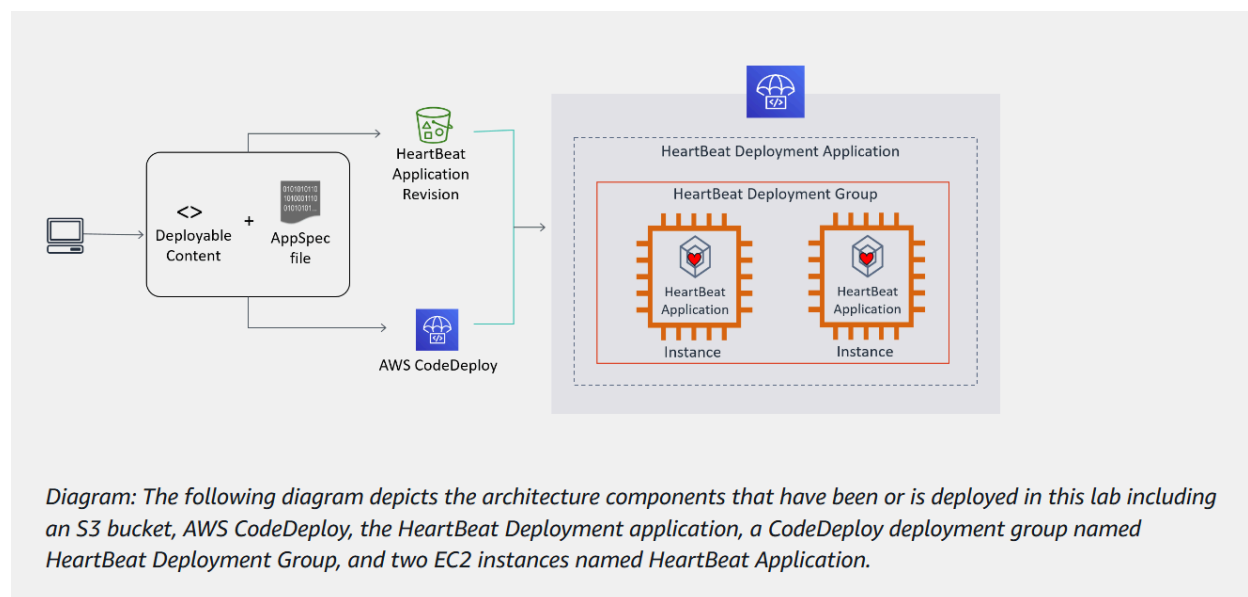


DevOps Lab 2: Deploy an application to an EC2 fleet using AWS CodeDeploy

Objectives according to Lab:

1. Make systematic deployments to a fleet of EC2 servers using CodeDeploy.
2. Verify if the CodeDeploy agent is installed and running on a Windows server.
3. Create a deployment application and group in CodeDeploy.
4. Review and prepare a deployment package to be installed by CodeDeploy.
5. Monitor deployment status in CodeDeploy and deployment targets.



- S3 - Stores code and data for Heartbeat application
- HeartBeat Deployment Application: This is the application that is being deployed.
(more like a function to send heartbeat signals at regular intervals)
- CodeDeploy Deployment Group: A deployment group is a logical collection of instances that CodeDeploy will deploy the application to. In this case, the deployment group is named “HeartBeat Deployment Group”.

- EC2 Instances: These are the Amazon Elastic Compute Cloud (EC2) instances that the application is being deployed to.

Objective 1:

Review EC2 Servers

1.1 Select EC2, and view running instances

Resources EC2 Global view ⚙️ 🔄

You are using the following Amazon EC2 resources in the Asia Pacific (Tokyo) Region:

Instances (running)	3	Auto Scaling Groups	0	Dedicated Hosts	0
Elastic IPs	0	Instances	3	Key pairs	0
Load balancers	0	Placement groups	0	Security groups	3
Snapshots	0	Volumes	3		

