ELASTIC BEANSTALK

* Normally, if we create an instance to deploy our application. we have to install web server, choose load balancer, configure auto scaling, create an rds. It is a long process; it will take so much time to deploy an application. To reduce the time, we use elastic beanstalk.
* Beanstalk makes it easy to deploy and manage applications in the cloud. we just need to upload our application and remaining all settings like application install, db install elb, asg etc will be managed by beanstalk automatically (or) We can select manually if we want.
* We have to select which type of instance, which web server to use, choose a platform and upload our application. That's it.
* After you deployed an application, you can update and manage the application versions and other aws infrastructure.
* After creating your application, elastic beanstalk provides you an cname url. With this url you can access your website (or) application.

CREATE AN APPLICATION

* Go to aws console, select beanstalk,

**Click create a new application**.

**Type an application name**.

**Select your environment tier** = web server (or) worker.

**Select an platform** (java, tomcat, python etc).

**Select version if you want**.

**Upload your code through local machine (or) through s3 bucket url.**

* This will create an sample app with settings configured by aws. It will take 5 mins(apprx).
* It will create EC2, load balancer, auto-scaling, s3 to save your source code, EC2 security grp, cloud watch alarms, domain name for your app. These configurations are automatically created by aws.
* If you want to configure these settings manually,

**On Creating Page.**

**Click configure more options.**

* To see the details of your newly created application. Click on your application name there you can see health status, platform and 5 most recent events.
* If you want to change the configuration after creating an application.

**Go to beanstalk,**

**Click configuration.**

* You can change instances, autoscaling, rds, loadbalancers, monitoring etc by clicking GEAR icon.

ENVIRONMENT

* Aws provides a detailed page for you web application created by beanstalk called environment.
* You have to create environments inside application console. You can create multiple environments inside one application console for the test cases.
* In this environment, you can see all your web application configurations and perform custom actions like restarting the server. You can clone your environment and rebuild it from scratch.
* You can terminate the environment, rebuild the environment but the instances will be terminated and new will be created, clone your environment with latest platform version.
* Beanstalk will store application versions in an S3 bucket created by beanstalk along with its version ID.
* With **SWAP url**, you will get the same url even after cloning your environment.
* In configuration page, it shows current configuration of your environment like autoscaling, load balancer, ec2 instances, vpc etc.
* In logs page, you can request logs from ec2 instances to your beanstalk s3 bucket. When you requests logs to this page, beanstalk delete the logs from s3 after 15 minutes. You can configure to upload logs to s3 for permanent even the logs rotate locally.

DEPLOYEMENTS

* To deploy a new version to your application, you need to upload the source bundle(zip (or) war file), aws automatically updates it with all the instances in the beanstalk environment. But, it will use downtime.
* To avoid downtime, we use deployment policies. each deployment is identified by deployment id. It displays the id both in health console and cli and it reports health status of instances if you enabled enhanced monitoring.
* There are 4 types of deployment policies,
* **All at once** - it will start deploying new version at once. it requires downtime.
* **Rolling** - instances divides into batches and apply updates from batch to batch.
* **Rolling with additional batch** - new instances will be created while deploying the old instances. New will be deleted after deployment is completed.
* **Immutable** - launch new instances and deploy updates to those instances. The old instances will be terminated after the new instance passes the health checks.
* To configure Deployments,

**Go to beanstalk page.**

**Select your environment.**

**Choose Configuration.**

* In Updates and deployments page,

**Click GEAR icon.**

**Select type Policy and batch type and batch size.**

**Click save.**

CONFIGURATION OPTIONS

* **INSTANCES** – You can select instance type, size of root volume, volume type. You can use your custom AMI. You have to stop the instances and create an image from it and copy that AMI ID to elastic beanstalk. You can’t add running ec2 instances to elastic beanstalk.
* **AUTOSCALING** –

**Select environment type (singe ec2 or load balanced).**

**Select min and max instances.**

**Select AZ.**

**Type breach time and no of times to scaling.**

**Type upper and lower threshold.**

* **LOAD BANALCER -** You can select port 80 and http protocol. You can add port 443 and https, you have to add ssl certificate through aws or 3rd party. Enable cross-zone balancing and connection draining and set time to draining.
* **UPDATES & DEPLOYEMENTS –**

**Select updates type (rolling or all at once).**

**If you select rolling.**

**Select % (or) no of instances to be updated as a batch**.

* **MONITORING -** Give path to health check. Select enhanced or basic.
* **SNS -** Type an email address to receive notifications from elastic beanstalk when instances errors , scaling occurs.
* **NETWORK –**

**Select your vpc.**

**public (or) private subnets and save.**

* Once you assigned vpc, you can’t change it.
* **RDS -** You can create DB instances with elastic beanstalk.

