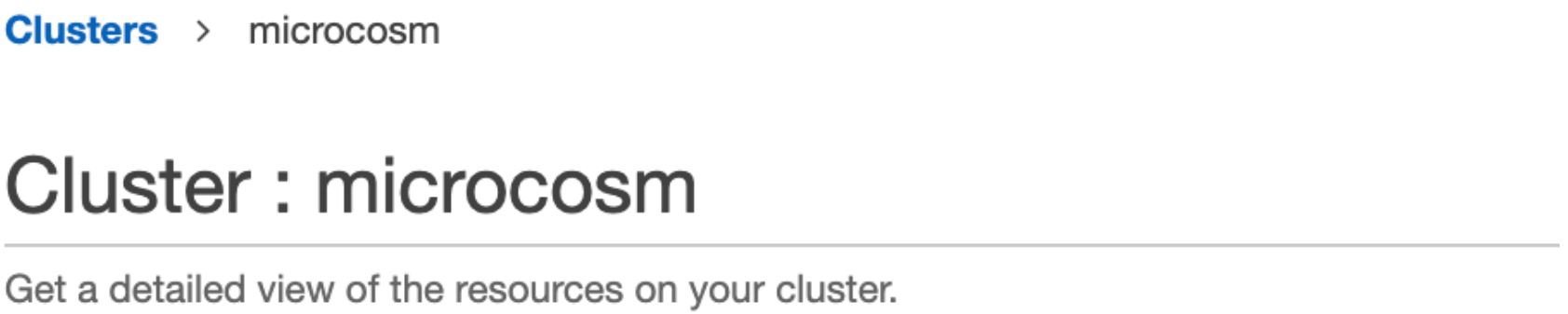
View list card

microcosm >

FARGATE



- Scroll down and select your service



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

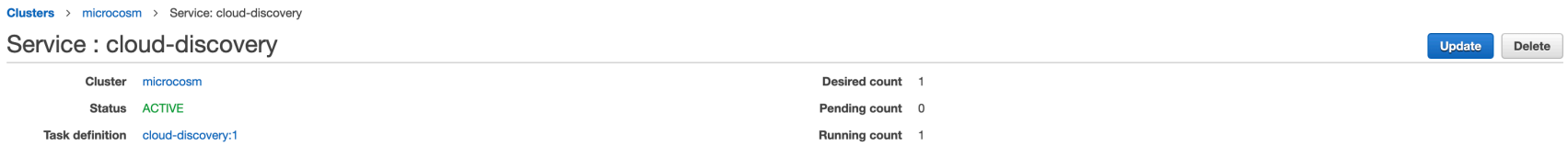
Services | **Tasks** | **ECS Instances** | **Metrics** | **Scheduled Tasks** | **Tags**