1.2 Two Heartbeat instances running, verified

Instances (3) Info 🔄 Connect Instance state ▼ Actions ▼ Launch instances ▼

All states ▼

Instance state = running X Clear filters < 1 > ⚙️

<input type="checkbox"/>	Name ✎	Instance ID	Instance state ▼	Instance type ▼	Status check	Alarm st
<input type="checkbox"/>	aws-cloud9-Lab-2-f...	i-0f279144a00b9c1c0	✓ Running 🔍 🔍	t3.medium	✓ 2/2 checks passed	View ala
<input type="checkbox"/>	HeartBeat-Instance	i-04847b4ab5856054c	✓ Running 🔍 🔍	t3.large	✓ 2/2 checks passed	View ala
<input type="checkbox"/>	HeartBeat-Instance	i-049881c31e88458c5	✓ Running 🔍 🔍	t3.large	✓ 2/2 checks passed	View ala

Objective 2

Review CodeDeploy

ELI5:

CodeDeploy is like a special delivery system for software. It takes your application code and makes sure it gets safely and efficiently delivered to all the computers or servers where you want it to run. Whether you have just one server or hundreds, CodeDeploy helps you manage the process of getting your software up and running smoothly.

2.1 Choose one HB instance, and select Connect



Instances (1/3) Info Refresh Connect Instance state Actions Launch instances

Find Instance by attribute or tag (case-sensitive) All states

Instance state = running Clear filters < 1 > Settings

	Name	Instance ID	Instance state	Instance type	Status check	Alarm st
<input type="checkbox"/>	aws-cloud9-Lab-2-f...	i-0f279144a00b9c1c0	Running	t3.medium	2/2 checks passed	View ala
<input checked="" type="checkbox"/>	HeartBeat-Insti	i-04847b4ab5856054c	Running	t3.large	2/2 checks passed	View ala
<input type="checkbox"/>	HeartBeat-Instance	i-049881c31e88458c5	Running	t3.large	2/2 checks passed	View ala

2.2 In the CLI window, enter the following command

```
powershell.exe -Command Get-Service -Name codedeployagent
```

```
Session ID: a54d7174-4dc8-4dea-9460-704a39a0ddb9-7thg5pgli3pjkah5yj42armzou Instance ID: i-04847b4ab5856054c Terminate
```

```
PS C:\Windows\system32> powershell.exe -Command Get-Service -Name codedeployagent
```

Output: Verified

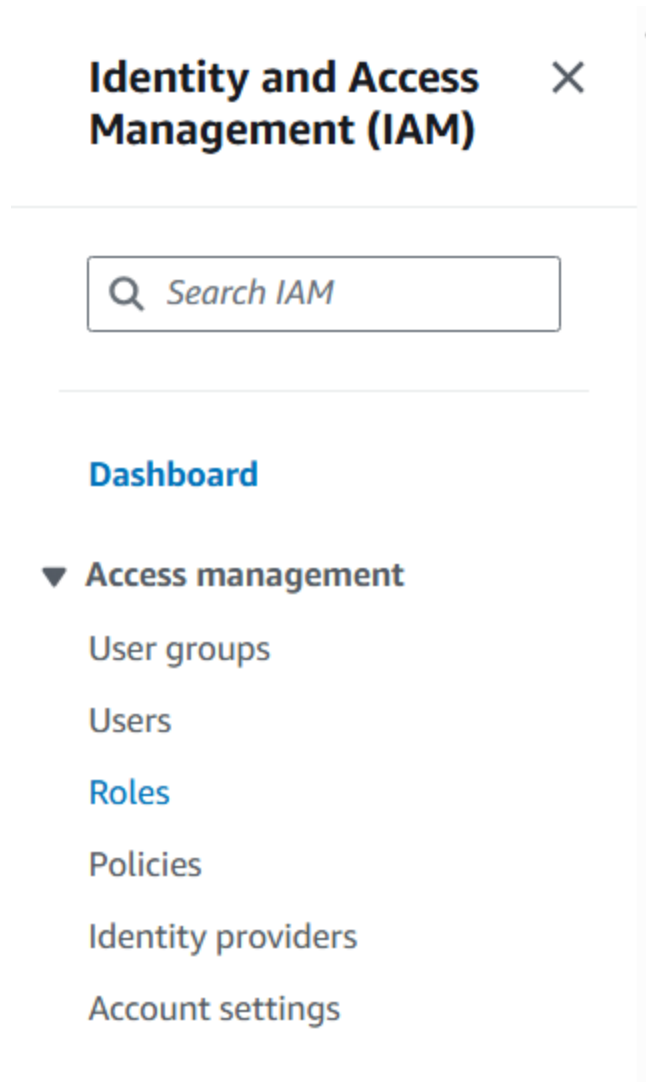
```
PS C:\Windows\system32> powershell.exe -Command Get-Service -Name codedeployagent
```

