

Deploy new Gitlab Runner

Gitlab Runner is used by the application teams to run the CI/CD jobs. At the core, this is an EC2 instance which runs application's CI/CD jobs on docker containers.

We recommend all the application teams to share one gitlab runner (tag: **bcp-generic-gitlab-runner**). This helps the cloud engineers to ease the deployment/management process for the new applications and also keeps the infrastructure costs low.

That said, there may be numerous instances where cloud engineers need to spin a new gitlab runner. Please follow the steps below to spin up a new runner.

This runbook assumes that Gitlab Runner deployment is to be done for spiking purpose, hence the Gitlab Group used is <https://gitlab.com/mc-estate-infra/spikes> and Terraform organization used is <https://app.terraform.io/app/mc-spikes>. Please replace these values respectively for the specific use case.

Deployment

Gitlab runners are **deployed in Devops AWS account** because it is a part of CI/CD for applications.

1. Blueprint to deploy the Gitlab runner is present here: <https://gitlab.com/mc-estate-infra/blueprints-repo/caas/caas-generic-gitlab-runner>
2. Clone this blueprint code into the respective gitlab group. E.g. for spiking purpose, create a new project under [Spikes gitlab group](#) and push the Gitlab Runner blueprint code there.
3. Create a new Terraform Workspace under <https://app.terraform.io/app/mc-spikes> and point it directly to the new Gitlab Project created in Step 2.
4. Fill in the Terraform variables in the Terraform workspace created in step 3: https://app.terraform.io/app/mc-spikes/workspaces/<your_workspace_name>/variables

Required Terraform Variables

Terraform variables are required to tweak the deployment of the Gitlab Runners. Please add the variables below and their recommended values as given. Feel free to change the values if need be:

Variable name	How to fetch the variable value
base_name	Base name is a string which signifies the purpose for this gitlab runner. E.g. if this runner is for a use case like paspo then this could be "paspo"
devops_aws_account	Devops AWS account credentials. These need to be fetched from the terraform workspace . Please ask @Deepa (Deactivated) to fetch it for you
spoke_aws_accounts	This includes all the AWS accounts where we need Gitlab runner to pick the jobs from. Choose and pick the account_ids and alias from Estate-base Terraform workspace
devops_vpc_id and devops_private_subnet_ids	Devops VPC id and subnet ids can be picked from the state file of Estate base Terraform workspace
gitlab_runner_registration_token	<p>Depends on the Group at which we want Gitlab runner to be available. Normally it is a best practice to make gitlab runner available at https://gitlab.com/min-au-apps group level so that any project inside it can use the runner.</p> <p>Once the group is decided, just go to <code>settings CI/CD Runners</code> and use the runner registration token present there.</p>

custom_tags

Mostly constant. The current tags used are

```
{
  BusinessUnit="Cybernetics"
  CostCenter="Wbs.1410.0.80019.1000"
  Env="prd"
  Owner="sean.cole@bhp.com"
  Secondary="paul.manning@bhp.com"
  DeploymentType="Automated"
  AppName="minerals.cloud"
}
```

1. If the plan looks good, click on apply.
2. Once the Terraform apply is green, check that the gitlab runner shows as online in Gitlab.

