2 - Jenkins and AWX Deployment

Prerequisites

1. AWS Account
2. Terraform Cloud Account
3. Preconfigured access in `~/.terraformrc` . Get the token from https://app.terraform.io by going to Settings Teams Team API Token. Generate a new token and create the file `~/.terraformrc`

```
credentials "app.terraform.io" {
 token = "iz5o8MNxgBBPwQ...."
}
```

#	Step	
1	Setup	1. Check out repository senofi/openidl-devops 2. Create a new folder under `openidl-devops/aws-infrastructure/environments/` by copying the sample folder `openidl-devops/aws-infrastructure/environments/sample-env` ⚠ Make sure there are no other credentials in the `~/.terraform/` folder (if it exists) as they will take precedence over the ones in file `~/.terraformrc`

2	Create IAM User & Role	1. Pull the AWS credentials from AWS Console for the AWS account you have access to. The AWS IAM user needs to have access to IAM to create roles and other users.
		2. Go to `openidl-devops/aws-infrastructure/environments/<env-folder>` as copied in the previous section
		3. Configure `openidl-devops/aws-infrastructure/environments/<env-folder>/org-vars.yaml`
		  a. `Fill in the IAM AWS access and secret keys under section iam of the YAML file`
		  b. `Configure the org ID and the environment ID (dev, test or prod)`
		4. Go to `<env-folder>/iam` and run `terragrunt plan`
		5. After a review apply the changes with `terragrunt apply`
		
		The script creates:
		
		• IAM role (used by the terraform user)
		• IAM user (terraform user)

3	Create Ops Kubernetes Cluster	1. Register manually a new SSH key pair in AWS by going to EC2 Key pairs (RSA, pem file). Create a new key with a name `awx-target` Keep the private key in the `environments` folder or anywhere on the file system you prefer
		2. Go to the Terraform Cloud workspace that was just created in the previous section and go to the States tab. Open the top state in the list and find `outputs` and copy `access_key` and `secret_key` values that will be used for the next step
		3. Go to `<env-folder>/k8s-cluster` and run `terragrunt plan`The previous step should fail but it should have created a new workspace in Terraform Cloud - e.g. `devnet-d3-k8s-cluster`
		4. Make sure the AWS variables are set in `org-vars.yaml` under `terraform:` property
		  a. aws_access_key = terraform user's access key ID
		  b. aws_secret_key = terraform user's secret access key
		  c. region = *us-east-2* or any other region you prefer
		  d. aws_role_arn = terraform role ARN
		  e. aws_external_id = terraform
		5. Run `terragrunt plan`
		6. Review and if things look ok run `terragrunt apply`
		7. Acknowledge the run with `yes` in the prompt
		
		The script creates:
		
		• Kubernetes cluster
		• PostgreSQL DB for Ansible Tower (AWX)
		• VPC, network

4	Import the Kubernetes Cluster connection config	Make sure you have an AWS profile set in your `~/.aws/config` and `~/.aws/credentials`
		
		~/.aws/config
		
		`[profile tf-user]`
		`region = us-east-2`
		`external_id = terraform`
		
		`[profile tf-role]`
		`external_id = terraform`
		`source_profile = tf-user`
		`role_arn = arn:aws:iam::<aws-account-number>:role/tf_automation`
		`region = us-east-2`
		
		~/.aws/credentials
		
		`[tf-user]`
		`aws_access_key_id = AKI...`
		`aws_secret_access_key = r3AB...`
		
		Find the name of the Kubernetes cluster and update the local config with it
		
		`export AWS_PROFILE=tf-role`
		`aws eks update-kubeconfig --name ops-k8s`

5	Install Nginx	1. Install Nginx Ingress controller
		
		`kubectl create ns ingress-nginx`
		`helm repo add ingress-nginx https://kubernetes.github.io/ingress-nginx`
		`helm install -n ingress-nginx lb ingress-nginx/ingress-nginx`
		
		It is possible that the nginx LB will not be assigned DNS and IP due to the security group for the cluster and the nodes tagged with the same annotation. To fix that find the security group for the nodes (e.g. ops-k8s-node) and remove the *owned* tag.

6	Install Jenkins	Use the helm chart for installing Jenkins onto the Kubernetes cluster created above.
		
		`cd <devops-repo>/jenkins`
		`kubectl create ns jenkins`
		`helm repo add jenkins https://charts.jenkins.io`
		`helm upgrade --install -n jenkins jenkins jenkins/jenkins --values values.yaml`
		
		Wait for Jenkins to start up.
		
		To view the Jenkins admin password:
		
		`kubectl exec --namespace jenkins -it svc/jenkins -c jenkins -- /bin/cat /run/secrets /additional/chart-admin-password && echo`
		
		Set up a cloud-provisioned Jenkins node as defined in the Kubernetes plugin config in Jenkins.

7	Install Ansible Tower (AWX)	Create the AWX DB by connecting to the RDS PostgreSQL instance created via Terraform.
		
		1. Create an SSH Tunnel. Lookup the RDS DB DNS and the EC2 instance that is the AWX target public DNS and replace them in the command line template:
		
		`ssh -i <env-folder>/awx-target.pem -N -L 5432:ops-tools-db.<instance-id>.us-east-2.rds.amazonaws.com:5432 ubuntu@<awx-target-ec2>.us-east-2.compute.amazonaws.com -vv`
		
		2. Connect with DBeaver (or another PostgreSQL client) on localhost port 5432 and run the following SQL after replacing <pass> with an actual password (as defined under `environments/<env>/org-vars.yaml`)
		
		`create database awx;`
		`create user awxuser with encrypted password '<pass>';`
		`grant all privileges on database awx to awxuser;`
		
		3. Configure the Kustomize script `awx-custom.yaml` by replacing the DB settings in awx-operator folder under openidl-devops Git repository.
		
		Install AWX with the Kustomize command.
		
		`cd awx-operator`
		`helm repo add awx-operator https://ansible.github.io/awx-operator/`
		`kustomize build . | kubectl apply -f -`
		
		Watch for the script failing and if it does run it again (timing issue due to the creation of the AWX RBAC)

8	Update DNS record (optional)	1. Go to the AWS Account Route53 2. Create a new Hosted Zone (e.g. d1.test.openidl-org-test.net) 3. Under the new hosted zone create a new entry of type A with an Alias for the Kubernetes cluster (e.g. ops.d1.test.openidl-org-test.net) to point to a Classic Load Balancer Now Jenkins and AWX should be available via http://ops.d1.test.openidl-org-test.net/ and http://ops.d1.test.openidl-org-test.net/jenkins.