Status	Name	DisplayName
Running	codedeployagent	CodeDeploy Host Agent Service

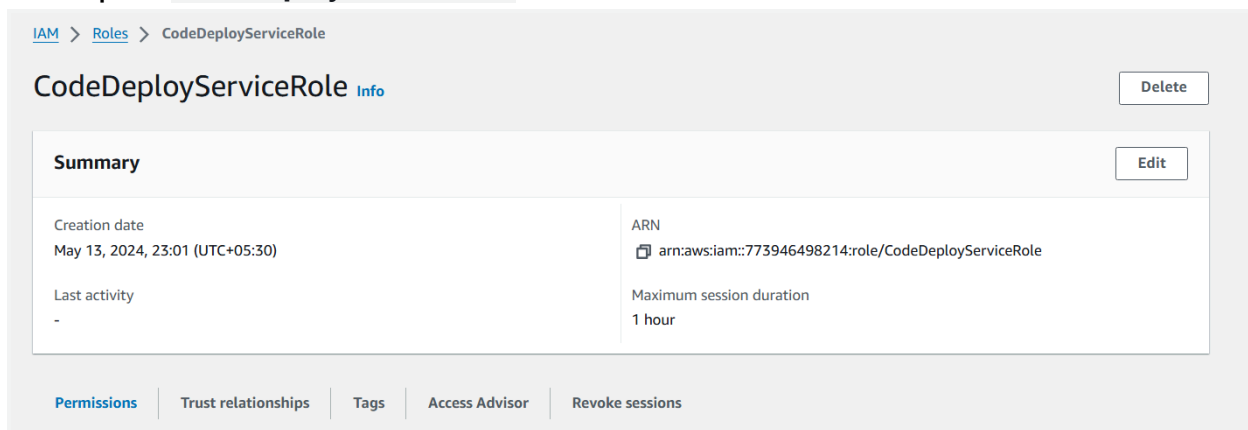
Objective 3:

REVIEW PERMISSIONS FOR CODEDEPLOY

3.1 Open IAM section and select Roles in left pane



3.2 Open `CodeDeployServiceRole`



3.3 Verify Permissions and if “managed policy” is attached to it

Permissions Trust relationships Tags Access Advisor Revoke sessions

Permissions policies (1) Info Refresh Simulate Remove Add permissions

You can attach up to 10 managed policies.

Filter by Type

All types < 1 > Settings

<input type="checkbox"/>	Policy name	Type	Attached entities
<input type="checkbox"/>	AWSCodeDeployRole	AWS managed	1

The policy which has been attached:

AWSCodeDeployRole Copy JSON

Provides CodeDeploy service access to expand tags and interact with Auto Scaling on your behalf.

