

# macOS EC2 Instances

Как пользоваться  
и масштабировать для CI/CD

Kyiv DevOps Community Meetup November 2021



# Привет! Я Сергей Василенко:

- 2011 по 2020: tech support → sys admin → tech lead → team manager
  - С осени 2020 — Developer Experience Engineer в Grammarly 
  - AWS Community Builder
- 
- Пишу в  [serhii.vasylenko.info](https://serhii.vasylenko.info) и  [vasylenko](https://twitter.com/vasylenko)
  - Буду рад связи с вами на LinkedIn: [linkedin.com/in/svasylenko](https://linkedin.com/in/svasylenko)

# План

- Детально о "mac1.metal" instance type
- Создание macOS AMI
- Что нужно для масштабирования
- Развертывание с помощью Terraform
- Эксплуатация

mac1.metal

# Что это и что умеет



- Mac mini: Intel Core i7 / 32GiB RAM
- AML: Mojave, Catalina, Big Sur и Monterey
- Поддержка Systems Manager:
  - Sessions Manager
  - Run Command
  - И другие фичи SSM, но не все

# Что это и что умеет



- Mac mini: Intel Core i7 / 32GiB RAM
- AMI: Mojave, Catalina, Big Sur и Monterey
- Поддержка Systems Manager:
  - Sessions Manager
  - Run Command
  - И другие фичи SSM, но не все
- EC2 macOS Init (аналог cloud-init для Linux EC2)
- ENA drivers (внутри VPC сеть действительно очень быстрая)
- Поддержка изменений разрешения виртуального дисплея
- Есть [канал в SNS](#) с обновлениями base AMI

# Нюансы

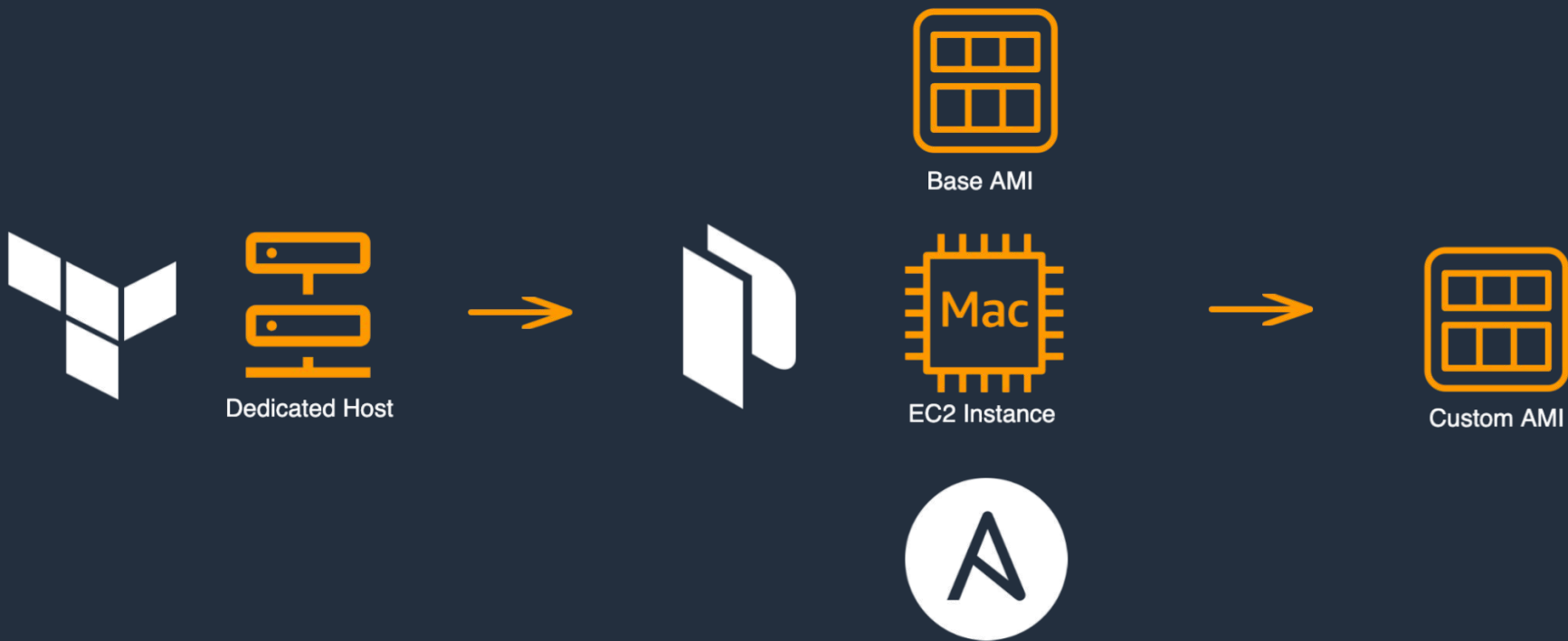
- Это Dedicated Host
- Стоит 1.083 USD в час
- Поддержка Savings Plans (макс. 44% экономии)
- Нет поддержки AWS Inspector (альтернатива — Tenable)

