

**\*This is still work in progress.**

## Index

1. Introduction
2. Pre-requisites
3. Configurations
4. Scripts
5. Run
6. Know issues

## [1.0] Introduction

Terraform + Ansible playbook to automate resource creation, provisioning, configuration with Docker and setting up Blazemeter crane docker container. The scripts will also make sure that the agent is created within your private location.

This project is based on setting up Private-location agents for running tests through Blazemeter. There are documented manual steps to help create a Blazemeter Private-location. However, the manual workload can be avoided using these scripts.

Likewise, using the basic REST APIs we can achieve similar results but it will be limited to a specific host. Again, if you want to spin up multiple Blazemeter agents automatically you can make use of these scripts. The same API calls are transformed into YAML to run through Ansible playbook, which can help us configure multiple Blazemeter private-location agents.

Below are the list of things these scripts will do:

1. Create EC2 VM in the AWS (Make changes if you have a different cloud provider) (see [4.1])
2. Create an agent into Blazemeter SaaS account.
3. Generate a docker command for Blazemeter-crane
4. Pull and run Blazemeter-crane docker container.

Within 10-15 mins, you will have your desired number of agents running, skipping all manual work.

## [2.0] Pre-requisites

- Blazemeter account (Admin access)
- Linux x86\_64 (Mainly Ubuntu or RedHat based distros)
- Cloud provider (If the private location and agents are to be setup on cloud instances)
- Below are the packages we need to available in your control node.

