Commit To Deployment

By, Shubhasish Panda

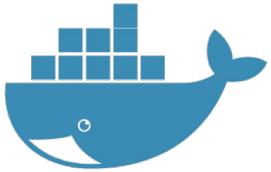
Problem

1) A “Pre-Commit” And a “Post-Commit” pipeline
2) Three separate environments for Devs, QA and prod
3) Fully automatic deployment via Jenkins
4) Merge requests triggers Jenkins Job
5) Each job per branch
6) Review board for code review

Solution

Jenkins

AWS ECS



Infrastructure Strategy

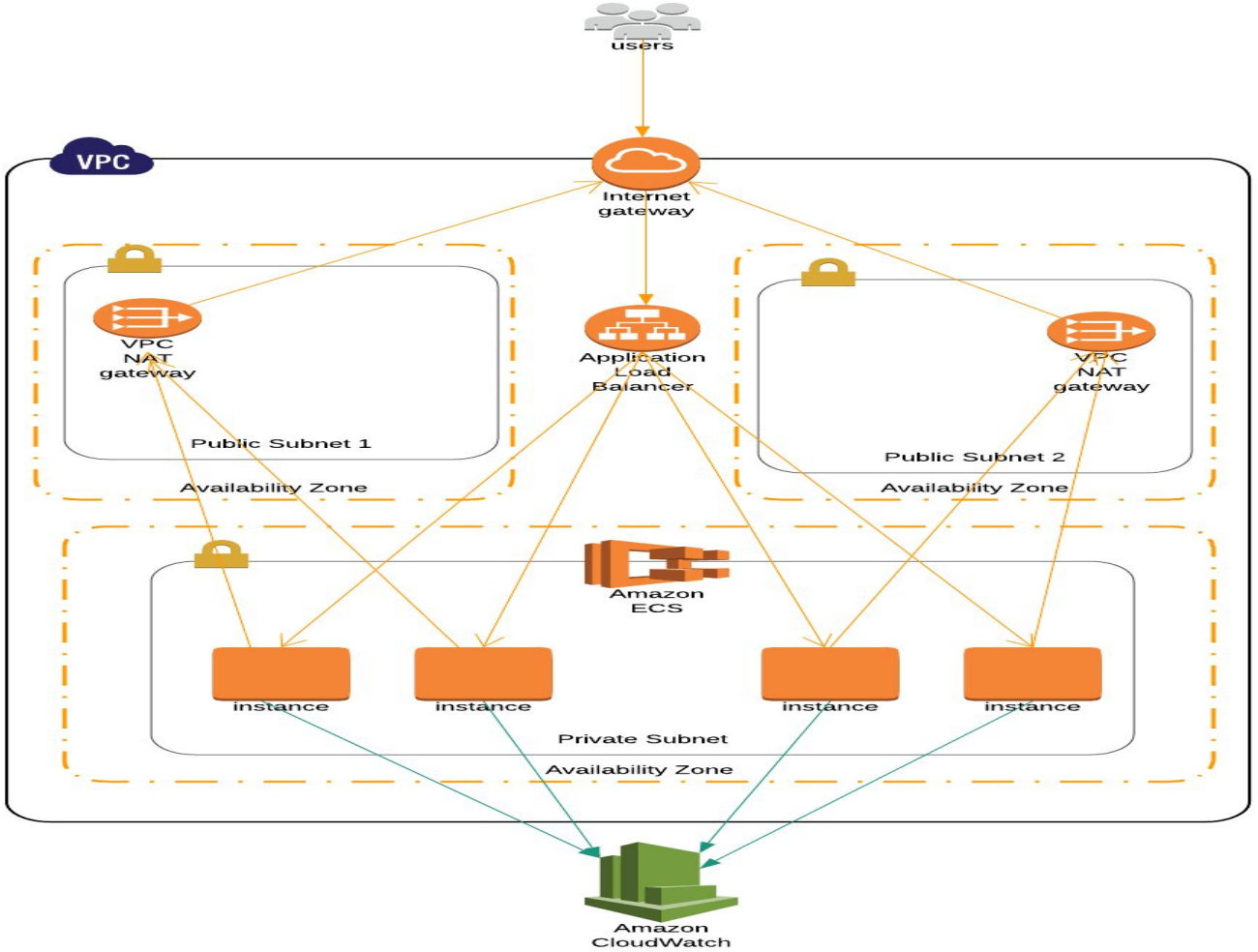
1) Isolated VPC for each Environments
2) Each VPC environments have a
    a) Public Subnet
    b) Private Subnet