# Нюансы

- Это Dedicated Host
- Стоит 1.083 USD в час. Минимум 24 часа (затем per-second billing)
- Поддержка Savings Plans (макс. 44% экономии)
- Нет поддержки AWS Inspector (альтернатива — Tenable)
- Размер APFS container при запуске
- Ограниченное количество регионов (13) и AZ
- Запуск и остановка дольше, чем у других типов EC2
- Удаление инстанса запускает механизм очистки Dedicated Host  
(обещают добавить выключатель для этого)



# Создание АМІ



Общая схема

# Подготовка Dedicated Host



```
resource "aws_ec2_host" "example_host" {  
    instance_type      = "mac1.metal"  
    availability_zone   = "us-east-1a"  
    auto_placement     = "on"  
    tags               = "..."  
}
```

\*Поддержка ec2\_host в AWS provider начиная с версии v3.61.0

# Создание AMI



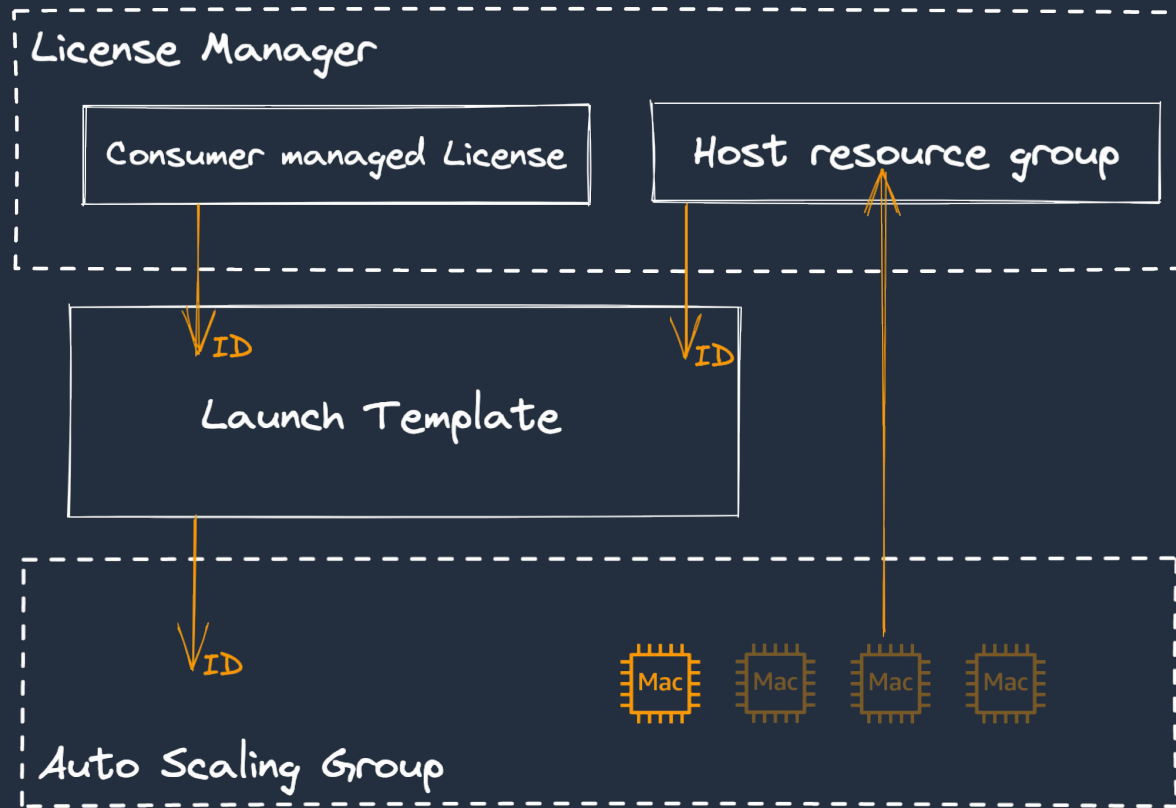
```
"aws_polling": { ... }  
"ssh_timeout": "20m"  
"subnet_filter": { ... "availability-zone-id" ... }  
"tenancy": "host"  
"source_ami_filter": { ... "name": "amzn-ec2-macos-11*" ... }
```