- **Python**

- \$ python3 --version

- **Terraform**

- \$ terraform --version

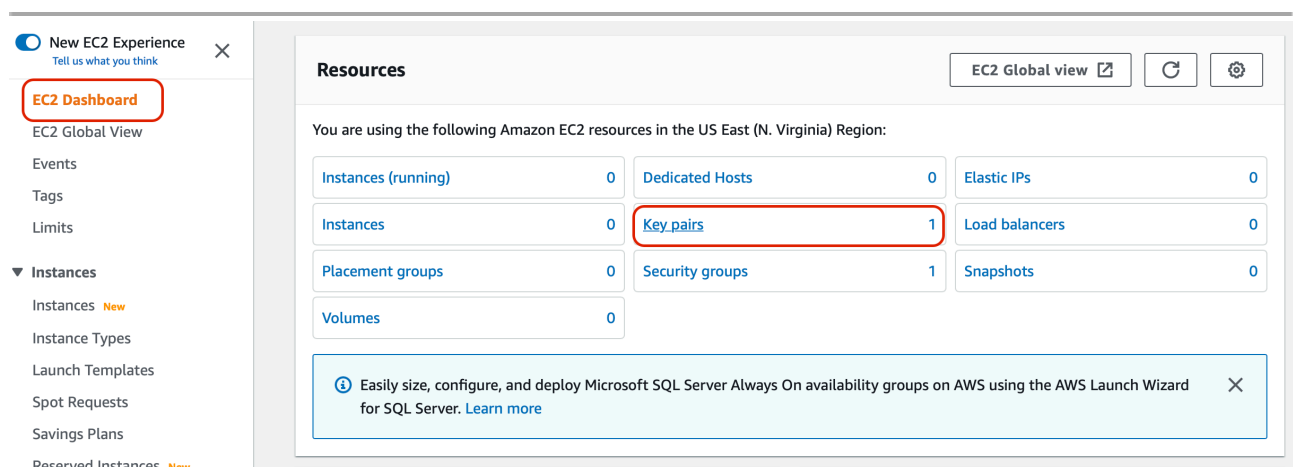
- **Ansible**

- \$ ansible --version

## [3.0] Configurations

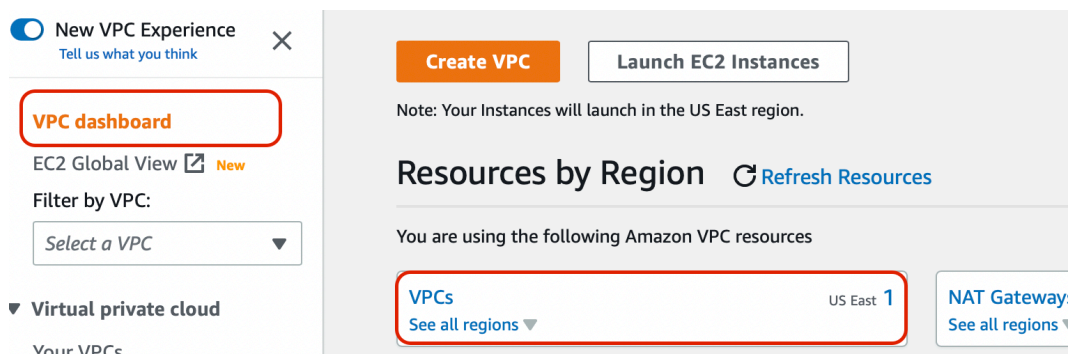
### [3.1] AWS keys (Refer to the figure below)

1. Login to AWS account, or create one if not available.
2. Navigate to EC2 section > under the Network & Security select Key pairs, or key pairs section displayed on the EC2 Dashboard
3. This will open key pair section, all the created keys can be found here.
4. Click on create key pair to create a new key pair (see the figure below)
5. Name the key pair, select .pem format and RSA as key type. (**Setting bmkey as the key name here**)
6. Click on create key pair to create it and the system will automatically download the key pair.
7. Change the permission of the key using `$ chmod 400 [key-path]`



### [3.2] VPC ID (AWS)

1. Login to AWS account,
2. Navigate to VPC section > click on VPCs section on the VPC dashboard to open the VPC panel.
3. Simply copy the VPC ID from here.



---

### [3.3] AWS or Cloud provider (IAM) api keys

If you are planning to utilise the terraform script to create a VM within your cloud, then there is a need to provide the keys (IAM section in AWS) into the script to provision a VM.

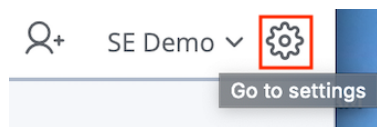
Make sure you go through your Cloud provider documentation to create these key pairs and extract for further use.

- Access key
- Secret key
- Secret token

---

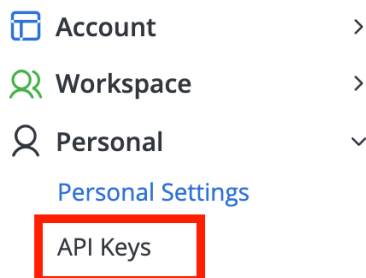
### [3.4] Blazemeter API key creation

Go to the user drop-down menu at the top right corner of the screen, then click on 'Settings'. You will be transferred to the 'Settings' Panel.

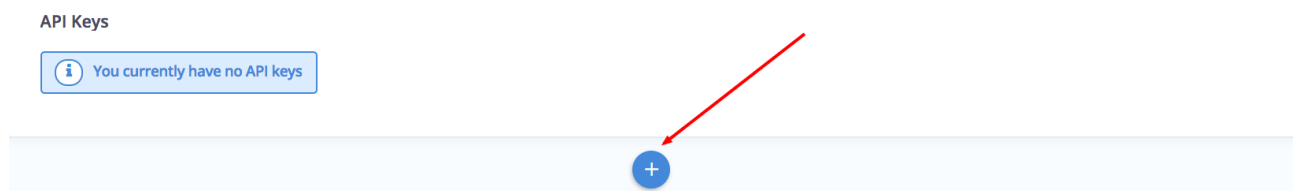


You will be able to see the 'API keys' option under the 'Personal' settings on the left of the screen.

#### Settings



Here, click on the '+' icon to create a new API key.



Then, give your API key a relevant name that is useful for your reference, and set its expiration period.

Upon clicking the "Generate" button, your new API Key ID and API Key Secret will be displayed.

## Generate API Key

Name:

Expiry:

API Key Id:

API Key Secret:

[Download as JSON](#)



Make sure to copy your new API key secret now. You won't be able to see it again!

WARNING!!! Your API Key Secret ("secret key") will only be shown this one time and can never be retrieved again. Make sure to copy the secret key before closing the window. If the secret key is lost, it must be regenerated.

See : [Blazemeter guide here.](#)

\*Note: These details will be required in vars.yml file (see [4.3])

---

## [3.5] Blazemeter Harbour ID

The Harbor ID and Ship ID can be found in the BlazeMeter Settings -> Workspaces -> Private Locations -> <Your Private Location>.