3) A public facing ALB for accepting connections, ECS for container Deployment

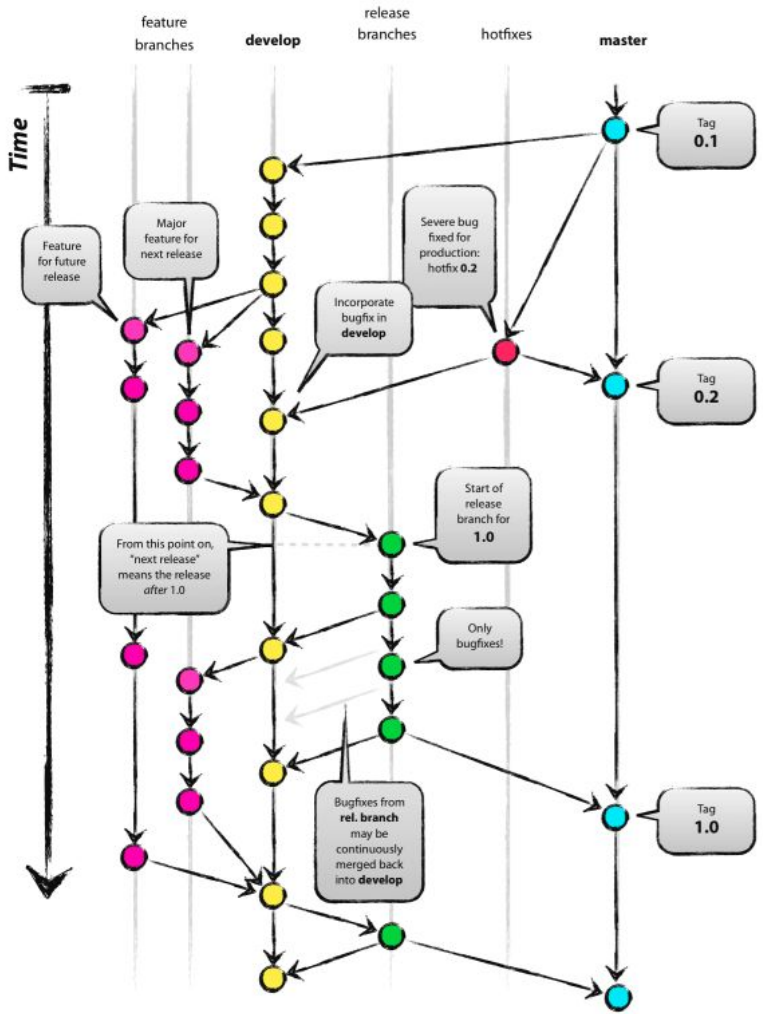
4) ALB in public Subnet, ECS in private

5) ALB routes traffic inside container

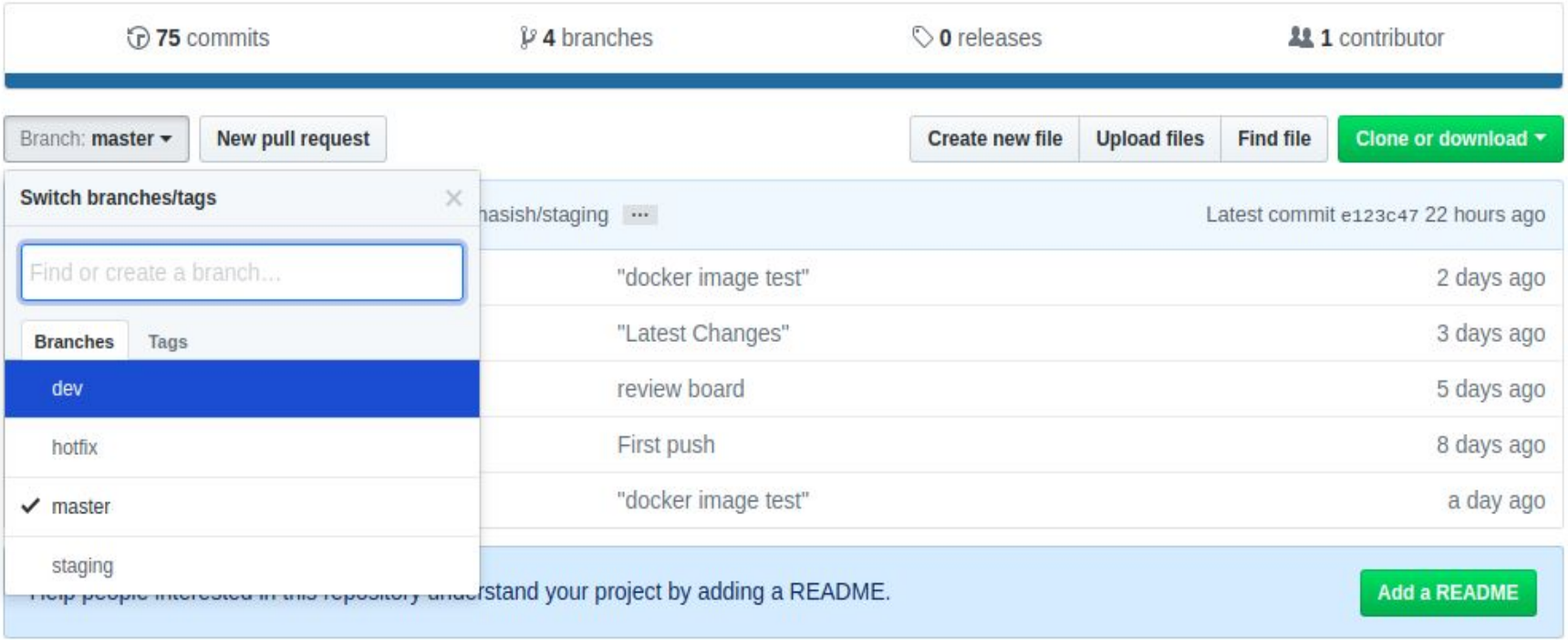
6) Jenkins and ReviewBoard in separate instances and in default VPC and subnets


users
VPC
Internet gateway
VPC NAT gateway
Application Load Balancer
VPC NAT gateway
Public Subnet 1
Availability Zone
Public Subnet 2
Availability Zone
Amazon ECS
instance
instance
instance
instance
Private Subnet
Availability Zone
Amazon CloudWatch

Branching


feature branches
develop
release branches
hotfixes
master
Time
Tag 0.1
Feature for future release
Major feature for next release
Severe bug fixed for production: hotfix 0.2
Incorporate bugfix in develop
Tag 0.2
Start of release branch for 1.0
From this point on, "next release" means the release after 1.0
Only bugfixes!
Bugfixes from rel. branch may be continuously merged back into develop
Tag 1.0

1) “dev” branch is used for deployment in “dev” environments
2) Feature branches branches out from “dev” and merge back into it
3) “Dev” branch is reviewed and merged to ‘staging’
4) “Staging” Branch is used for “staging” deployments for QA
5) “Staging” code is approved and merged into “master”
6) “Master” Branch is the stable branch and used for “production” deployment
7) “Hotfixes” branches out from master and merged back into “master”
8) Anything new branches out from master


75 commits
4 branches
0 releases
1 contributor
Branch: master
New pull request
Create new file
Upload files
Find file
Clone or download
Switch branches/tags
Find or create a branch...
Branches
Tags
dev
hotfix
master
staging
hasish/staging
Latest commit e123c47 22 hours ago
"docker image test"
2 days ago
"Latest Changes"
3 days ago
review board
5 days ago
First push
8 days ago
"docker image test"
a day ago
Add a README

Deployment Strategy

1) 3 different Environments
    a) Dev
    b) Staging
    c) Production
2) Dev Environment is the integration environment used by Developers and latest Developmental deployments are done here
3) Staging Environment is used by QA for testing and bug fix releases
4) Production is used by users and only the approved version or hotfixes are deployed here

dev >

FARGATE

0	0	0
Services	Running tasks	Pending tasks

EC2

0	0	0	No data	No data	1
Services	Running tasks	Pending tasks	CPUUtilization	MemoryUtilization	Container instances

production >

FARGATE

0	0	0
Services	Running tasks	Pending tasks

EC2

0	0	0	No data	No data	1
Services	Running tasks	Pending tasks	CPUUtilization	MemoryUtilization	Container instances

staging >

FARGATE

0	0	0
Services	Running tasks	Pending tasks

CI/CD

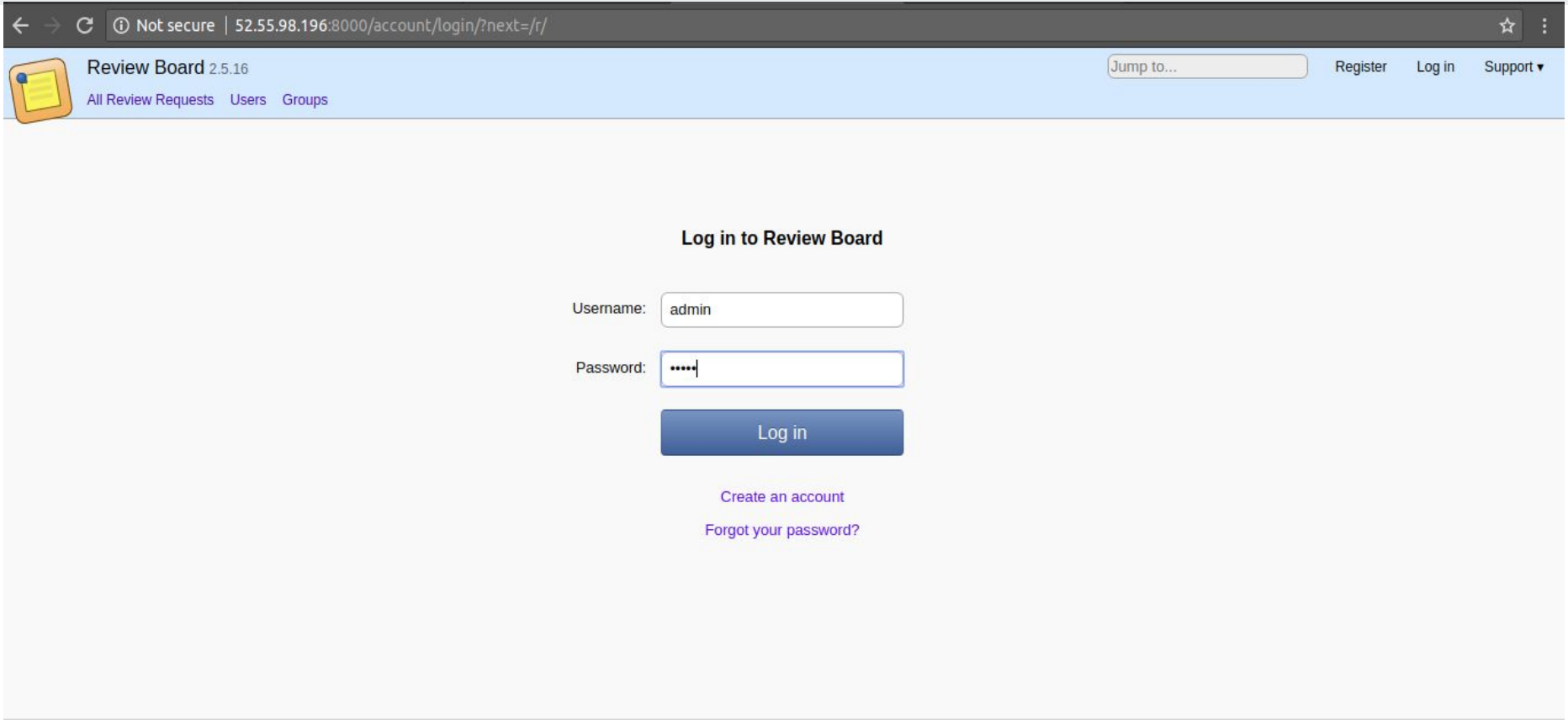
1) Jenkins is used for CI/CD
2) Each application is configured with jenkins as a “multibranch pipeline” job
3) Jobs are triggered by “web-hooks”
4) Each branch triggers it’s respective job
5) Deployment is done by combination of (Jenkins + Cloudformation)