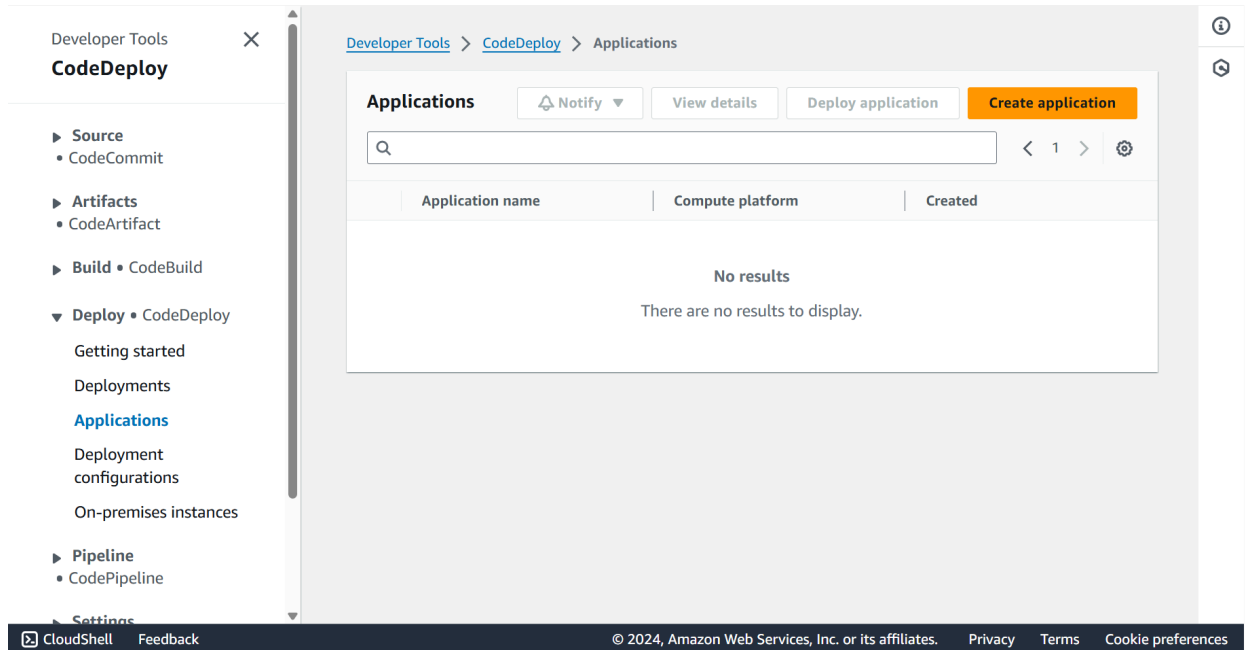
```

1  {
2    "Version": "2012-10-17",
3    "Statement": [
4      {
5        "Effect": "Allow",
6        "Action": [
7          "autoscaling:CompleteLifecycleAction",
8          "autoscaling:DeleteLifecycleHook",
9          "autoscaling:DescribeAutoScalingGroups",
10         "autoscaling:DescribeLifecycleHooks",
11         "autoscaling:PutLifecycleHook",
12         "autoscaling:RecordLifecycleActionHeartbeat",
13         "autoscaling:CreateAutoScalingGroup",
14         "autoscaling:CreateOrUpdateTags",
15         "autoscaling:UpdateAutoScalingGroup",
16         "autoscaling:EnableMetricsCollection",
17         "autoscaling:DescribePolicies",
18         "autoscaling:DescribeScheduledActions",
19         "autoscaling:DescribeNotificationConfigurations",
20         "autoscaling:SuspendProcesses"

```

Objective 4 - Create CodeDeploy Application and Deployment Group

4.1 Open CodeDeploy in AWS console and open Applications, from left pane



4.2 Create a platform with following specs:

[Developer Tools](#) > [CodeDeploy](#) > [Applications](#) > Create application

Create application

Application configuration

Application name
Enter an application name

100 character limit

Compute platform
Choose a compute platform

EC2/On-premises ▼

Tags

Add tag

[Cancel](#) [Create application](#)

4.3 Choose Create Dep. Group option

Deployments

Deployment groups

Revisions

Deployment groups

View details

Edit

Create deployment group

< 1 > ⚙

Name	Status	Last attempt...	Last successf...	Trigger count
<div>No deployment groups</div> <div>Before you can deploy your application using CodeDeploy, you must create a deployment group.</div> <div>Create deployment group</div>				

4.4 Select/fill these values

Deployment group name

Enter a deployment group name

HeartBeatProduction-App-Group

100 character limit

Service role

Enter a service role

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

arn:aws:iam::773946498214:role/CodeDeployServiceRole

Deployment type

Choose how to deploy your application

☒ In-place

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

☐ Blue/green

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

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 is deployed and the success or failure conditions for a deployment.