# Конфигурация



- Модулей для macOS не много, но достаточно:
  - homebrew
  - launchd
  - osx\_defaults
  - shell (для всех остальных случаев)
- Встроенные в macOS CLI утилиты умеют очень многое
- Установка Xcode:
  - xip --expand Xcode\_XX.Y.xip
  - xcodebuild -license accept
  - xcodebuild -runFirstLaunch

**Как масштабировать**



Общая схема

# Особенности решения



- Поддержка ресурсов AWS провайдером:
  - `aws_licensemanager_license_configuration`
  - `aws_launch_template`
  - `aws_autoscaling_group`

Host resource group — через CloudFormation (но в Terraform)

- AWS Resource Access Manager, если хотите использовать отдельный аккаунт организации





```
resource "aws_licensemanager_license_configuration" "this" {
  name                     = local.full_name
  license_counting_type   = "Socket"
}

resource "aws_cloudformation_stack" "this" {
  name           = local.full_name
  template_body = file("${path.module}/resource-group-cf-stack-template.json")
  parameters = {
    GroupName = local.full_name template
  }
  on_failure = "DELETE"
}
```

Пример кода для Licence Configuration и Host Group



```
data "aws_ec2_instance_type_offerings" "mac1metal" {
  filter {
    name      = "instance-type"
    values    = ["mac1.metal"]
  }
  location_type = "availability-zone-id"
}

data "aws_subnet_ids" "macmetal_supported" {
  vpc_id = data.aws_vpc.this.id
  filter {
    name      = "availability-zone-id"
    values    = data.aws_ec2_instance_type_offerings.mac1metal.locations
  }
}
```

Пример кода для выборки AZ, которые поддерживают mac1.metal

Эксплуатация

# Конфигурация при запуске с EC2 macOS Init

- Launch Daemon, который запускается от root при загрузке
- Выполняет набор команд поочередно и по группам
- Модульная конфигурация, TOML формат
- Выполняет скрипты, переданные через user-data
- Несколько сценариев запуска
- [Open Source](#)

# Auto Scaling lifecycle hooks

- Возможность выполнить действие при запуске/удалении
- Интеграция с EventBridge, а через него:
  - ... с Systems Manager Run Command
  - ... или с Systems Manager Automation
  - ... или с Lambda
  - ... или с SNS
  - ... или с Step Functions

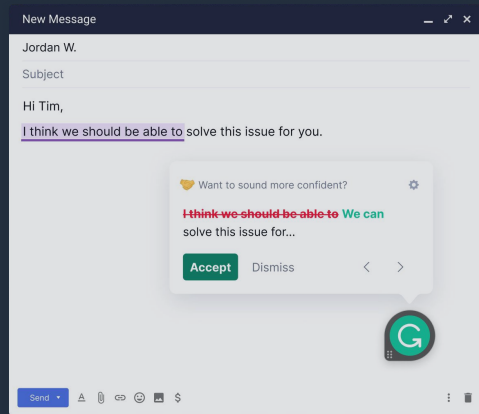
Пример, когда это полезно: удаление регистрации runner в GitLab

Когда и для чего нужен `mac1.metal`

# Пример использования в Grammarly

Сборка и тестирование приложений:

- Grammarly Keyboard
- Grammarly for Mac
- Safari extension
- Chrome extension



Каждое приложение разрабатывает отдельная команда —  
у каждой команды свой набор серверов



- Высокая стоимость  
(хотя Saving Plan поможет)



- Нативная интеграция с AWS
- Self-service для команд
- Высокая производительность:
  - RAM / CPU
  - Network / Disk (EBS)
- Простота масштабирования
- Удобные инструменты:
  - Ansible/Packer — создание AMI
  - Terraform — развертывание



# Спасибо за внимание!

Готов ответить на ваши вопросы 😊