Stage View

	Declarative: Checkout SCM	Build	Testing	Push	Deploy	Cleanup
Average stage times: (Average full run time: ~22s)	639ms	2s	5s	2s	6s	579ms
#11 Jan 01 17:09 — No Changes	320ms	325ms	5s	1s	6s	839ms
#10 Jan 01 17:05 — No Changes	610ms	333ms	5s	1s	5s	828ms
#9 Jan 01 17:01 — 1 commits	718ms	324ms	5s	1s	5s	841ms

Review Board

1) With each Commit and Push, a review request is created on the review board
2) The reviewer reviews the request and approves it
3) The approval process creates a pull request too


Not secure | 52.55.98.196:8000/account/login/?next=/r/
Review Board 2.5.16
All Review Requests
Users
Groups
Jump to...
Register
Log in
Support ▾
Log in to Review Board
Username:
admin
Password:
Log in
Create an account
Forgot your password?

Overall Process

1) Developers commit code to “dev” branch, a review request is created on “Review Board” and Jenkins “dev” job is triggered
2) The Job
    a) Checkout the code
    b) Docker Build the code
    c) Run a test container and test it
    d) Push the image to dockerhub
    e) Triggers deployment to “dev” environment
    f) Publish results to review board
3) Reviewer reviews the code and approves it
4) Approval process creates a pull request to “Staging” branch

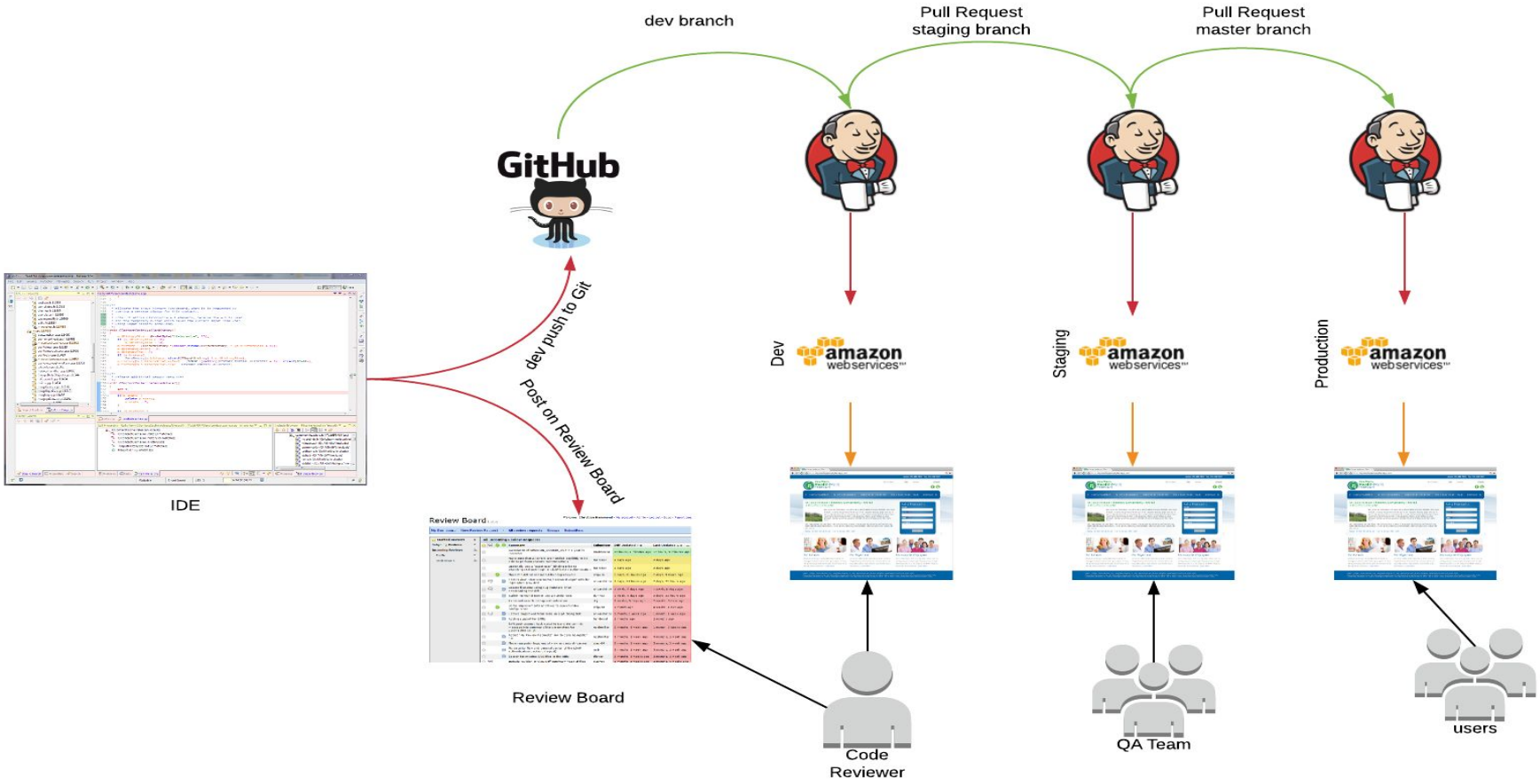
5) Code merged to “staging” branch

6) A jenkins job is triggered and follow the previous processes and deploy it to staging environment

7) QA do manual testing and approves the code and instigate over email or messaging

8) Staging Code is merged into master and it triggers a Jenkins job

9) The job deploy the application to production


dev branch
Pull Request
staging branch
Pull Request
master branch
GitHub
dev push to Git
Post on Review Board
IDE
Review Board
Review Board
Dev
Staging
Production
amazon
web services
Code
Reviewer
QA Team
users

