

# Deploy to AWS with Elastic Beanstalk

With Elastic Beanstalk, you can quickly deploy and manage applications in the AWS Cloud without worrying about the infrastructure that runs those applications. Elastic Beanstalk reduces management complexity without restricting choice or control. You simply upload your application, and Elastic Beanstalk automatically handles the details of capacity provisioning, load balancing, scaling, and application health monitoring.

## Before you begin:

- An IAM user is configured with sufficient permissions to perform a deployment to your application and upload artifacts to the S3 bucket.
- You have configured the Elastic Beanstalk application and environment.
- Set up an S3 bucket where deployment artifacts will be copied. The name of your S3 bucket should be: "*<application name>-elasticbeanstalk-deployment*", where *<application name>* is the name of your Elastic Beanstalk application.

## Steps

1. Clone the [Elastic Beanstalk pipe example repository](#)
2. Add your AWS credentials to Bitbucket Pipelines. In your repo go to **Settings**, under *Pipelines*, select **Repository variables** and add the following variables:

### Basic usage variables

- **AWS\_ACCESS\_KEY\_ID (\*)**: Your AWS access key.
- **AWS\_SECRET\_ACCESS\_KEY (\*)**: Your AWS secret access key. Make sure that you save it as a secured variable.
- **AWS\_DEFAULT\_REGION (\*)**: The AWS region code (us-east-1, us-west-2, etc.) of the region containing the AWS resource(s). For more information, The AWS region code (us-east-1, us-west-2, etc.) of the region containing the AWS resource(s). For more information, see [Regions and Endpoints](#).
- **APPLICATION\_NAME (\*)**: The name of the Elastic Beanstalk application.
- **ENVIRONMENT\_NAME (\*)**: Environment name.
- **ZIP\_FILE (\*)**: The application source bundle to deploy (zip, jar, war).
- **(\*) = required variable**.

### Example

```
1 image: atlassian/default-image:2
2
3 pipelines:
4   default:
5     - step:
6       name: "Build and Test"
7       script:
8         - pytest test/test.py
9         - zip application.zip Dockerfile application.py cron.yaml Dockerrun.aws.
10        # Define an artifact to pass the zip file to the next step
11        artifacts:
12          - application.zip
13      - step:
14        name: "Deploy to Production"
15        # Track production environments builds using deployments.
16        deployment: production
17        script:
18          - pipe: atlassian/aws-elasticbeanstalk-deploy:0.5.0
19            variables:
20              AWS_ACCESS_KEY_ID: $AWS_ACCESS_KEY_ID
21              AWS_SECRET_ACCESS_KEY: $AWS_SECRET_ACCESS_KEY
22              AWS_DEFAULT_REGION: "us-east-1"
23              APPLICATION_NAME: "application-test"
24              ENVIRONMENT_NAME: 'production'
25              ZIP_FILE: "application.zip"
```

By default, the new application revision will have the version label "*<application name>-<bitbucket build number>-<short commit hash>*" but this can be changed using the VERSION\_LABEL variable.

## Outcome

You're now set up to deploy a new version of your application to Elastic Beanstalk.

## Advanced usage

For advanced use cases It's recommended the *build once and deploy many* approach. If you have multiple environments, use `COMMAND` variable to separate your CI/CD workflow into different operations/pipes:

- `COMMAND: 'upload-only'`: It will upload the artifact and release a version in Elastic Beanstalk.
- `COMMAND: 'deploy-only'`: It will deploy the specified version to the desired environment(s)

### Upload-only variables

- **AWS\_ACCESS\_KEY\_ID (\*)**: Your AWS access key.
- **AWS\_SECRET\_ACCESS\_KEY (\*)**: Your AWS secret access key.
- **AWS\_DEFAULT\_REGION (\*)**: The AWS region code (us-east-1, us-west-2, etc.) of the region containing the AWS resource(s). For more information, see [Regions and Endpoints](#).
- **APPLICATION\_NAME (\*)**: The name of the Elastic Beanstalk application.
- **COMMAND (\*)**: Command to be executed during the deployment. Valid options are `all`, `upload-only`, `deploy-only`. Default: `all`.
- **ZIP\_FILE (\*)**: The application source bundle to deploy (zip, jar, war).
- **S3\_BUCKET**: Bucket name used by Elastic Beanstalk to store artifacts. Default: `${APPLICATION_NAME}-elasticbeanstalk-deployment`
- **VERSION\_LABEL**: Version label for the new application revision. Default: `${ENVIRONMENT_NAME}-${BITBUCKET_COMMIT:0:8}_YYYY-mm-dd_HHMMSS`.
- **DESCRIPTION**: Description for the new application revision. Default: a URL pointing to the pipeline result page.
- **DEBUG**: Turn on extra debug information. Default: `false`.