Create | **Update** | **Delete** | **Actions**

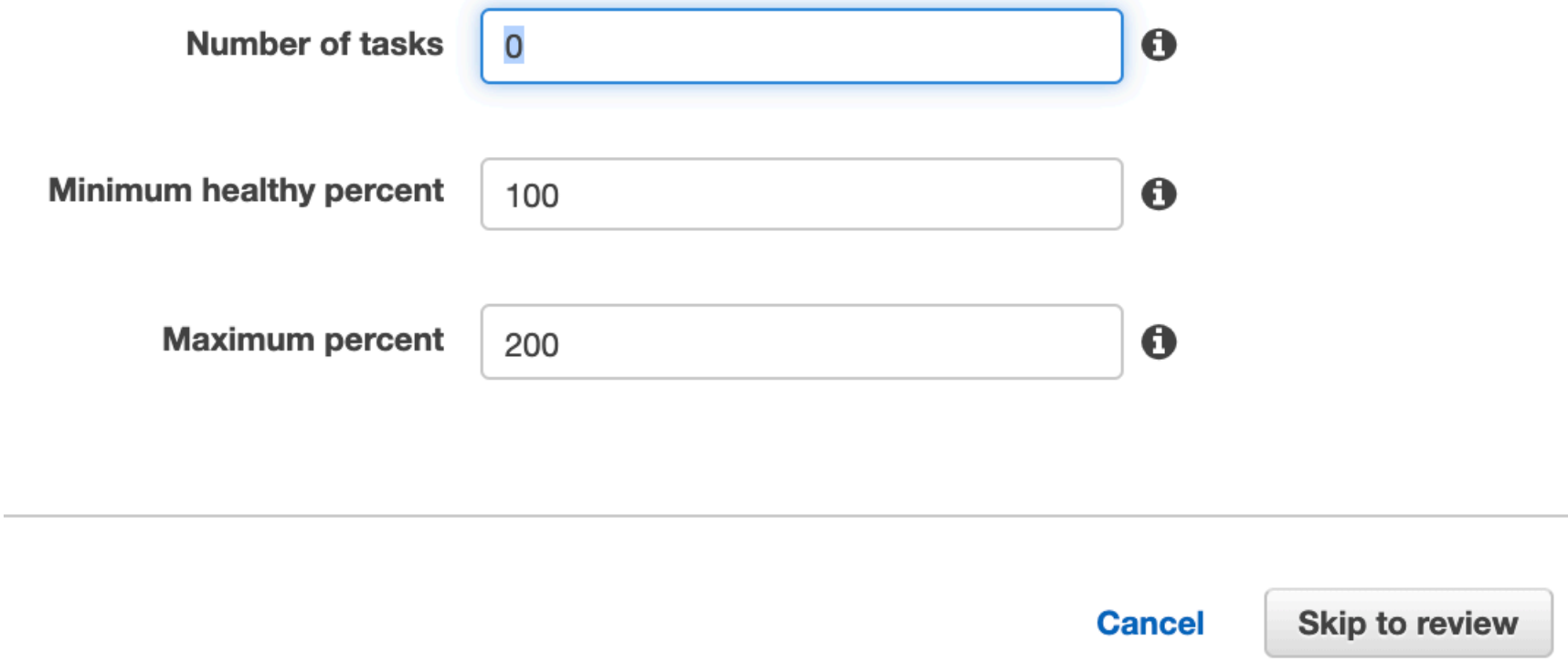
Filter in this page **Launch type** ALL **Service type** AL