Steps

Run the “setup.sh”, it will create the required infrastructures, A Jenkins server and a Review Board Server

```
subhasishp@subhashish-ub:~/techgig/techgig$ ./setup.sh
AWS CLI already present
Creating s3 deployment bucket
{
    "Location": "/techgig.infra"
}
Uploading Infrastructure cloudfomation scripts to bucket
Uploading ecs-cluster.yaml
upload: infrastructure/ecs-cluster.yaml to s3://techgig.infra/ecs-cluster.yaml
Uploading jenkins.yaml
upload: infrastructure/jenkins.yaml to s3://techgig.infra/jenkins.yaml
Uploading load-balancers.yaml
upload: infrastructure/load-balancers.yaml to s3://techgig.infra/load-balancers.yaml
Uploading master.yaml
upload: infrastructure/master.yaml to s3://techgig.infra/master.yaml
Uploading reviewboard.yaml
upload: infrastructure/reviewboard.yaml to s3://techgig.infra/reviewboard.yaml
Uploading security-groups.yaml
upload: infrastructure/security-groups.yaml to s3://techgig.infra/security-groups.yaml
Uploading vpc.yaml
upload: infrastructure/vpc.yaml to s3://techgig.infra/vpc.yaml
Deploying dev environment
{
    "StackId": "arn:aws:cloudformation:us-east-1:266394801394:stack/dev/04714e40-ec76-11e7-a127-500c28b27a35"
}
Deploying stagging environment
{
    "StackId": "arn:aws:cloudformation:us-east-1:266394801394:stack/staging/05663590-ec76-11e7-87dd-500c286f3262"
}
Deploying Production environment
{
    "StackId": "arn:aws:cloudformation:us-east-1:266394801394:stack/production/064b1750-ec76-11e7-8dbe-503aca2616c5"
}
Deploying Jenkins
{
    "StackId": "arn:aws:cloudformation:us-east-1:266394801394:stack/jenkins/07313190-ec76-11e7-9117-50d5ca632656"
}
```

Filter: Active ▾ By Stack Name

Showing 8 stacks

Stack Name	Created Time	Status	Description
infra-components-VPC-IAYEJ... NESTED	2018-01-01 15:53:38 UTC+0550	CREATE_IN_PROGRE...	This template deploys a VPC, with a pair of public and private subnets sprea...
production-VPC-R60WE1YY... NESTED	2018-01-01 15:53:37 UTC+0550	CREATE_IN_PROGRE...	This template deploys a VPC, with a pair of public and private subnets sprea...
staging-VPC-18X2OT814405K NESTED	2018-01-01 15:53:34 UTC+0550	CREATE_IN_PROGRE...	This template deploys a VPC, with a pair of public and private subnets sprea...
dev-VPC-1O3C6EON4EE63 NESTED	2018-01-01 15:53:34 UTC+0550	CREATE_IN_PROGRE...	This template deploys a VPC, with a pair of public and private subnets sprea...
infra-components	2018-01-01 15:53:32 UTC+0550	CREATE_IN_PROGRE...	This template deploys a VPC, with a pair of public and private subnets sprea...
production	2018-01-01 15:53:30 UTC+0550	CREATE_IN_PROGRE...	This template deploys a VPC, with a pair of public and private subnets sprea...
staging	2018-01-01 15:53:29 UTC+0550	CREATE_IN_PROGRE...	This template deploys a VPC, with a pair of public and private subnets sprea...
dev	2018-01-01 15:53:27 UTC+0550	CREATE_IN_PROGRE...	This template deploys a VPC, with a pair of public and private subnets sprea...

Filter: Active ▾ By Stack Name

Showing 19 stacks

Stack Name	Created Time	Status	Description
infra-components-ReviewBoa... NESTED	2018-01-01 16:03:10 UTC+0550	CREATE_COMPLETE	This cloudfomation script will create and configure an ec2 instances and r...
infra-components-Jenkins-AG... NESTED	2018-01-01 16:03:09 UTC+0550	CREATE_COMPLETE	This cloudfomation script will create and configure an ec2 instances and r...
infra-components-DefaultVP... NESTED	2018-01-01 16:01:28 UTC+0550	CREATE_COMPLETE	This template deploys a default VPC, with a default public subnet. It deplo...
infra-components	2018-01-01 16:01:23 UTC+0550	CREATE_COMPLETE	This template deploys a default VPC, a public subnet. It also deploys an In...
production-ECS-1KFZT13RI... NESTED	2018-01-01 16:00:27 UTC+0550	CREATE_COMPLETE	This template deploys an ECS cluster to the provided VPC and subnets u...
staging-ECS-1JLQ3F651HORA NESTED	2018-01-01 16:00:08 UTC+0550	CREATE_COMPLETE	This template deploys an ECS cluster to the provided VPC and subnets u...
dev-ECS-1TTW7HSFWQ259 NESTED	2018-01-01 15:59:59 UTC+0550	CREATE_COMPLETE	This template deploys an ECS cluster to the provided VPC and subnets u...
production-ALB-1ED9KJOG0... NESTED	2018-01-01 15:57:43 UTC+0550	CREATE_COMPLETE	This template deploys an Application Load Balancer that exposes our vari...
staging-ALB-NQ60PYXUPU19 NESTED	2018-01-01 15:57:39 UTC+0550	CREATE_COMPLETE	This template deploys an Application Load Balancer that exposes our vari...
dev-ALB-1GIWDDJ1LPUG3 NESTED	2018-01-01 15:57:38 UTC+0550	CREATE_COMPLETE	This template deploys an Application Load Balancer that exposes our vari...
production-SecurityGroups-1... NESTED	2018-01-01 15:57:16 UTC+0550	CREATE_COMPLETE	This template contains the security groups required by our entire stack. W...
staging-SecurityGroups-IFTK... NESTED	2018-01-01 15:57:13 UTC+0550	CREATE_COMPLETE	This template contains the security groups required by our entire stack. W...
dev-SecurityGroups-1JUTAL... NESTED	2018-01-01 15:57:12 UTC+0550	CREATE_COMPLETE	This template contains the security groups required by our entire stack. W...
production-VPC-R60WE1YY... NESTED	2018-01-01 15:53:37 UTC+0550	CREATE_COMPLETE	This template deploys a VPC, with a pair of public and private subnets spr...
staging-VPC-18X2OT814405K NESTED	2018-01-01 15:53:34 UTC+0550	CREATE_COMPLETE	This template deploys a VPC, with a pair of public and private subnets spr...

Github and Dockerhub

1) Create a new “github” repository and a “dockerhub” repository
2) The name of the github repository and the docker-hub repository must be same
3) Creating “Docker-Hub” registry is optional, it will get created automatically

shubhasish / techgig_service

Unwatch 1 | Star 0 | Fork 0

Code | Issues 0 | Pull requests 0 | Projects 0 | Wiki | Insights | Settings

No description, website, or topics provided.

Edit

Add topics

82 commits | 5 branches | 0 releases | 1 contributor

Branch: master | New pull request | Create new file | Upload files | Find file | Clone or download

shubhasish Merge pull request #8 from shubhasish/staging ... Latest commit d8f2cf9 6 hours ago

deployment	Merge branch 'staging' into dev	6 hours ago
helloWorld	"Latest Changes"	3 days ago
.reviewboardrc	review board	5 days ago
Dockerfile	First push	8 days ago
Jenkinsfile	"docker image test"	a day ago

Help people interested in this repository understand your project by adding a README. Add a README

PUBLIC | AUTOMATED BUILD

shubhashish/helloworld ☆

Last pushed: 5 hours ago

Repo Info | Tags | Dockerfile | Build Details | Build Settings | Collaborators | Webhooks | Settings