CodeDeployDefault.AllAtOnce



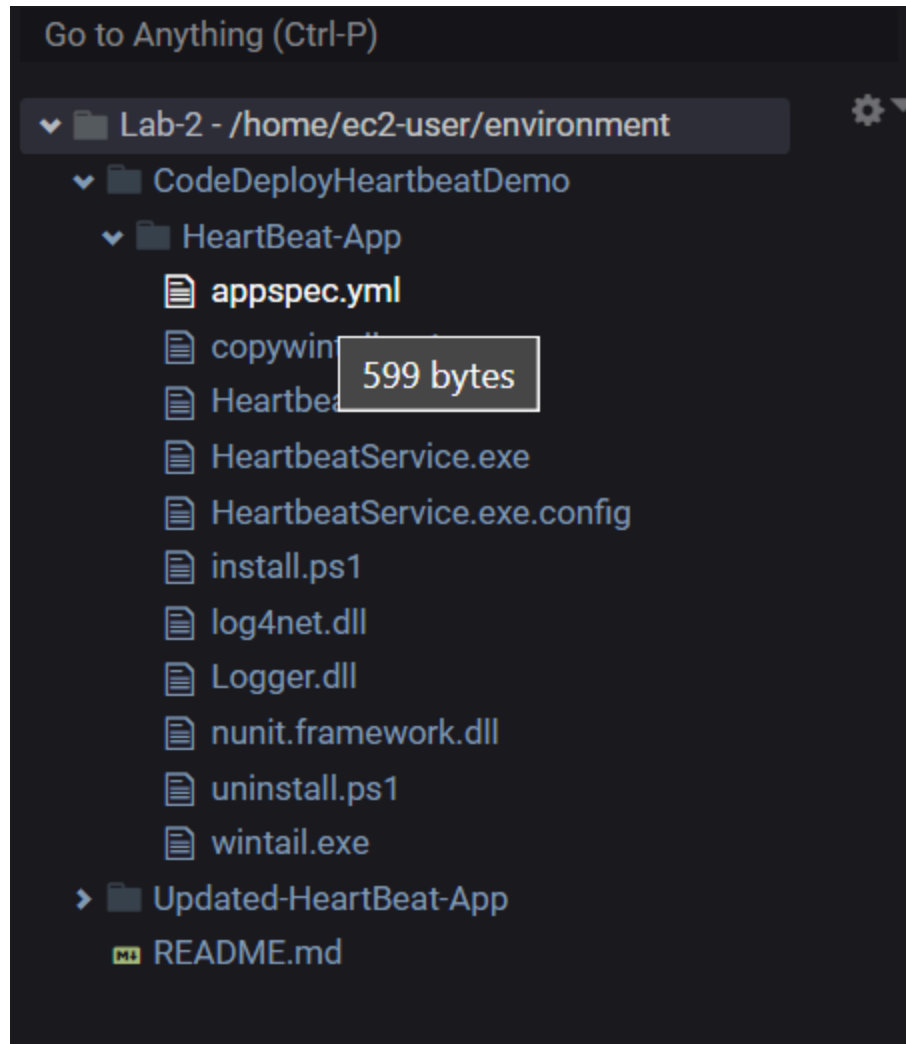
or

Create deployment configuration

Click on Create Group

Objective 5 - Review the application bundle

5.1 Open Cloud9 using the URL given, and choose the specified file



5.2 Review the file

Expected	Actual
----------	--------



```
version: 0.0
os: windows
files:
  - source: Heartbeat.dll
    destination: c:\HeartbeatService
  - source: HeartbeatService.exe
    destination: c:\HeartbeatService
  - source: HeartbeatService.exe.config
    destination: c:\HeartbeatService
  - source: log4net.dll
    destination: c:\HeartbeatService
  - source: Logger.dll
    destination: c:\HeartbeatService
  - source: wintail.exe
    destination: c:\temp

hooks:
  ApplicationStop:
    - location: uninstall.ps1
      timeout: 30
  AfterInstall:
    - location: install.ps1
      timeout: 30
    - location: copywintail.ps1
      timeout: 30
```

```
1 version: 0.0
2 os: windows
3 files:
4   - source: Heartbeat.dll
5     destination: c:\HeartbeatService
6   - source: HeartbeatService.exe
7     destination: c:\HeartbeatService
8   - source: HeartbeatService.exe.config
9     destination: c:\HeartbeatService
10  - source: log4net.dll
11    destination: c:\HeartbeatService
12  - source: Logger.dll
13    destination: c:\HeartbeatService
14  - source: wintail.exe
15    destination: c:\temp
16
17 hooks:
18   ApplicationStop:
19     - location: uninstall.ps1
20       timeout: 30
21   AfterInstall:
22     - location: install.ps1
23       timeout: 30
24     - location: copywintail.ps1
25       timeout: 30
26
```

Verified

Objective 6 - Deploying the Bundle

6.1 Run the following code in terminal

```
AWSLabsUser-5QpvY9aWGTARFA7cFEJrZ2:~/environment $ bucketName=heartbeat-codedeploy-artifacts-agg-20103
AWSLabsUser-5QpvY9aWGTARFA7cFEJrZ2:~/environment $ aws s3 mb s3://$bucketName
make_bucket: heartbeat-codedeploy-artifacts-agg-20103
```