**(\*) = required variable**.

Upload the artifact `application.zip`, and create a version `deploy-$BITBUCKET_BUILD_NUMBER-multiple` in Elastic Beanstalk.

### Example

```
1 pipe: atlassian/aws-elasticbeanstalk-deploy:0.5.0
2 variables:
3   AWS_ACCESS_KEY_ID: $AWS_ACCESS_KEY_ID
4   AWS_SECRET_ACCESS_KEY: $AWS_SECRET_ACCESS_KEY
5   AWS_DEFAULT_REGION: $AWS_DEFAULT_REGION
6   APPLICATION_NAME: 'application-test'
7   COMMAND: 'upload-only'
8   ZIP_FILE: 'application.zip'
9   S3_BUCKET: 'application-test-bucket'
10  VERSION_LABEL: 'deploy-$BITBUCKET_BUILD_NUMBER-multiple'
```

### Deploy-only variables

- **AWS\_ACCESS\_KEY\_ID (\*)**: Your AWS access key.
- **AWS\_SECRET\_ACCESS\_KEY (\*)**: Your AWS secret access key.
- **AWS\_DEFAULT\_REGION (\*)**: The AWS region code (us-east-1, us-west-2, etc.) of the region containing the AWS resource(s). For more information, see [Regions and Endpoints](#).
- **APPLICATION\_NAME (\*)**: The name of the Elastic Beanstalk application.
- **COMMAND (\*)**: Command to be executed during the deployment. Valid options are `all`, `upload-only`, `deploy-only`. Default: `all`.
- **ENVIRONMENT\_NAME (\*)**: Environment name.
- **ZIP\_FILE (\*)**: The application source bundle to deploy (zip, jar, war).
- **S3\_BUCKET**: Bucket name used by Elastic Beanstalk to store artifacts. Default: `${APPLICATION_NAME}-elasticbeanstalk-deployment`
- **VERSION\_LABEL**: Version label for the new application revision. Default: `${ENVIRONMENT_NAME}-${BITBUCKET_COMMIT:0:8}_YYYY-mm-dd_HHMMSS`.
- **WAIT**: Wait for deployment to complete. Default: `false`.
- **WAIT\_INTERVAL**: Time to wait between polling for deployment to complete (in seconds). Default: `10`.
- **DEBUG**: Turn on extra debug information. Default: `false`.

**(\*) = required variable**.

Deploy your version `deploy-$BITBUCKET_BUILD_NUMBER-multiple` into the environment `production` and wait until the deployment to complete.

### Example

```
1 - pipe: atlassian/aws-elasticbeanstalk-deploy:0.5.0
2   variables:
3     AWS_ACCESS_KEY_ID: $AWS_ACCESS_KEY_ID
4     AWS_SECRET_ACCESS_KEY: $AWS_SECRET_ACCESS_KEY
5     AWS_DEFAULT_REGION: $AWS_DEFAULT_REGION
6     APPLICATION_NAME: 'application-test'
7     COMMAND: 'deploy-only'
8     VERSION_LABEL: 'deploy-$BITBUCKET_BUILD_NUMBER-multiple'
9     ENVIRONMENT_NAME: 'production'
10    WAIT: 'true'
```

Access Pipelines deployment guides

[Deploy to AWS with S3](#)

[Deploy to AWS with CodeDeploy](#)

- [Deploy to AWS with Elastic Beanstalk](#)

[Deploy to AWS EKS \(Kubernetes\)](#)

[Deploy a Lambda function update to AWS](#)

Show more

On this page

Before you begin:

[Steps](#)

[Advanced usage](#)

Community

[Questions, discussions, and articles](#)

Was this helpful? ☐ Yes ☐ No [Provide feedback about this article](#)

### Still need help?

The Atlassian Community is here for you.

Ask the Community