Short Description

Creating Automated Build

Full Description

codegladiator

codeGladiatorCode

Docker Pull Command

```
docker pull shubhashish/helloworld
```

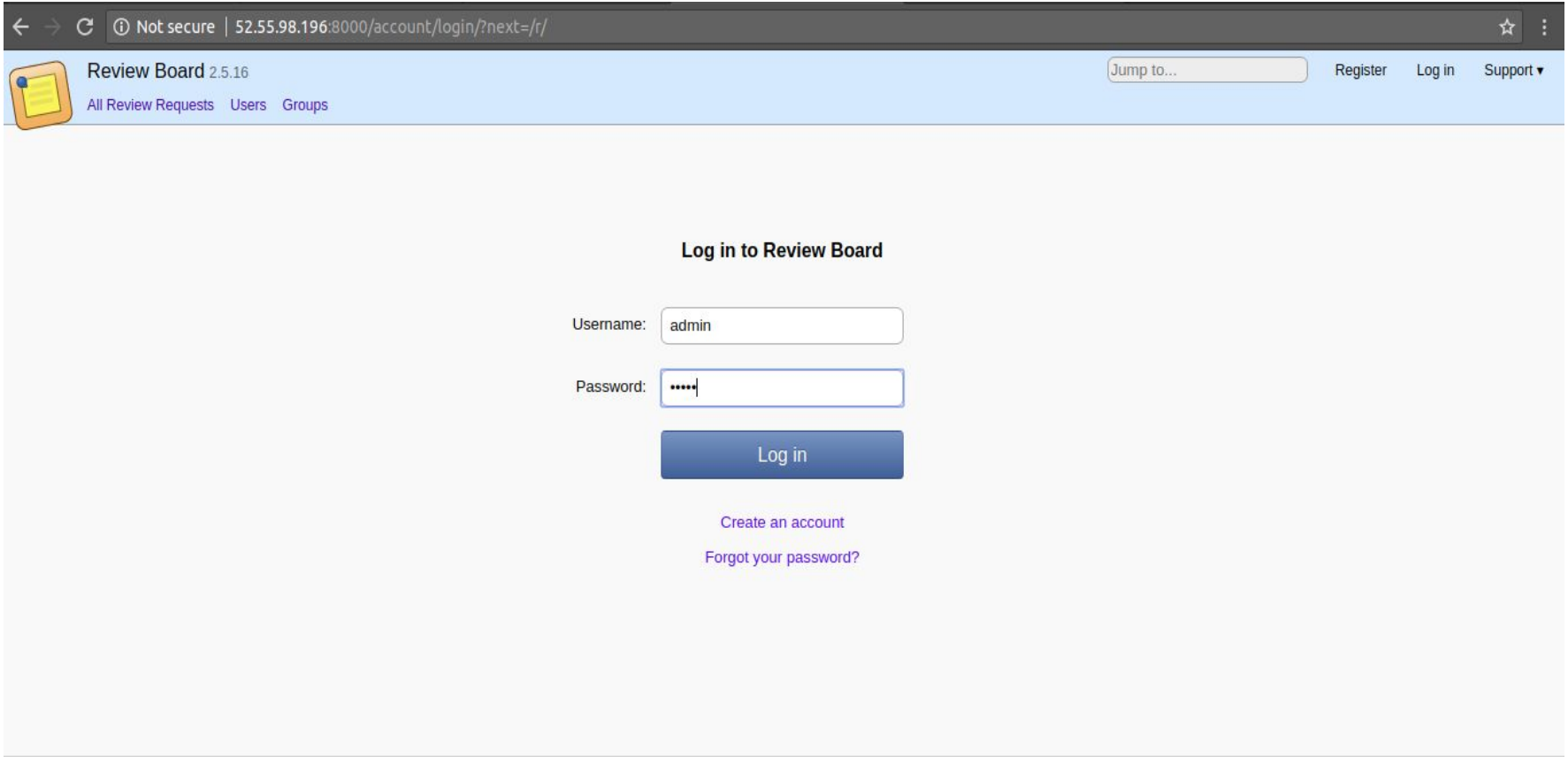
Owner

shubhashish

Source Repository

Review Board

1) Open review Board Server (IP:8000)
2) Login as admin
3) Add the repository
4) Install RBTools on your local (pip install -U RBTools)
5) Add the “.reviewboardrc” and a “post-commit” script to your application git folder
6) Whenever a dev push a commit, a review is created on “Review Board”


Not secure | 52.55.98.196:8000/account/login/?next=/r/
Review Board 2.5.16
All Review Requests
Users
Groups
Jump to...
Register
Log in
Support ▾
Log in to Review Board
Username:
admin
Password:
Log in
Create an account
Forgot your password?

Authentication
E-Mail
Diff Viewer
Logging
SSH
File Storage
Support

MANAGE

Users 1 Add
Review Groups 0 Add
Default Reviewers 0 Add
Repositories 0 Add
Webhooks 0 Add
Hosting Accounts 0 Add

SYSTEM INFORMATION

Server Cache
Public Read-only Access
Syntax Highlighting
Server Log
Log Profiling
Review E-mails
E-mail TLS Authentication
Indexed Search
Security Checklist

GENERAL INFORMATION

Name: * techgig_service

Show this repository

Use this to control whether or not a repository is shown when creating new review requests. Existing review requests are unaffected.

REPOSITORY HOSTING

Hosting service: * GitHub

Account: * <Link a new account>

Link this repository to an account on the hosting service. This username may be used as part of the repository URL, depending on the hosting service and plan.

Account username: * shubhasish

Account password: * ••••••••••••

REPOSITORY INFORMATION

Repository type: * Git

Repository plan: * Public

The plan for your repository on this hosting service. This must match what is set for your repository.

Repository name: * techgig_service

The name of the repository. This is the <repo_name> in http://github.com/<username>/<repo_name>/

BUG TRACKER

Use hosting service's bug tracker

Type: * (None)

ACCESS CONTROL

Publicly accessible

Review requests and files on public repositories are visible to anyone. Private repositories must explicitly list the users and groups that can access them.

.reviewboardrc

Revision ef41d548a64209be489061d2621ea6d1204d6525

```
1 REVIEWBOARD_URL="http://localhost:8000/"
2 REPOSITORY="techgig_service"
```

New Change

```
1 REVIEWBOARD_URL="http://52.55.98.196:8000/"
2 REPOSITORY="techgig_service"
```

helloWorld/handlerClass.py

Revision fc8086023ad3162116a993fbd95b26654bfd173b

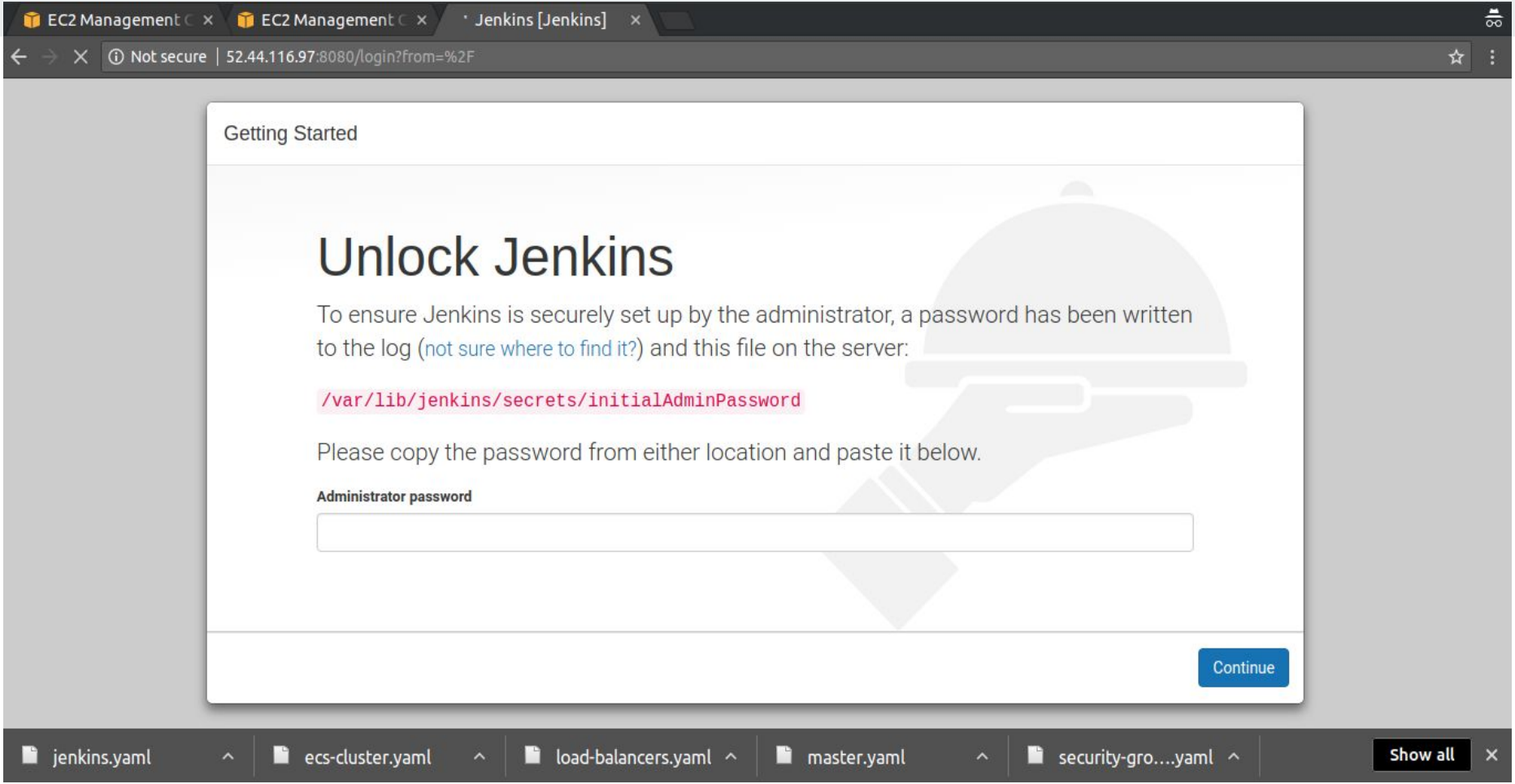
```
1  import flask
2  from flask_restful import Resource
3
4
5  class HelloWorld(Resource):
6
7      def get(self):
8          return flask.jsonify({"Success": "true", "Message": "Hello World, we are testing a blue-green deployment."})
9
10
11
12 class HealthCheck(Resource):
13     def get(self):
14         return "O.K"
```