**Select DB.**

**DB instance type.**

**Storage.**

**multi AZ.**

**snapshot before terminate and save.**

* Once you terminate the beanstalk environment, rds will automatically terminate.
* You can also add existing rds to beanstalk, it will be added to environment. You can’t remove rds from beanstalk after you added once.

MONITORING

* Monitoring console displays the application health and environment status.
* In the left pane click monitoring, it shows the cpu utilization, average latency, network in and out. You can also view overall statistics via graphs.
* By default, basic cloud watch metrics are enabled, which gives five-minute periods. You can enable one-minute period by changing configuration settings.
* You can also add a new graph. By clicking add,

**Select resource load balancer (or) autoscaling.**

**Select metric.**

**Select statistics.**

**Enter description.**

**Click add.**

* By default, EC2 and load balancer metrics are enabled in all environments.

ENHANCED MONITORING

* It gives the detailed metrics of elastic beanstalk with advanced features. Enhanced monitoring includes a health agent in every AMI used in beanstalk. It monitors webserver logs and system metrics. Beanstalk analyzes these metrics to get overall metrics of environment.
* The eb health agent reports metrics to beanstalk every **10** **sec** (default=**5 min**).
* To consider an instance healthy in webserver environment, instance has to pass 12 health checks.
* To enable enhanced monitoring.

**Go to beanstalk environment.**

**On health panel,**

**Click GEAR icon and select enhanced**.

* It is enabled by default in version 2 platforms.
* The environmental health page displays the metrics, health status of each instance in environment. This health page displays only after enhanced monitoring is enabled.
* The health page displays instances health with its Instance ID, deployment ID, time since deployment, version label of code.
* Monitoring represents health status of instances and overall environment based on 4 colors. Even if one instance is unhealthy, it will change the color.

**GREEN** - OK (Instances passing health checks)

**GREY** - PENDING (instance operation is in progress).

**YELLOW** - WARNING (less no of instances is not responding to health checks).

**RED** - DEGRADED (or) SEVERE (more instances are not responding to health checks).

* Beanstalk uses custom web-server log format to get info about requests to enhanced monitoring, which analyzes logs, identifies issues and sets the instances healthy.
* It provides logs with detailed request info, client info and with a time-stamp.
* EC2 instances in eb-env generate logs of webserver, application server etc and store logs locally by individual instances. You can use these logs to troubleshoot problems.
* You can request last **100** logs from aws console in eb-env called **tail** logs. You can retrieve **bundle** (**all**) logs from eb-env. It creates a file with logs and uploads it to S3. The logs will be deleted after 15 minutes.
* Logs are stored in general location in an instance -

**/var/log/eb-activity.log**

**/var/log/eb-commandprocessor.log**

**/var/log/eb-version-deployment.log**

EB-CLI

# You can use cli to create and manage elastic beanstalk.

# For this we have to install PYTHON, PIP and EBCLI.

# To install eb cli, first you have to install PIP.

# To install pip,

# Yum install python-pip.

# After that we have to install EB-CLI,

# Pip install awsebcli --upgrade --user

# Now, you have installed eb-cli. But, you can’t use eb-cli because, you have to set home path to eb-cli.

# Go to .bashrc file,

**export PATH=LOCAL\_PATH:$PATH and save the file.**

**source ~/bashrc – To Update the bashrc file.**

# Now, you can use ebcli. To check whether it’s installed (or) Not, type eb --version. It will show you the eb-cli version.

# If your source code is in GIT, Initialize git and initialize eb in your project dir. Type EB INIT.

# It will ask you,

# REGION, ACCESS KEYS, APPLICATION NAME, PLATFORM, SSH USE, KEY PAIR.

# After initializing EB, you have to create the eb environment. Type EB CREATE.

# It will ask you,

# ENV NAME, CNAME and shows all the details we selected while EB INIT and creates ENVIRONMENT.

# It will take few minutes to create the complete environment. You can see the process in cli (or) in aws console.

# EB STATUS = It will shows the status and details of your application environment (env id, deployed version, platform, region, cname, last updated, status, health and app name).

# EB HEALTH = It will show the health of instances in eb and overall environment (cpu, requests per sec, load average etc).

# Use –refresh = to update health every 10 seconds.

# EB EVENTS = It will show you the list of events line by line in cli.

# EB LOGS = It will pull logs from an instance to your environment. By default, it will show first instance logs. If you want specific instance logs type = eb logs --instanceid..

# Eb logs –all = It pull logs from all instances to environment and saves them to sub-dir under .elasticbeanstalk/logs.

# EB CONFIG = To show all configuration options from your running environment.

# EB OPEN = It opens your environment website in a browser.

# EB DEPLOY = It deploys the latest code (or) commit to environment in your current project directory.

# EB TERMINATE = It terminates the environment.

# EB CODESOURCE codecommit = To get source code from aws codecommit. It will shows repo and branches from code commit. Select your repo to create application.