CI with AVCTL and Github Actions

This guide explains how we deploy an agent, or update a deployed agent on agentverse using AVCTL.

Structure

We have aGitHub repo(opens in a new tab) where you can template or clone this code.

It is made up of three parts, but really two we have our deployment scripts in.github/workflows and inscripts/ and our Agent is underagent/ .

├─── README.md ├─── .github │ └─── workflows │ └─── deploy-agent.yaml ├─── agent │ ├─── README.md │ ├─── agent.py │ ├─── poetry.lock │ └─── pyproject.toml └─── scripts └─── deploy-agent.sh This logic is very simple, we have a Github workflow that triggers when the branch main is updated. It then calls the deployment script inscripts/deploy-agent.sh ; this script first checks if the Agent already has an address defined in.avctl/config.toml ; in case it does, is this agent registered? If it isn't, we proceed and register this agent.

If it is registered, let's stop the remote Agent and update it.

You can see the deploy script below, or onGithub(opens in a new tab)

Self hosted deploy-agent.sh

Define the function

get_agent_address ()

{ local file= ".avctl/config.toml "

Check if the file exists

if [- f "file"]; then

Extract the address value

agent_address=(grep 'address ='

"file"

| sed - E 's/.= "(.)"/\1/')

Check if the address is not empty

if [- n "agent_address"]; then echo agent_address else echo "" fi else echo "" fi }

Define the specific directory to work on

defined_directory

"agent/"

Change to the specified agent directory

cd "defined_directory"

Create a .staging.avctl folder for new agents if it doesn't exist

avctl hosting init

get the agent address if it exists

agent_address

(get_agent_address)

Get the agent's name from the README.md top line header

agent_name

(head

n 1 README . md | sed - e 's/#//g'

| xargs)

If the address exists...

if [- n "agent_address"]; then avctl hosting get agent - a "agent_address" response = (avctl hosting get agent - a "agent_address") \

Check if the agent is already in existence, if it isn't, deploy as new, else sync.

if [? - eq 0]; then avctl hosting stop - a "agent_address" avctl hosting sync - a "agent_address" else avctl hosting deploy - n "agent_name" --no - dependency - check || true fi

Agent doesn't exist, so let's deploy

else avctl hosting deploy - n "agent_name" --no - dependency - check || true fi

Getting started

Head on over to theAgentverse(opens in a new tab) and sign in. Under your profile link (top right) there is an option forAPI Keys :

Clicking this, takes you to an API Key window; here click new+ New API Key , give the key a name and give this API key full permissions. Click generate API Key at the bottom of the page, and copy the output. Detailed instructions can also be foundhere

Once you've got yourAPI_KEY , be sure to have forked theGithub repo(opens in a new tab) , and visit that repo. Go to settings, on the left hand menu selectSecrets and variables , and click actions from the drop down.

You'll get a window like shown below:

ClickNew repository secret and enter theAPI_KEY ; we have named oursAGENTVERSE_API_KEY .

Great! With that set, copy it in your Agent code below Agent definition part. It is assumed here that your Agent is tested, and you also have an account onAgentverse(opens in a new tab) .

Now, let's push:

git add . git commit -m "updating agent" git push Visit your forked repo Github page, and under actions you should see the runner in action:

Possible error

You may need to locally run:

sudo git update-index --chmod=+x scripts/deploy-agent.sh This tells git to update the permission on the executable script. Then push up the changes for them to take effect.

Running this locally:

Please follow the installation guidehere

Update the permissions ondeploy-agent.sh (You should only need to do this once):

chmod +x scripts/deploy-agent.sh Login to Agentverse from terminal:

avctl auth login Then, from terminal run:

./scripts/deploy-agent.sh You should see output similar too, dependent on your Agents deployed state:

josh@vm avctl-ci-example % ./scripts/deploy-agent.sh Project already initialized Agent exists on agentverse under address: 'agent1qfx5mmewjs4x9ysyxemsaxv6empds4mmpx4sav84yagmhed5yczdwtqkcxu' Agent agent1qfx5mmewjs4x9ysyxemsaxv6empds4mmpx4sav84yagmhed5yczdwtqkcxu has been stopped. Pushing latest code... Everything is up to date. Nothing to push Agent agent1qfx5mmewjs4x9ysyxemsaxv6empds4mmpx4sav84yagmhed5yczdwtqkcxu is now running! josh@vm avctl-ci-example %

Quirks

For security reasons, the Agentverse defines your Agent's address and stores your Private Key. An address you set locally will not be applied on Agentverse using the above method.

Further steps

To get familiar with AVCTL, we recommend reading the other guides in this series:AVCTL andAVCTL hosting .

Last updated on November 6, 2024

Was this page helpful?

You can also leave detailed feedbackon Github

AVCTL Hosting commands Agents and Functions creation APIs

On This Page