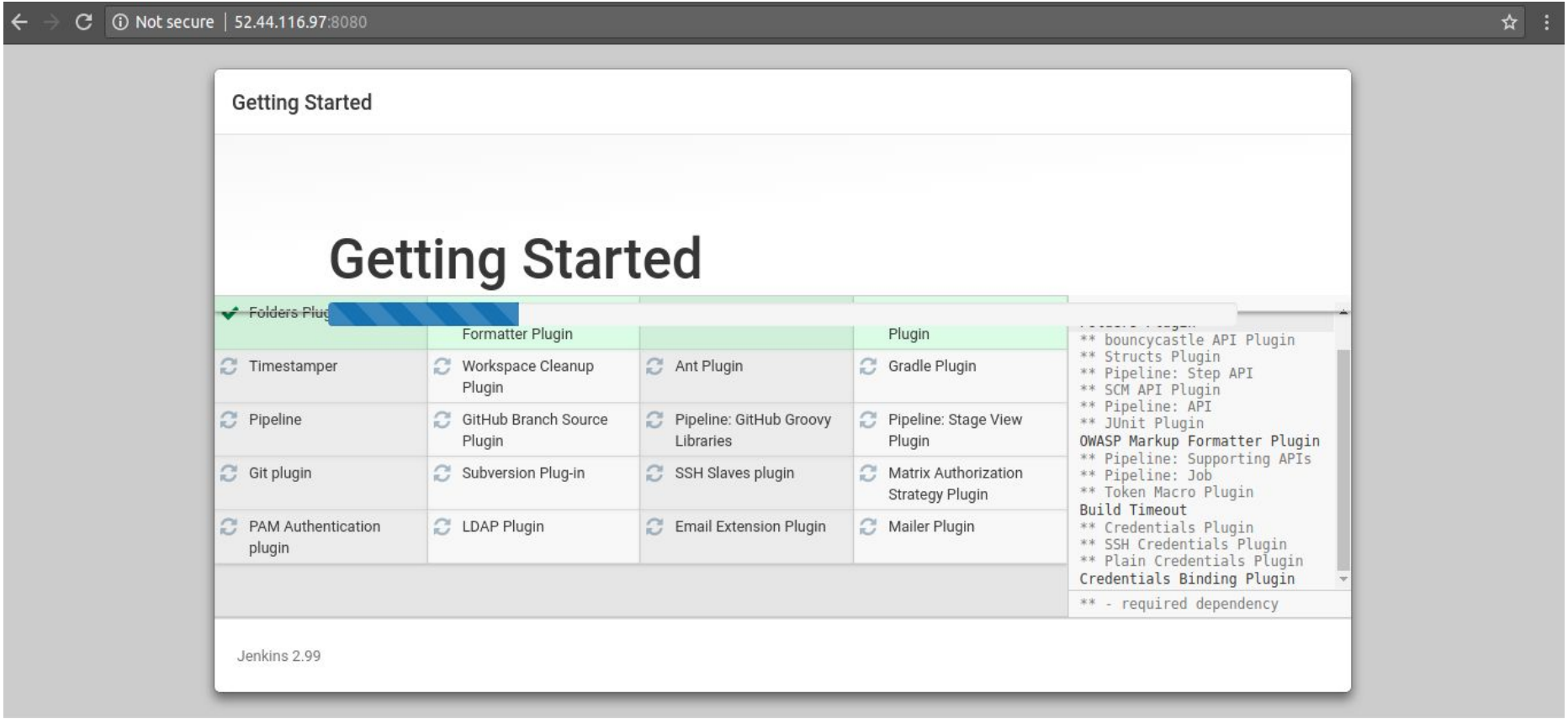
New Change

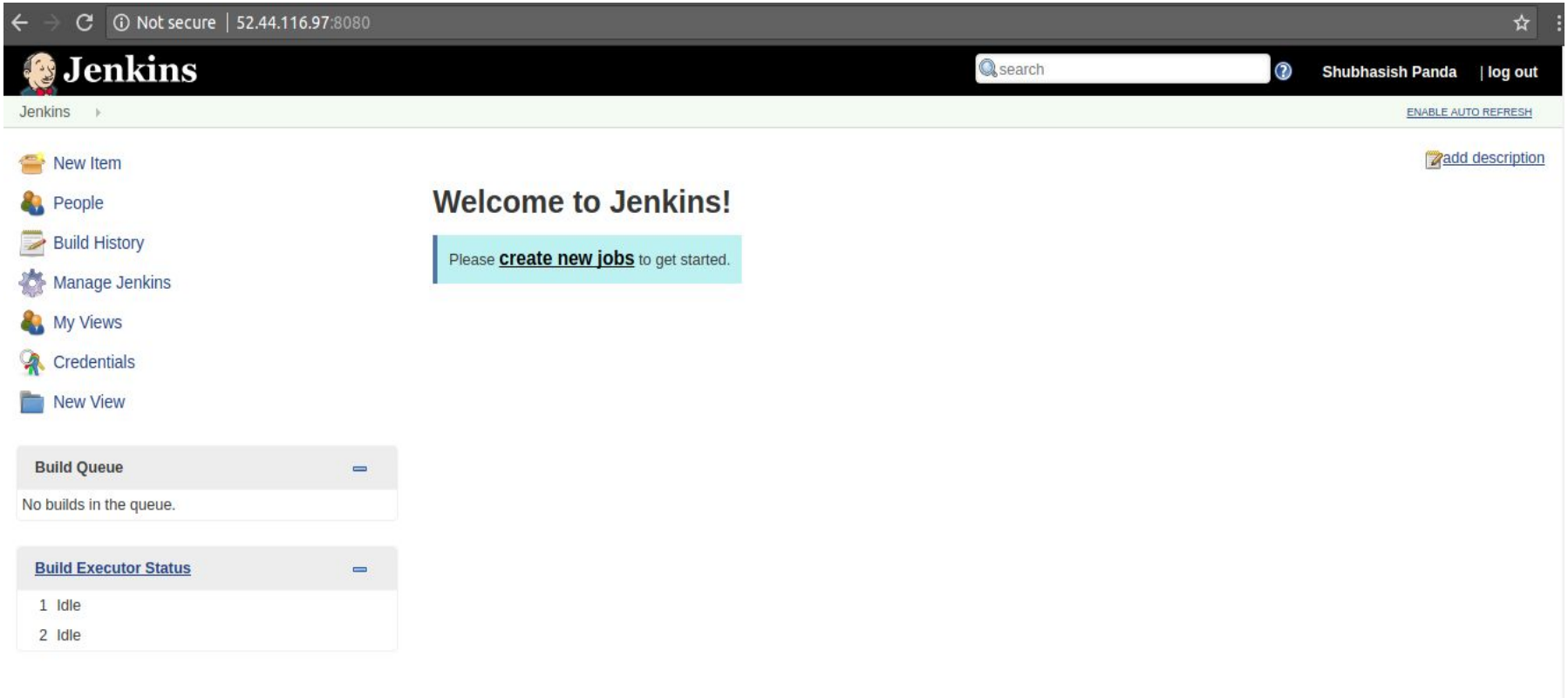
```
1  import flask
2  from flask_restful import Resource
3
4
5  class HelloWorld(Resource):
6
7      def get(self):
8          return flask.jsonify({"Success": "true", "Message": "Hello World, we are testing reviewboard."})
9
10
11
12 class HealthCheck(Resource):
13     def get(self):
14         return "O.K"
```