	Service Name	Status
☐	sonarqube	ACTIVE
☐	gitlab	ACTIVE
☐	hubot	ACTIVE

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

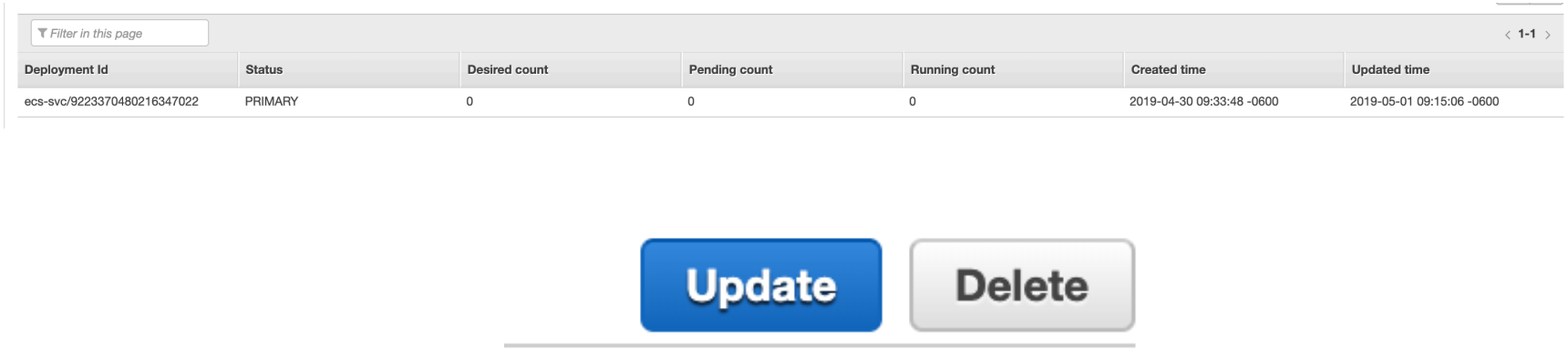


- Set the Number of Tasks to `0` and press Skip to review

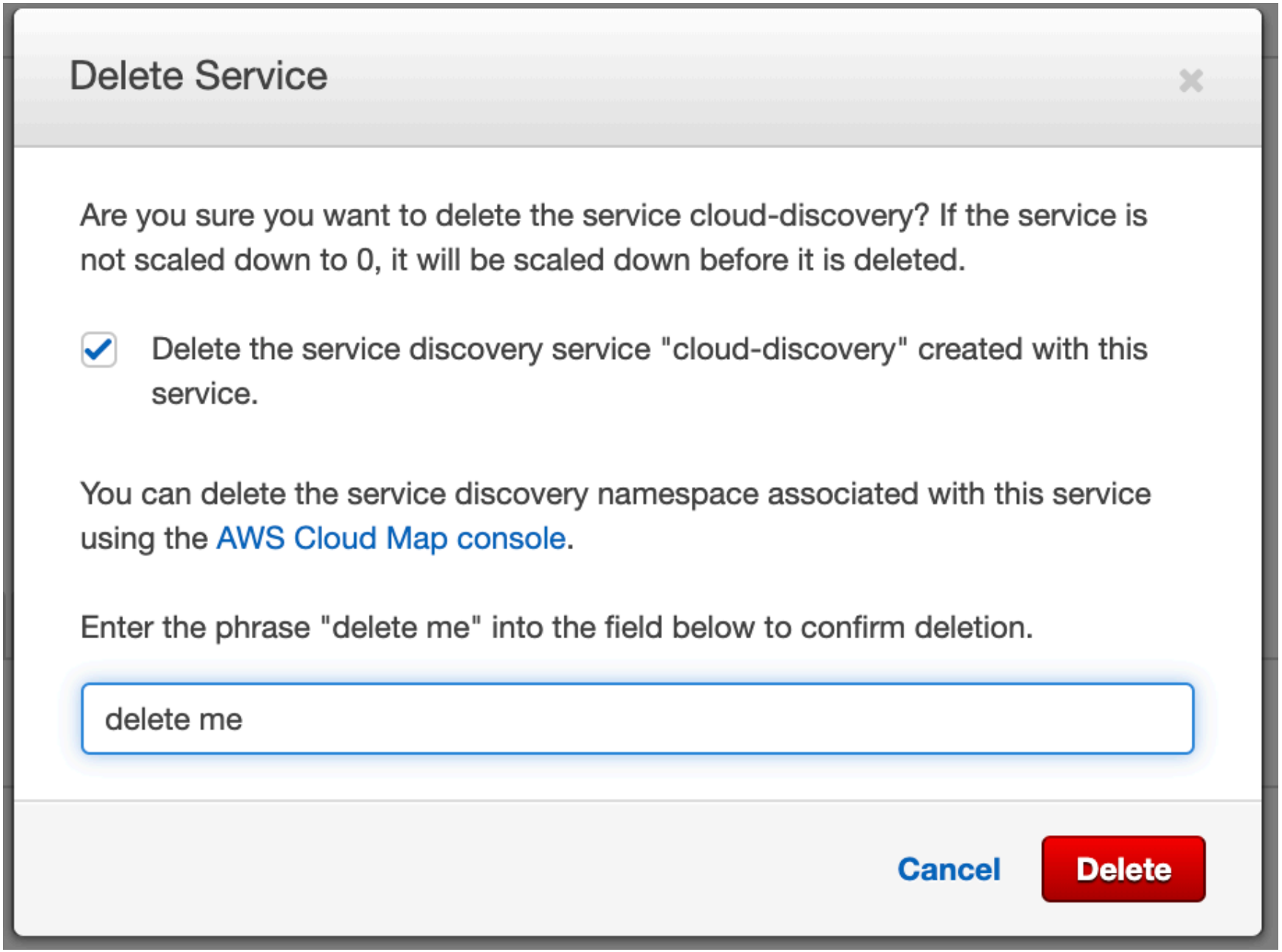


- Press Update service and the View Service

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



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


Delete Service
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.
delete me
Cancel
Delete

Notes:

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

Return to Table of Contents

APPENDIX C

Deploy Stand Alone Microcosm Template

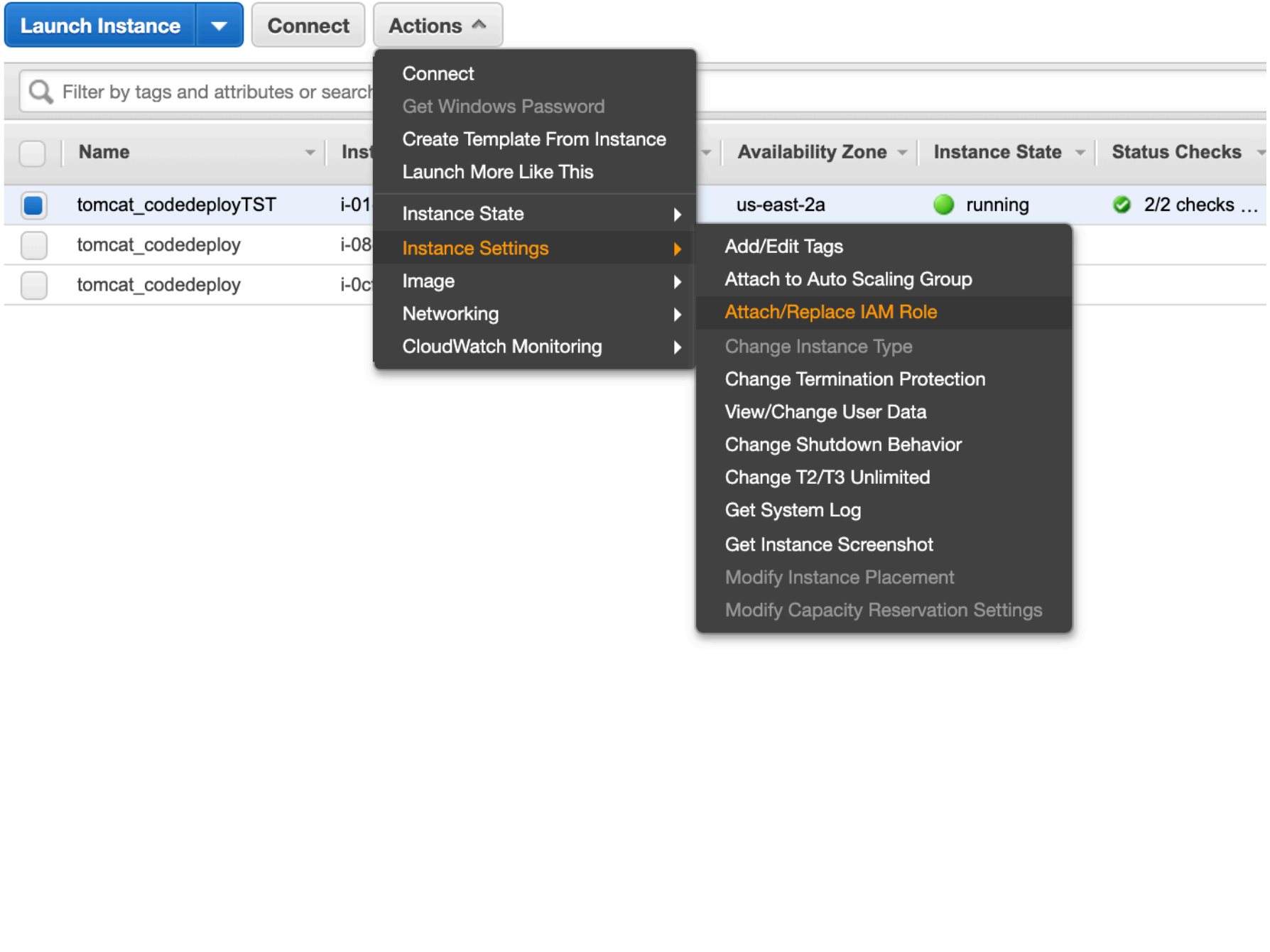
- In Cloud Formation, Create a new stack using the `MicrocosmComponents_StandAlone_AWS.template.yaml` File
- Follow the prompts on the screen and enter the slack hubot information ref
- Follow the instructions in the main section of this document to configure each system in the pipeline