Harbor ID: Located under Private Location Details, under the Id column (indicated by the red arrow).

See [Blazemeter guide here.](#)

\*Note: These details will be required in vars.yml file (see [4.3])

## [4.0] Scripts [guide to making changes]

Make appropriate changes to the script to fit your organisational policy or individual goals.

---

### [4.1] Terraform - main.tf

The terraform script has the highest scope of amendments as it will be used to create a security group, provision an Instance and setup security and network policies.

Add local variables, these will be used throughout the script - ami-id of the instance, use VPC ID (see [3.2]), key\_name (see [3.1]) and key\_path.

Declare the provider and other required information linked with it, access key, secret key and token as per AWS (Or any cloud provider you are using) (See [3.3])

Now within the security group creation, make sure to setup a correct network policy. I have kept this very open with rules that I have setup. However, these tend to change when instance is being created on an enterprise level.

It is possible to write a different terraform script entirely, however, make sure to record the private IPs of the created instance into a file (using >>, >, etc.) (So that it can be later used as an inventory file)

---

### [4.2] Ansible - bm-engine.yml

This is a very simple yaml file, invoking other yaml files based on the order of these tasks and facts gathered initially.

The file will also declare the hosts and a variable file.

Moreover based on the OS of the VM, it will call the playbook to setup Docker.

The file can be run directly, If you want to bypass terraform- instance creation, considering that you already have an instance/s ready. (See [5.0])

---

### [4.3] Ansible - default/vars.yml

This is a var file, as the name suggests. Here we will need to add the variables as per our Blazemeter account. API keys and secret. (see [3.4]) and Harbour\_ID (see [3.5])

---

### [4.4] Ansible - plays/docker-ubuntu.yml & plays/docker-rhel.yml

File that will install Docker, into Ubuntu as well as RedHat based system. The YAML will make sure that the docker is installed and is up to date.

This file can be ignored if Docker is already setup on your machine. However, if creating a fresh VM, this is recommended, as docker may not be available by default.

Moreover, the bm-agent.yml file is triggered through these files, hence ignoring this file will skip the bm-agent.yml file (Which is the main file in this configuration)

Therefore, if you ignore this file, make sure to run the bm-agent.yml file separately (In that case you will need to trigger the file through the bm-engine.yml).

---

## [4.5] Ansible - plays/bm-agent.yml

This is the main file, responsible for creating a Blazemeter agent on SaaS and pulling + running the docker container (Blazemeter-crane) within our Machine (VM/EC2 or any remote machine on cloud.). The file will also print out the docker command, in case you wish to make additional changes to the docker command. If none, the file will automatically run the generated command.

\*Note: This file need not be modified. If modification is required, make the changes carefully.

## [5.0] Run

All well set, ready to apply this.

If you are want to use all the features, terraform resource creation and ansible configuration, use these commands.

```
$ terraform init
$ terraform validate
$ terraform plan
$ terraform apply
```

If you already have the machines running, you can run the ansible-playbook.

```
$ ansible-playbook -i [inventory_file] --user [username] --private-key [key_path] bm-engine.yml
```

Now, wait and let the script do it's thing.

## [6.0] Known issues

---

### [6.1] Issues with CentOS

Using CentOS you may see errors like

```
FAILED! => {"changed": false, "msg": "Failed to download metadata for repo 'appstream': Cannot prepare internal mirrorlist: No URLs in mirrorlist", "rc": 1, "results": []}
```

CentOS Linux 8 had reached the End Of Life (EOL) on December 31st, 2021. It means that CentOS 8 will no longer receive development resources from the official CentOS project. After Dec 31st, 2021, if you need to update your CentOS, you need to change the mirrors to [vault.centos.org](http://vault.centos.org) where they will be archived permanently. Alternatively, you may want to upgrade to CentOS Stream.

Step 1: Go to the `/etc/yum.repos.d/` directory.

```
$ cd /etc/yum.repos.d/
```

Step 2: Run the below commands

```
$ sed -i 's/mirrorlist/#mirrorlist/g' /etc/yum.repos.d/CentOS-*
$ sed -i 's|#baseurl=http://mirror.centos.org|baseurl=http://vault.centos.org|g' /etc/yum.repos.d/CentOS-*
```

Step 3: Now run the yum update

```
$ yum update -y
```