Jenkins

1) Open the Jenkins
2) Do the first time configuration
3) Add credentials
    a) Github (github_id)
    b) Docker-Hub (dockerhub_id)
    c) AWS (access key and id) (aws_id)
4) Select “Create Job” and select “multibranch-pipleline” as job configuration
5) Add the “Github” Source


EC2 Management
EC2 Management
Jenkins [Jenkins]
Not secure | 52.44.116.97:8080/login?from=%2F
Getting Started
Unlock Jenkins
To ensure Jenkins is securely set up by the administrator, a password has been written to the log (not sure where to find it?) and this file on the server:
/var/lib/jenkins/secrets/initialAdminPassword
Please copy the password from either location and paste it below.
Administrator password
Continue
jenkins.yaml
ecs-cluster.yaml
load-balancers.yaml
master.yaml
security-gro....yaml
Show all


Not secure | 52.44.116.97:8080
Getting Started
Getting Started
Folders Plug
Formatter Plugin
Plugin
Timestamper
Workspace Cleanup Plugin
Ant Plugin
Gradle Plugin
Pipeline
GitHub Branch Source Plugin
Pipeline: GitHub Groovy Libraries
Pipeline: Stage View Plugin
Git plugin
Subversion Plug-in
SSH Slaves plugin
Matrix Authorization Strategy Plugin
PAM Authentication plugin
LDAP Plugin
Email Extension Plugin
Mailer Plugin
** bouncycastle API Plugin
** Structs Plugin
** Pipeline: Step API
** SCM API Plugin
** Pipeline: API
** JUnit Plugin
OWASP Markup Formatter Plugin
** Pipeline: Supporting APIs
** Pipeline: Job
** Token Macro Plugin
Build Timeout
** Credentials Plugin
** SSH Credentials Plugin
** Plain Credentials Plugin
Credentials Binding Plugin
** - required dependency
Jenkins 2.99


Not secure | 52.44.116.97:8080
Jenkins
search
Shubhasish Panda
| log out
Jenkins
ENABLE AUTO REFRESH
New Item
People
Build History
Manage Jenkins
My Views
Credentials
New View
add description
Welcome to Jenkins!
Please create new jobs to get started.
Build Queue
No builds in the queue.
Build Executor Status
1 Idle
2 Idle

Back to credential domains
Add Credentials

Global credentials (unrestricted)

Credentials that should be available irrespective of domain specification to requirements matching.

Name	Kind	Description
This credential domain is empty. How about adding some credentials?		

Icon: S M L

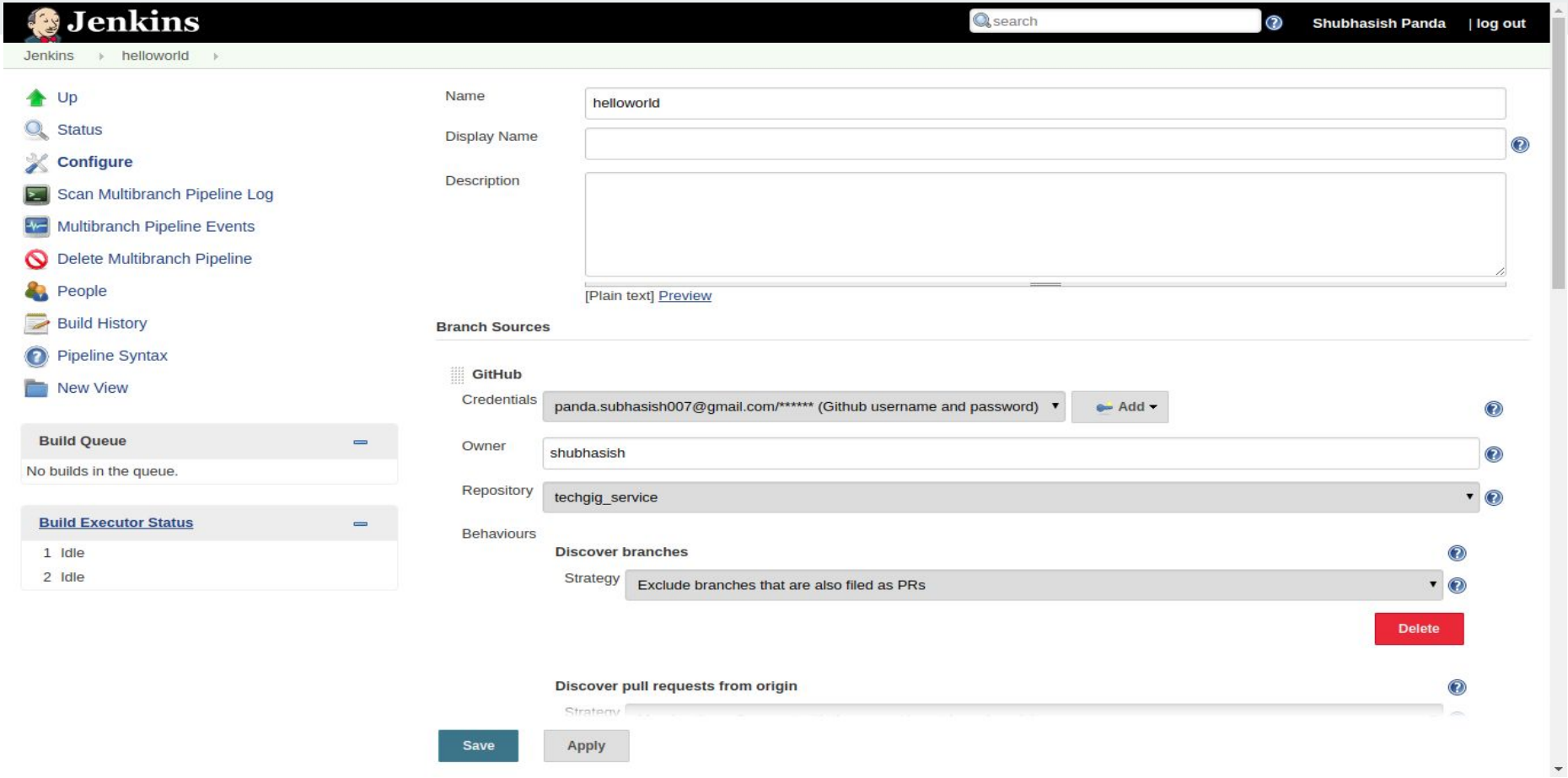
Back to credential domains
Add Credentials

Global credentials (unrestricted)

Credentials that should be available irrespective of domain specification to requirements matching.