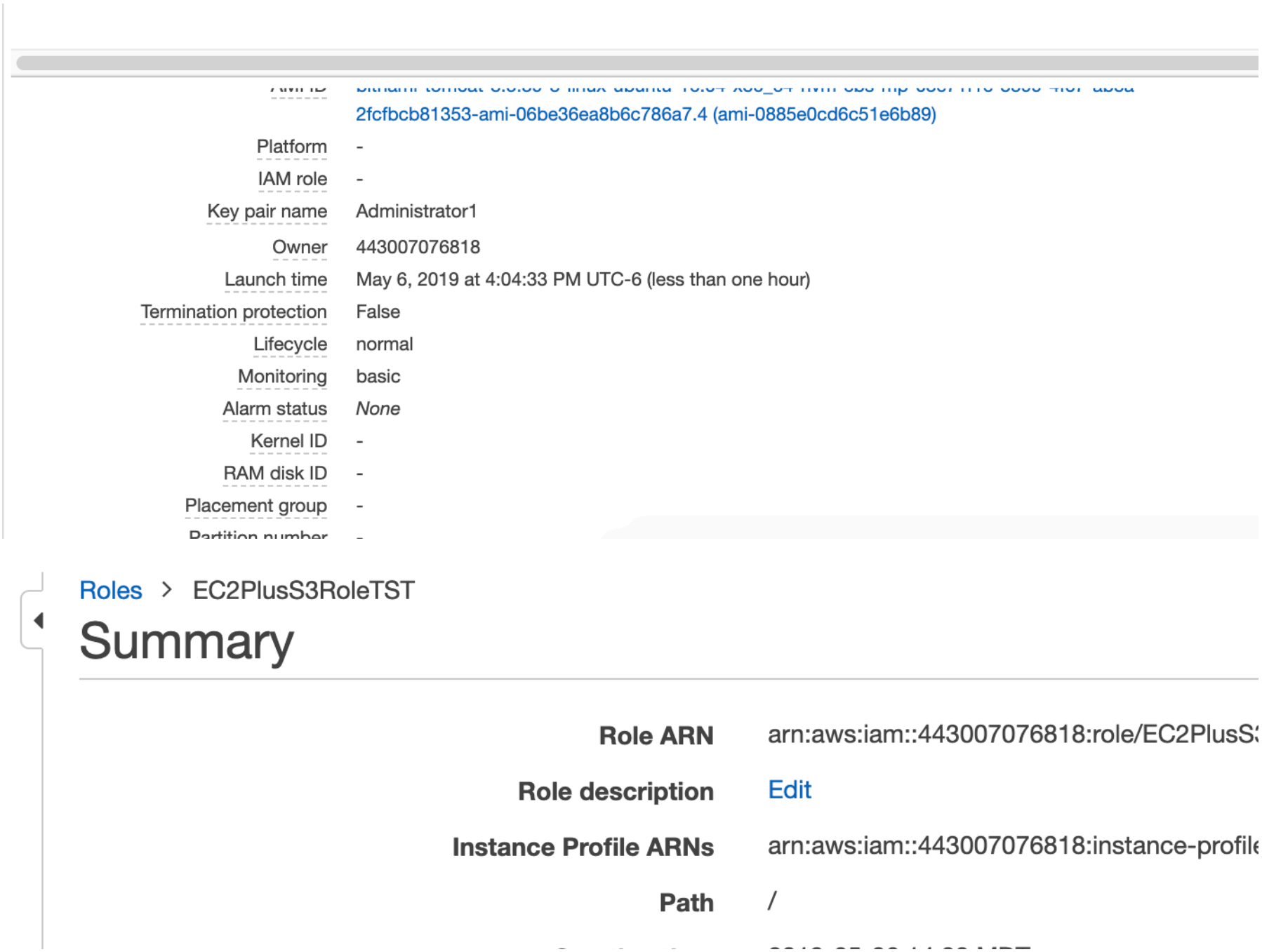
Return to Table of Contents

Deploy Stand Alone Petclinic Deployment Template

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

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


Launch Instance
Connect
Actions
Filter by tags and attributes or search
Name
Inst
Availability Zone
Instance State
Status Checks
tomcat_codedeployTST
i-01
us-east-2a
running
2/2 checks ...
tomcat_codedeploy
i-08
tomcat_codedeploy
i-0c
Connect
Get Windows Password
Create Template From Instance
Launch More Like This
Instance State
Instance Settings
Image
Networking
CloudWatch Monitoring
Add/Edit Tags
Attach to Auto Scaling Group
Attach/Replace IAM Role
Change Instance Type
Change Termination Protection
View/Change User Data
Change Shutdown Behavior
Change T2/T3 Unlimited
Get System Log
Get Instance Screenshot
Modify Instance Placement
Modify Capacity Reservation Settings