6.2 Deploy the application to CodeDeploy Targets

Run the following two commands

in second command: push the revision to the S3 bucket that you created in the previous step, and register the information with CodeDeploy

```
AWSLabsUser-5QpvY9aWGTARFA7cFEJrZ2:~/environment $ cd ~/environment/CodeDeployHeartbeatDemo
AWSLabsUser-5QpvY9aWGTARFA7cFEJrZ2:~/environment/CodeDeployHeartbeatDemo (main) $ aws deploy push --application-name HeartBeatProduction
--App --source HeartBeat-App --s3-location s3://$bucketName/HeartBeat-App.zip
```

6.3 To deploy the application revision from the Amazon S3 bucket that you created earlier to the target (Amazon EC2) instances, run the following command:

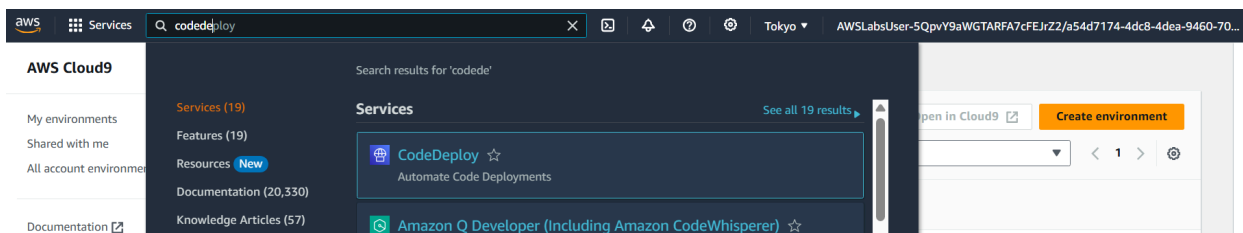
This command creates a deployment associated with the following using the application revision named HeartBeat-App.zip in the Amazon S3 bucket that you created earlier:

- Application named CodeDeploy-Demo
- Deployment configuration named CodeDeployDefault.AllAtOnce
- Deployment group named HeartBeat-Deployment

```
AWSLabsUser-5QpvY9aWGTARFA7cFEJrZ2:~/environment/CodeDeployHeartbeatDemo (main) $ aws deploy create-deployment --application-name HeartBeatProduction-App --deployment-group-name HeartBeatProduction-App-Group --deployment-config-name CodeDeployDefault.AllAtOnce --description "Initial Deployment" --s3-location bucket=$bucketName,key=HeartBeat-App.zip,bundleType=zip
{
  "deploymentId": "d-73EU4LXP5"
}
```

Objective 7 - Monitor Deployment Status

7.1 Go to Cloud9 Dashboard, and search and select CodeDeploy



7.2 Match the deployment id and select that deployment

A screenshot of the AWS CodeDeploy 'Deployment history' table. The table has columns: Deployment Id, Status, Deployments, Compute Resources, Application, Deployment Group, Revision ID, Initiating User, and Start Time. The first row shows a deployment with ID 'd-73EU4LXP5', Status 'Succeeded', and Start Time 'May 14'.

Deployment Id	Status	Deployments	Compute Resources	Application	Deployment Group	Revision ID	Initiating User	Start Time
d-73EU4LXP5	Succeeded	In-place	EC2/On-premises	HeartBeat...	HeartBeat...	s3://heart...	User action	May 14

7.3 Two deployment id for two Heartbeat Applications

Both of them have a status of **In Progress**. This is because you used the deployment configuration **CodeDeployDefault.AllAtOnce** in your CLI call.

This instructs CodeDeploy to deploy the update to all servers at the same time.

Deployment lifecycle events						
<div><input type="text"/></div> <div>< 1 > ⚙</div>						
Instance ID	Duration	Status	Most recent event	Events	Start time	End time
i-04847b4ab5856054c	11 seconds	<div>Succeeded</div>	ValidateService	View events	May 14, 2024 12:25 AM (UTC+5:30)	May 14, 2024 12:25 AM
i-049881c31e88458c5	11 seconds	<div>Succeeded</div>	ValidateService	View events	May 14, 2024 12:25 AM (UTC+5:30)	May 14, 2024 12:25 AM

7.4 Success in Deployment of the two EC2 instances

7.5 Open EC2 Instances, and connect to Heartbeat Instance

AWS

Services

Search

[Alt+S]