Name	Kind	Description
panda.subhasish007@gmail.com/****** (Github username and password)	Username with password	Github username and password
shubhashish/****** (docker hub username and password)	Username with password	docker hub username and password
AKIAILCXOOYUK6N2BTBA/****** (AWS access-key and id)	Username with password	AWS access-key and id

Icon: S M L


Jenkins
search
Shubhasish Panda
| log out
Jenkins
helloworld
Up
Status
Configure
Scan Multibranch Pipeline Log
Multibranch Pipeline Events
Delete Multibranch Pipeline
People
Build History
Pipeline Syntax
New View
Build Queue
No builds in the queue.
Build Executor Status
1 Idle
2 Idle
Name
helloworld
Display Name
Description
[Plain text] Preview
Branch Sources
GitHub
Credentials
panda.subhasish007@gmail.com/****** (Github username and password)
Add
Owner
shubhasish
Repository
techgig_service
Behaviours
Discover branches
Strategy
Exclude branches that are also filed as PRs
Delete
Discover pull requests from origin
Strategy
Save
Apply

helloworld

Branches (5) | Pull Requests (0)

S	W	Name ↓	Last Success	Last Failure	Last Duration	
		dev	54 min - #11	1 hr 18 min - #5	17 sec	
		hotfix	N/A	1 hr 34 min - #1	14 sec	
		**master**	46 min - #2	1 hr 34 min - #1	16 sec	
		staging	46 min - #3	1 hr 34 min - #1	15 sec	
		test	N/A	58 min - #1	3.9 sec	

Icon: S M L

Legend | RSS for all | RSS for failures | RSS for just latest builds

That’s It !!! You are ready to go

CI/CD Process

1) Dev Commits code, review request created and application deployed in dev environment
2) Dev code merged with staging, jenkins job triggered, application is built and deployed to staging
3) QA done, deployment informed via mail, staging merged with master, jenkins job run. The application is built, test and deployed to prod

dev-helloworld

Stack name:	dev-helloworld
Stack ID:	arn:aws:cloudformation:us-east-1:266394801394:stack/dev-helloworld/83eb51b0-eee8-11e7-965f-500c219a3c36
Status:	CREATE_COMPLETE
Status reason:	
Termination protection:	Disabled
IAM role:	
Description	This is an example of a long running ECS service that serves a JSON API of products.


```
{
  "Message": "Hello World, we are testing a blue-green deployment.",
  "Success": "true"
}
```

Open **shubhasish** wants to merge 5 commits into `staging` from `dev`

Conversation 0 | Commits 5 | Files changed 2

shubhasish commented 34 seconds ago — Owner

No description provided.

Shubhasish and others added some commits 33 minutes ago

- "Latest Changes" ✗ 33183ad
- "Latest Changes" ✓ f85e49b
- "Latest Changes" ✓ 51f7d90
- "Latest Changes" ✓ d6e1227
- Merge branch 'staging' into dev — Verified — a8d94e3

Add more commits by pushing to the **dev** branch on **shubhasish/techgig_service**.

This branch has no conflicts with the base branch

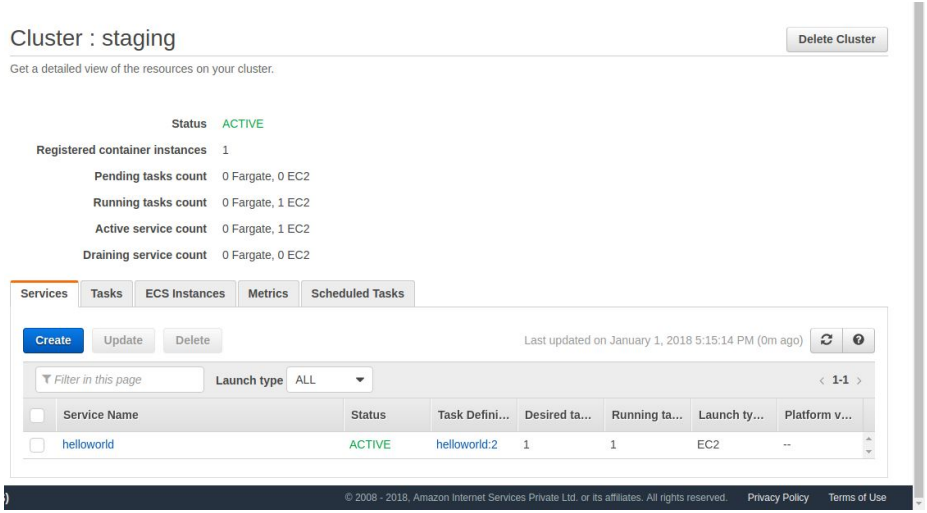
Merging can be performed automatically.

Merge pull request | or view command line instructions.

staging-helloworld

Other Actions ▾ | Update Stack

Stack name:	staging-helloworld
Stack ID:	arn:aws:cloudformation:us-east-1:266394801394:stack/staging-helloworld/225fbde0-eee9-11e7-804c-50fae98a10d2
Status:	CREATE_COMPLETE
Status reason:	
Termination protection:	Disabled
IAM role:	
Description	This is an example of a long running ECS service that serves a JSON API of products.



Staging #8

Open **shubhasish** wants to merge 6 commits into `master` from `staging`

Conversation 0 | Commits 6 | Files changed 2

shubhasish commented just now — Owner

No description provided.

Shubhasish and others added some commits 36 minutes ago

"Latest Changes"	✗	33183ad
"Latest Changes"	✓	f85e49b
"Latest Changes"	✓	51f7d90
"Latest Changes"	✓	d6e1227
Merge branch 'staging' into dev	Verified	a8d94e3
Merge pull request #7 from shubhasish/dev ···	Verified ✓	b9d0d50

Add more commits by pushing to the **staging** branch on **shubhasish/techgig_service**.

All checks have passed — Show all checks

1 successful check

This branch has no conflicts with the base branch

Merging can be performed automatically.

Merge pull request ▾ or view command line instructions.

production-helloworld

Stack name:	production-helloworld
Stack ID:	arn:aws:cloudformation:us-east-1:266394801394:stack/production-helloworld/afc6ee10-eee9-11e7-b688-500c286374d1
Status:	CREATE_COMPLETE
Status reason:	
Termination protection:	Disabled
IAM role:	
Description	This is an example of a long running ECS service that serves a JSON API of products.

Cluster : production

Delete Cluster

Get a detailed view of the resources on your cluster.

Status	ACTIVE
Registered container instances	1
Pending tasks count	0 Fargate, 0 EC2
Running tasks count	0 Fargate, 1 EC2
Active service count	0 Fargate, 1 EC2
Draining service count	0 Fargate, 0 EC2

Services | Tasks | ECS Instances | Metrics | Scheduled Tasks

Create | Update | Delete

Last updated on January 1, 2018 5:19:52 PM (0m ago)

Filter in this page | Launch type ALL | 1-1

Service Name	Status	Task Defini...	Desired ta...	Running ta...	Launch ty...	Platform v...
helloworld	ACTIVE	helloworld:4	1	1	EC2	--


```
{
  "Message": "Hello World, we are testing a blue-green deployment.",
  "Success": "true"
}
```

Extent of Automation

1) In my scripts i have tried to keep manual intervention as low as possible
2) The scripts are generic hence extensible and scalable
3) The infrastructure is a plug and play model, new models can be introduced with ease
4) Only manual intervention required is to configure credentials, pem files and webhook
5) A bare bone infrastructure to play with can be set up in just matter of 30 mins-1hr
6) And a full automated Infrastructure can be created within 1hr-1.5 hrs

To-Do

Complete Automation integrating webhooks and commit hooks

Thanks