CODE DEPLOY

* It is a deployment service that automates application deployments in ec2 instances (or) on-permises instances (physical servers).
* You can deploy your source code from S3, github.
* To use code-deploy, first we have to create an **instance** **role** (To gives access to ec2 to access other aws resources) and **code**-**deploy** **service** **role** (To gives access to code-deploy to use other aws resources).
* To create instance profile role. First you have to create instance profile policy. Go to IAM Policies and click create policies. In JSON field paste this.

**{**

**"Version": "2012-10-17",**

**"Statement": [**

**{**

**"Action": [**

**"s3:Get\*",**

**"s3:List\*"**

**],**

**"Effect": "Allow",**

**"Resource": "\*"**

**}**

**]**

**}**

* After creating this policy, create a role with name instance profile and attach this policy to that role.
* To create code deploy service role.

**Go to roles.**

**Select code deploy.**

**Search for service role.**

**Select the policy.**

**Create the service role.**

* We have to create instances and install **code** **deploy** **agent** in those instances after that we can deploy the code to those instances.
* To deploy code in instances, first we have to write a file called **APPSPEC**.**YML**, in this file we will write the steps on how the code should be deployed in the instances. This file is written in **yaml** script.
* Code Deploy supports 2 types of deployments.

**IN-place** = deploys code in running instances.

**Blue/green** = Creates new instances and deploys latest version code in those instances.

* Deploying to instances has 3 types.

**Deploy all at a time** = Deploys code to all instances at same time..requires downtime.

**Deploy once at a time** = Deploys one instance at a time.

**Deploys halt at a time** = Half number of instances at a time.

INSTALL CODE-DEPLOY AGENT

* Download file from this url,

**wget https://bucket-name.s3.amazonaws.com/latest/install**

* Aws provides different s3 url’s for different regions. For ex: **aws-codedeploy-ap-south-1a** for Mumbai.
* Choose and Type the url based on your region In the place of bucket name.
* A file called **INSTALL** will be downloaded and give **execute** permissions for that file.
* **./install auto** = to install code-deploy agent.
* **Service codedeploy-agent status** = To see code-deploy agent is running (or) not.
* Now, you can deploy your code to this instance with aws code-deploy service.

CREATE AN APPLICATION

* To Create an Application.

**Go to Code-deploy Console.**

**Click create application.**

**Type Application name.**

**Type Deployment group name.**

**Select Deployment type (in-place or blue/green).**

**Select environment** (asg or instances or on-permises).

**Choose instances to this deployment group**.

**Select Deployment configuration**(all, once, half).

**Select Service Role**.

**Click Create Application.**

* After creating application, click on the application name, you will see deployment groups, revisions.
* To deploy latest application content,

**Select deployment group.**

**Click Actions.**

**Click deploy new version.**

* In deployment page,

**Select your application.**

**Select your deployment group.**

**Select repo type**(s3, git).

**If you select S3, type the url of your project dir.**

**If you select GIT, paste the git repo url in the field and paste commit ID to take code.**

**Click, Create.**

* To deploy a new version again,

**Click on Create Deployment.**

**Select your application.**

**Select deployment group.**

**Type commit ID and git repo.**

**Click Create.**

* It will download the source bundle from git and install latest version based on the order in appspec.yml file and once it successed try browse the page with instance public IP (or) domain name. You will see latest version of your code.
* You can create multiple deployment groups with instances for different scenarios like test and production.
* So that, You can test your latest code in test deployment group and once it successed you can deploy it in production deployement group.
* You can see all your deployed versions in the application page. For every deployment, you will get an deployment ID. With this ID, you can see all the content and logs.
* The deployment data will be stored in their respective instance along with the deployment ID. **(/opt/codedeploy-agent/deployment-root//deployment-ID).**

CODE-DEPLOY LOGS

* Code-deploy logs are stored in the respective instances within the deployment ID directories.
* **/opt/codedeploy-agent/deployment-root/deployment-logs/** = It will store all deployment logs individually along with its deployment ID.
* **/var/log/aws/codedeploy-agent/codedeploy-agent.log =** It will store all agent logs here.
* You can use cloudwatch logs to monitor code-deploy logs in cloud watch console by installing the cloud watch logs agent.
* You have to specify the **logs path** and **instance ID** in **/etc/awslogs/awslogs.conf** file and it will push the logs to cloud watch console.

MONITORING

* There are several aws monitoring tools to monitor code-deploy (cloud watch logs, events, cloud trail and sns).
* **Cloud** **Watch** **Logs** = It will push application logs to aws console. We have to install cloud watch logs agent in every instance and configure our application in awslogs.conf file and it will show our application logs in cloud watch console.
* **Cloud** **Trail** = It will capture all API calls made by users along with their ACC ID, source ip, what request made, who and when it happened. You can create a trail and it will store all logs in S3 bucket you specified. It will show logs in cloud trail console.
* **SNS =**  You can create triggers for code-deploy along with SNS notifications. So, whenever a trigger is happened in a code deployement group, an sms (or) email will come the reciepents who are in the sns topic.
* You can create triggers for deployments and for instances states. First, you have to create an SNS topic with recipients to get notified.
* Before you use sns triggers, you have to give access to code-deploy to use sns. For that we need to add sns policy to code-deploy service role**.**
* Go to code-deploy service role,

**Click incline policies,**

**Select custom policy, Type policy name** (snspublish),

**Copy this content in policy area, click validate policy and click apply policy.**

**{**

**"Version": "2012-10-17",**

**"Statement": [**

**{**

**"Effect": "Allow",**

**"Action": "sns:Publish",**

**"Resource": "\*"**

**}**

**]**

**}**

* Now, you can use sns triggers for code-deploy.
* To create Triggers,

**Go to deployement group page,**

**Click create trigger,**

**Type a trigger name,**

**Select events** (deployments (fail, pass, stops etc) or (instances (starts, fails etc)),

**Select sns topic, click create**.

ELB WITH CODE-DEPLOY

* **IN**-**PLACE** = If you are using elb in in-place deployment, elb deregisters instances from load balancers from deployment group and stops routing traffic to that instance. After latest version is deployed successfully in that instance, elb will re-registered to that instance and adds to same deployment group. After this, It will re-routes the traffic to that instance.
* **BLUE/GREEN** = If you are using elb in blue/green deployment, elb will re-routes the traffic to newly created instances, by blocking the traffic to old instances. The new instances will be added to elb and de-registers the old instances from elb.
* You can add elb while creating new deployment group (or) you can edit and add elb to an existing deployment group.
* To add elb to deployment group. Go to Deployment group page,
* Select **ENABLE** load balancer, select type of elb and click create.