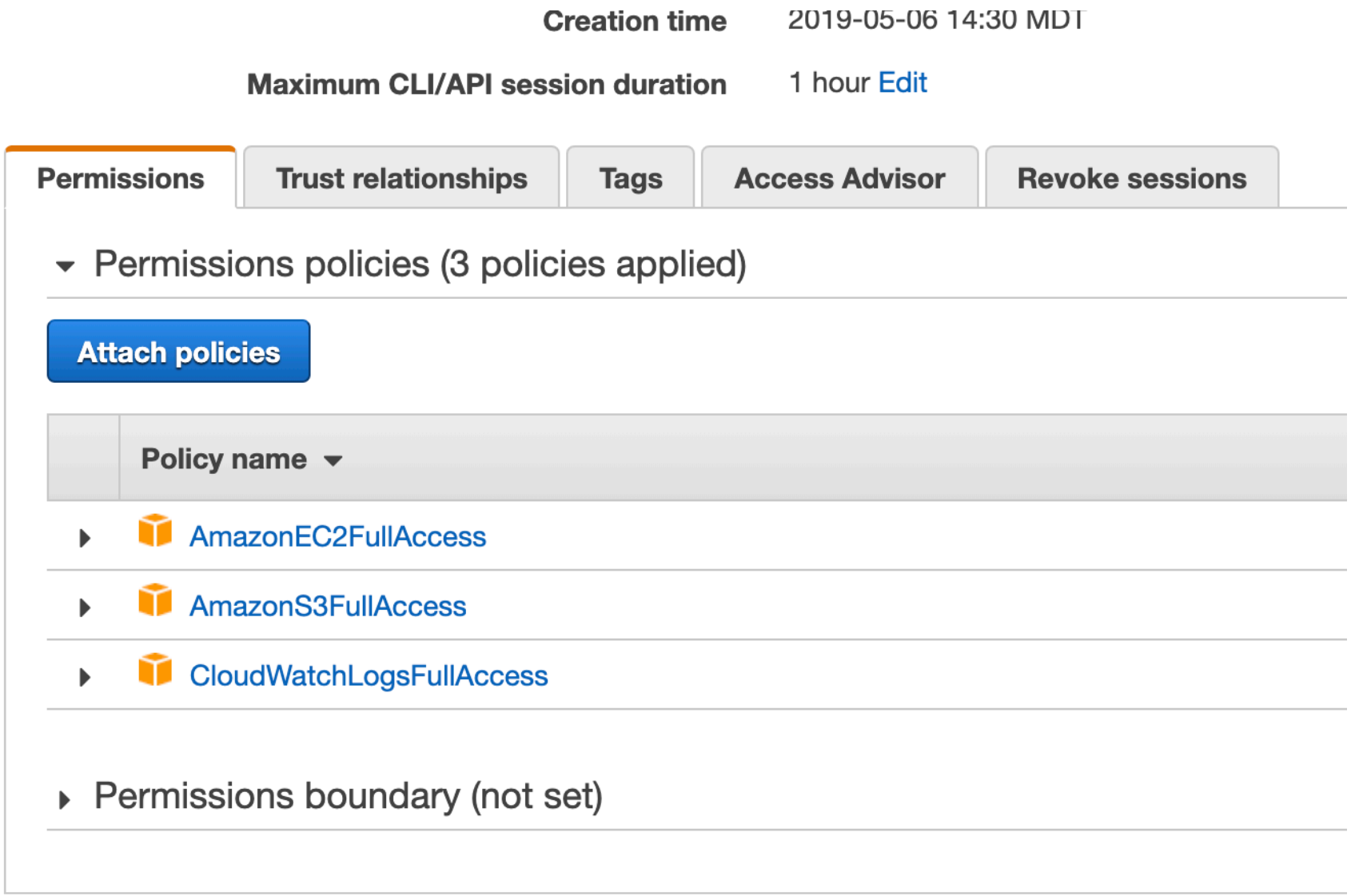


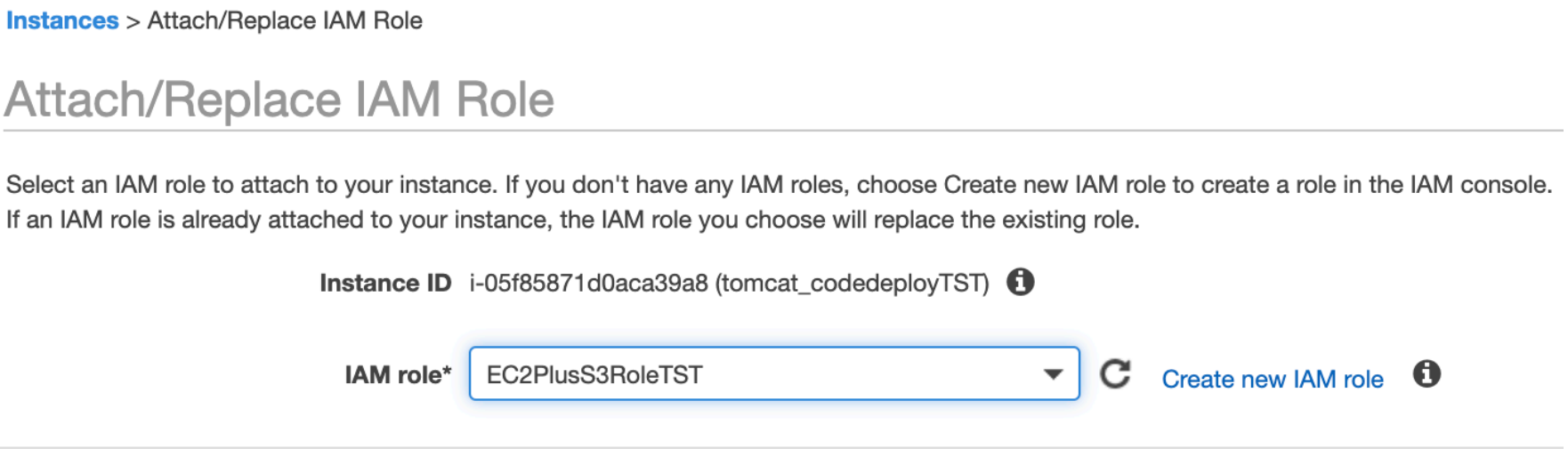
AMI ID	bitnami-tomcat-8.0.53-0-linux-ubuntu-16.04-x86_64-hvm-ebs-mp-2fcfbcb81353-ami-06be36ea8b6c786a7.4 (ami-0885e0cd6c51e6b89)
Platform	-
IAM role	-
Key pair name	Administrator1
Owner	443007076818
Launch time	May 6, 2019 at 4:04:33 PM UTC-6 (less than one hour)
Termination protection	False
Lifecycle	normal
Monitoring	basic
Alarm status	*None*
Kernel ID	-
RAM disk ID	-
Placement group	-
Partition number	-

Roles > EC2PlusS3RoleTST

Summary

Role ARN	arn:aws:iam::443007076818:role/EC2PlusS3
Role description	Edit
Instance Profile ARNs	arn:aws:iam::443007076818:instance-profile
Path	/


Creation time
2019-05-06 14:30 MDT
Maximum CLI/API session duration
1 hour Edit
Permissions
Trust relationships
Tags
Access Advisor
Revoke sessions
Permissions policies (3 policies applied)
Attach policies
Policy name
AmazonEC2FullAccess
AmazonS3FullAccess
CloudWatchLogsFullAccess
Permissions boundary (not set)



Return to Table of Contents

Applying Layer Templates

####In AWS CloudFormation

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

Note Creation can take approx 30 minutes for all layers

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

Return to Table of Contents